

Discovering and Mitigating Social Data Bias

by

Fred Morstatter

A Dissertation Presented in Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy

Approved June 2017 by the  
Graduate Supervisory Committee:

Huan Liu, Chair  
Subbarao Kambhampati  
Ross Maciejewski  
Kathleen M. Carley

ARIZONA STATE UNIVERSITY

August 2017

## ABSTRACT

Exabytes of data are created online every day. This deluge of data is no more apparent than it is on social media. Naturally, finding ways to leverage this unprecedented source of human information is an active area of research. Social media platforms have become laboratories for conducting experiments about people at scales thought unimaginable only a few years ago.

Researchers and practitioners use social media to extract actionable patterns such as where aid should be distributed in a crisis. However, the validity of these patterns relies on having a representative dataset. As this dissertation shows, the data collected from social media is seldom representative of the activity of the site itself, and less so of human activity. This means that the results of many studies are limited by the quality of data they collect.

The finding that social media data is biased inspires the main challenge addressed by this thesis. I introduce three sets of methodologies to correct for bias. First, I design methods to deal with data collection bias. I offer a methodology which can find bias within a social media dataset. This methodology works by comparing the collected data with other sources to find bias in a stream. The dissertation also outlines a data collection strategy which minimizes the amount of bias that will appear in a given dataset. It introduces a crawling strategy which mitigates the amount of bias in the resulting dataset. Second, I introduce a methodology to identify bots and shills within a social media dataset. This directly addresses the concern that the users of a social media site are not representative. Applying these methodologies allows the population under study on a social media site to better match that of the real world. Finally, the dissertation discusses perceptual biases, explains how they affect analysis, and introduces computational approaches to mitigate them.

The results of the dissertation allow for the discovery and removal of different levels of bias within a social media dataset. This has important implications for social media mining, namely that the behavioral patterns and insights extracted from social media will be more representative of the populations under study.

## DEDICATION

To my parents, Miles and Liela Morstatter.

## ACKNOWLEDGMENTS

I would like to thank my advisor, Huan Liu, for all of his terrific support throughout the course of my PhD. Without his tremendous encouragement and mentoring, this thesis would not have been possible. I consider myself to be truly lucky to have been under his guidance and I look forward to our continued collaborations in the future. I would like to thank my committee members, Subbarao Kambhampati, Ross Maciejewski, and Kathleen M. Carley for their valuable feedback.

I have been a member of the Data Mining and Machine Learning (DMML) lab at Arizona State University starting before I began my PhD, and continuing throughout the course of my graduate studies. As a member of this lab, I have had the pleasure of working with and learning from these many highly talented and encouraging researchers: Ali Abbasi, Salem Alelyani, Aneeth Anand, Geoffrey Barbier, Ghazaleh Beigi, Kewei Chen, William Cole, Harsh Dani, Philippe Faucon, Huiji Gao, Pritam Gundecha, Xia Hu, Isaac Jones, Shamanth Kumar, Jundong Li, Yunzhong Liu, Tahora H. Nazer, Ashwin Rajadesingan, Suhas Ranganath, Justin Sampson, Shashvata Sharma, Kai Shu, Jiliang Tang, Rob Trevino, Ran Wang, Suhang Wang, Xufei Wang, Liang Wu, Reza Zafarani, and Zheng Zhao. I would like to thank Zheng for his support while I was an undergraduate student. He is a terrific mentor, and helped me to become interested in a research career.

Throughout the course of my PhD I had the privilege of mentoring some terrific undergraduate students. Through ASU's FURI program I had the opportunity to work with Dan Baird, Lisa Baer, and Grant Marshall. Through Barrett, The Honors College's Undergraduate Thesis program I had the opportunity to work with Kian Fakhri, Mark Karlsrud, Grant Marshall, and Lew Ruskin. Through The School of Computing, Informatics and Decision Systems Engineering's capstone program

I worked with Wes Bowman, Stefano Chang, Jose Eusebio, Marcus Finney, Cory Harasha, Arian Hatefi, Christina Huff, David Kahn, Nitin Karki, Jake Krammer, Alexandra Nazareno, Justin Sampson, Sumbhav Sethia, Joshua Stiefer, and Terrance Williams. Outside of these interactions, I have had the privilege of supervising undergraduates who helped with our funded projects: Daniel Howe, Mark Karlsrud, and Grant Marshall. I am grateful to all of these students for their assistance in tasks in the lab, and for helping me to develop my skills as a mentor.

Arizona would not have been enjoyable without the support of my dear friends. I would especially like to thank Bijan Fakhri, Kian Fakhri, Daniel Gonzalez, Daniel Howe, Kurt Landenberger, Ashwin Rajadesingan, Robert Trevino, Jose Trigueros, Alex Vaske, Liang Wu, and all of my other friends for their support.

I would also like to extend my sincerest thanks to Dr. Rebecca Goolsby, the program manager from the Office of Naval Research. Her constant mentoring, support, and guidance has helped to shape the course of my research and to address impactful research questions. I have thoroughly enjoyed our discussions and training events.

Finally, to my mother and father for their love and support. To Rio Cavendish for her love, support, and patience through my many years of graduate study. I am extremely grateful to you.

The material presented herein is based upon work supported, in part, by the Office of Naval Research (ONR) through grant numbers N000141612257, N000141410095, N000141110527, N000141010091 and by ONR through MINERVA grant N000141310835 on State Stability.

# TABLE OF CONTENTS

	Page
LIST OF TABLES .....	xi
LIST OF FIGURES .....	xiii
CHAPTER	
1 INTRODUCTION .....	1
1.1 Problem Statement .....	2
1.1.1 The Scale of the Problem .....	4
1.2 Problems Addressed in this Thesis .....	6
2 RELATED WORK .....	10
2.1 Dimensions of Social Data Bias .....	11
2.1.1 Unintentional Exogenous Bias .....	11
2.1.2 Unintentional Endogenous Bias .....	14
2.1.3 Intentional Exogenous Bias .....	15
2.1.4 Intentional Endogenous Bias .....	16
2.2 Sampling Bias in Social Media Data .....	17
2.3 Assessing the Impact of Data Collection Bias .....	19
2.3.1 Work with the Streaming API .....	19
2.3.2 Bias in Black Box Systems .....	20
2.3.3 Bias in Twitter Data .....	20
2.4 Mitigating Data Collection Bias .....	21
2.5 Identifying Bots in Social Media Data .....	25
2.6 Detecting Shills in Online Social Networks .....	28
2.7 Perceptual Bias in Social Data .....	31
2.7.1 Developing An Emoji Mapping with Word Embeddings ....	32

CHAPTER	Page
2.7.2 Linguistic Resources for Informal Text .....	32
2.7.3 Emoji Analysis .....	33
2.7.4 Platform Specific Emoji Rendering .....	34
3 DETECTING SOCIAL DATA COLLECTION BIAS .....	36
3.1 A Dataset for Detecting Bias .....	38
3.2 Statistical Measures .....	40
3.2.1 Top Hashtag Analysis .....	40
Kendall's $\tau$ of Top Hashtags .....	41
Comparison with Random Samples .....	43
3.2.2 Topic Analysis .....	44
Topic Discovery .....	45
Topic Comparison .....	46
Comparison with Random Samples .....	47
3.3 Network Measures .....	48
3.3.1 Node-Level Measures .....	48
3.3.2 Network-Level Measures .....	50
3.4 Geographic Measures .....	51
3.5 Discussion .....	53
4 ASSESSING AND MITIGATING SOCIAL DATA COLLECTION BIAS	56
4.1 Assessing Bias in Previously Collected Datasets .....	56
4.1.1 Discovering Bias in the Streaming API without the Firehose	58
Validation of the Streaming API .....	58
Vetting the Randomness of the Sample API .....	59
Finding Bias in the Trend of a Hashtag .....	62



CHAPTER	Page
Signal Usability Under Sparsity .....	64
4.1.2 Geographic and Temporal Stability of Queries .....	65
Between-Country Results .....	67
Between-Time Results .....	67
4.1.3 Discussion .....	68
4.2 Mitigating Social Data Collection Bias .....	69
4.2.1 Designing Novel Crawling Approaches.....	70
4.2.2 Round Robin Splitting .....	71
4.2.3 Spectral Splitting.....	73
4.2.4 Performance of Splitting Approaches .....	76
5 BIAS FROM MALICIOUS ACTORS .....	78
5.1 Identifying Bots: The Importance of Recall.....	79
5.1.1 Bot Definition .....	81
5.1.2 Problem Statement .....	81
5.1.3 Labeled Datasets for Bot Detection .....	82
Arab Spring in Libya .....	83
Arabic Honeypot Dataset .....	84
5.1.4 Bot Detection Approaches .....	86
Heuristics .....	87
Topic Modeling .....	89
Supervised Model for Bot Identification.....	90
5.1.5 Experimental Results .....	94
Bot Classification .....	94
Sensitivity of Topic Modeling .....	96

CHAPTER	Page
5.1.6 Discussion .....	97
5.2 Tampering Bias in Social Data Streams .....	98
5.2.1 Anticipating the Sample API .....	100
5.2.2 Manipulating the Sample API .....	101
5.2.3 Discussion .....	102
5.3 Bias from Sophisticated Actors: The Case of Shills .....	103
5.3.1 Social Media Shills .....	105
5.3.2 Collecting Data for Studying Shills .....	107
5.3.3 Characterizing Shills .....	110
5.3.4 Skill Characteristics .....	110
Feature Analysis: Clustering .....	114
Feature Analysis: User Behavior .....	114
5.3.5 Identifying Shills .....	115
5.3.6 Skill Detection .....	116
Sensitivity to $K$ .....	117
Feature Importance .....	118
Error Analysis .....	119
Identifying the Most Informative Shills .....	120
Comparing Shills with Other Covert Users .....	122
5.3.7 Discussion .....	124
6 PERCEPTUAL BIAS IN SOCIAL DATA: CROSS-PLATFORM EMOJI MISINTERPRETATION .....	126
6.1 Analyzing Platform-Specific Emoji Usage .....	129
6.1.1 Collecting a Platform-Specific Emoji Dataset .....	130

CHAPTER	Page
6.1.2 Measuring Emoji Embedding Agreement .....	132
6.1.3 Assessing Emoji Sentiment .....	134
6.1.4 The Scale of Misinterpretation.....	136
6.2 A Cross-Platform Emoji Mapping Solution .....	138
6.2.1 Building the Embedding .....	139
6.2.2 Constructing the Mapping .....	140
6.3 Evaluating the Mapping .....	141
6.3.1 Predictive Evaluation .....	142
6.3.2 Experimental Setup .....	143
6.3.3 Experimental Results .....	145
6.3.4 Results by Sentiment Threshold .....	146
6.4 Discussion .....	146
7 CONCLUSIONS .....	150
7.1 Methodological Contributions .....	150
7.2 Future Directions.....	152
REFERENCES .....	154

## LIST OF TABLES

Table	Page
1. Parameters Used to Collect Data from Syria. Coordinates Below the Boundary Box Indicate the Southwest and Northeast Corner, Respectively. The Last Keyword Is the Arabic Word for “Syria.” .....	39
2. Average Centrality Measures for Twitter Retweet Networks for 28 Daily Networks. “All” Is All 28 Days Together. ....	49
3. Comparison of Network-Level Social Network Analysis Metrics. ....	52
4. Geotagged Tweet Location by Continent. Excluding Boundary Box from Parameters. ....	53
5. Data Collected to Test Bias Detection Approach Introduced in This Chapter.	59
6. Significance Levels of $\tau_\beta$ Statistic for Top $K$ Hashtags, Sample API vs. Firehose. All Lists of Size Greater Than 40 Had $p$ -Values $< 10^{-6}$ . ....	59
7. Number of Comparisons, Median, Average, and Standard Deviation of Twitter ID Jaccard Scores across All Comparisons. Because the Temporal Comparisons Are between Query, We Have One Less Than in the Geographic Comparison. ....	67
8. Sample Coverage by Approach. “1-Split” Is the Amount of Data That We Would Get with No Splitting Approach. ....	76
9. Statistics of the Data Used for Bot Detection. ....	83
10. <i>HeuristicTime</i> Measure on the Arabic HoneyPot Dataset. ....	95
11. <i>BoostOR</i> Measure on the Arabic HoneyPot Dataset. ....	95
12. The Precision, Recall and $F_1$ Measure of Different Models on Libya Dataset.	96
13. The Precision, Recall and $F_1$ Measure of Different Models on HoneyPot Dataset. ....	96

Table	Page
14.Dataset Statistics .....	110
15.Confusion Matrix Showing the Average Cosine Similarity between the Two Groups of Clusters. ....	114
16.Classification Results. ....	117
17.Most Important Features in Our Dataset. These Are the Features Which Effected Performance by at Least 3%. ....	119
18.Results of Different Approaches on the Bot Detection Task. “Shills” Is the Logistic Regression Approach for Detecting Shills. ....	123
19.Amount of Data Collected by Platform. The “Source(S)” Column Indicates the Applications We Chose to Represent Each Platform. ....	129
20.Average Jaccard Coefficient of Emojis across Platforms. Random Indicates a Random Sample across All Tweets of the Size of the IOS Corpus. ....	131
21.Example Emojis Provided to Illustrate the Difference in Meaning. The Names and Codes Are Official, Provided by the Emoji Unicode Standard. ....	135
22.Sentiment Threshold Significance T-Test between “Mapping” and “No Map- ping” Experimental Designs. The Only Insignificant Mapping Is “IOS → Windows.” .....	147
23.Sentiment Threshold Significance T-Test between “Mapping” and “No Emoji” Experimental Designs. Significance T-Test Results for Sentiment Classifica- tion with Respect to Sentiment Threshold. All the Results Are Significant with $\alpha = 0.05$ . ....	148

## LIST OF FIGURES

Figure	Page
1. Overview of the Social Data Collection Process: Human Behavior Becomes Data upon Which Research Is Conducted. Humans Generate Data on Social Platforms (“1”) Which Is Then Collected by Researchers in Order to Answer Questions about Their Behavior (“2”). At Both Points, Nonrepresentative Sampling Procedures Are in Place That Will Skew the Results Taken by Researchers on Sampled Social Data. ....	3
2. Breakdown of Papers That Use Twitter (Left), and among Them Who Use the Different APIs as Data Sources (Right). Twitter Is a Major Source for Social Media Research, with 46% of All Papers Using It in Their Findings. Overall, $46\% \times 78\% = 36\%$ of All Papers Use the Streaming API. “Unclear” Refers to Papers That Did Not Properly Disclose How They Obtained Their Tweets. ....	4
3. An Overview of the Topics in This Dissertation. ....	6
4. Overview of the Two Dimensions along Which We Characterize Bias. “Intent” Refers to the Knowledge of the Actor about the Extent to Which They Are Biasing the Data. “Source” Refers to Whether the Bias Originates from Within or Outside of the System. ....	10
5. Tag Cloud of Top Terms from Each Dataset. ....	37
6. Raw Tweet Counts for Each Day from Both the Streaming API and the Firehose. ....	39
7. Distribution of Coverage for the Streaming Data by Day. Whiskers Indicate Extreme Values. ....	40

Figure	Page
8. Relationship between $n$ - Number of Top Hashtags, and the Correlation Coefficient, $\tau_\beta$ . . . . .	42
9. Random Sampling of Firehose Data. Relationship between $n$ - Number of Top Hashtags, and $\tau_\beta$ - The Correlation Coefficient for Different Levels of Coverage. . . . .	43
10.The Jensen-Shannon Divergence of the Matched Topics at Different Levels of Coverage. The X-Axis Is the Binned Divergence. No Divergence Was $> 0.15$ . The Y-Axis Is the Count of Each Bin. $\mu$ Is the Average Divergence of the Matched Topics, $\sigma$ Is the Standard Deviation. . . . .	45
11.The Distribution of Average Jensen-Shannon Divergences in the Random Data (Blue Curve), with the Single Average Obtained through the Streaming Data (Red, Vertical Line). $Z$ Indicates the Number of Standard Deviations the Streaming Data Is from the Mean of the Random Samples. . . . .	45
12.Rank Correlation of Sample API and Gnip Firehose. Relationship between $n$ - Number of Top Hashtags, and $\tau_\beta$ - the Correlation Coefficient for Different Levels of Coverage. . . . .	61
13.Figures Outlining Different Steps of the Process for Finding Bias in the Streaming API. . . . .	62
14.Cumulative Known Zeros Ranked by Hashtag Popularity. We See That the Most Popular Hashtags Have Relatively Few Points of Missing Data, While the Less Popular Hashtags Have Many More. To Help the Reader Separate the Higher-Ranked Hashtags' Missing Values, We Plot the $x$ -Axis on a Log Scale. . . . .	65

Figure	Page
15.The General Idea behind the Splitting Mechanisms Proposed in This Section. The Grey Areas Denote the Relevant Tweets That We Want to Collect. If We Just Have One Crawler Tracking “#imwithher,” Then We Get the Entirety of the Grey Area in the Circle but Lose the Grey “X”s. By Creating Additional Crawlers, We Can Get a Greater Number of Those “X”s, in Turn Getting a Larger Sample of the Data. ....	70
16.Overview of The Round Robin Splitting Approach. ....	71
17.Round Robin Splitting Based on Word Co-Occurrence Tends to Show a Steady Rate of Gain as Additional Crawlers Are Added. ....	73
18.Overview of the Spectral Clustering Splitting Approach. ....	73
19.The Results of Spectral Clustering Show an Increase in the Overall Sample Coverage. However, the Clustering Creates Unbalanced Splits Where One Stream, While Still a Good Cluster, May Contain Significantly More Words Than Others. The Lack of Balance Manifests through Instability in the Rate of Gain from Each Additional Crawler.....	74
20.Results of Different Methodologies at the 3-Split Level. ....	75
21.Illustration of the Models. ....	90
22.Performance of BoostOR Varying Number of Topics.....	94
23.Millisecond of Publish Time from Tweet ID as a Function of the Millisecond When the Tweet Was Sent. The Large Error near the 850 Mark Is an Artifact of the Modulus Property of the Reading. ....	101
24.The Distribution of Feature Values for Features Identified as Most Important by the Wilcoxon Rank-Sum Test. ....	115



Figure	Page
25. $F_1$ As a Function of the Number of Topics Used to Create the Dataset. Here We See That the Performance of the Model Peaks at $K = 50$ Topics.....	118
26.Low-Dimensional Embedding of the Features with Tags Comparing Their Real Label with That Assigned by the Classifier. TP, TN, FP, and FN Are “True Positives”, “True Negatives”, “False Positives”, and “False Negatives” Respectively.....	119
27.Differences in Word Choices between Normal Shills and the Shills That Add the Most Predictive Power in the Model. The $X$ -Axis Is the Frequency of the Word in the Corpus, and the $Y$ -Axis Is the Probability That a Predictive Shill Uses This Word, Divided by the Probability That a Regular Shill Uses It. Words above $10^0$ Are Preferred by the Predictive Shills, and Words below This Line Are Preferred by Normal Shills. We Annotate Words That Are Used Significantly More Than the Other Class. ....	121
28.Average Sentiment for Each Platform by Emoji. Error Bars Indicate the Variance Calculated from the Bootstrapped Samples. A Sentiment Score of 0.0 Is “Neutral,” 1.0 Is “Perfectly Positive,” and -1.0 Is “Perfectly Negative.” the $X$ -Axis Labels Indicate the Unicode Character Code of the Emoji. ....	133
29.An Illustration of Platform-Dependent Emoji Mapping Construction. The Three Rectangles Blue, Green, Red Means Corpus of Three Platforms. Each Line Is a Sentence. Triangle and Star Denote Emojis. Red Triangle Means an Emoji in Red Platform While Yellow Triangle Means the Same Emoji in Yellow Platform. ....	137

30. Performance Results across All “Source” X “Target” Pairs. Asterisks Indicate (Source, Target) Pairs Where the Mapping Is Not Significantly Better Than the Two Baselines. $F_1$ Is Computed with Respect to the “Positive” Class. Tweets with Sentiment Scores in the Range of $(-0.2, 0.2)$ Were Deleted. . . . .	142
--	-----

## Chapter 1

### INTRODUCTION

Social media is an important outlet to understanding human activity. Over the last few years many social media sites have given users a way to express their interests, friendships, and behavior in an online setting. Because of their ubiquity, these platforms have been critical in many global events. During the Arab Spring protests, these platforms helped protesters to organize. Across several natural disasters such as Hurricane Sandy, several earthquakes, typhoons, and floods, social media has been used both by the affected to request assistance as well as for humanitarian aid agencies to spread information about critical aid resources. Social media is used by everyday people to discuss current events, and their lives. There are several sites with hundreds of millions of users, and a few sites with billions of users all sharing, posting, and discussing what they see around them.

Noticing the richness, extent, scale, and dynamic nature of social media data, researchers welcome the opportunities to use social data to answer questions regarding human behavior, information diffusion, and influence propagation on social media (Liu et al. 2016). Though not all social media sites provide sample data even for research purposes, some sites do provide mechanisms through which researchers can obtain a sample of data to conduct their research. One example is Twitter, a microblogging site where users exchange short, 140-character messages called “tweets.” Ranking as the 8th most popular site in the world by the Alexa rank in August of 2016,<sup>1</sup> the site boasts 313 million monthly users publishing 500 million tweets per day. Twitter’s

---

<sup>1</sup><http://www.alexa.com/topsites>

platform for rapid communication is a vital platform in recent events including Hurricane Sandy,<sup>2</sup> the Arab Spring (Shamanth Kumar et al. 2013), and several political campaigns (Tumasjan et al. 2010; Gayo-Avello, Metaxas, and Mustafaraj 2011). As a result, Twitter’s data has been coveted by both computer and social scientists to better understand human behavior and dynamics. Because of its open nature with sharing data as well as the vast richness and size of the data generated on Twitter, many research projects use Twitter data to understand human behavior online. This has led to Twitter being called the “fruit fly”, or model organism, for computational social sciences research (Tufekci 2014).

Despite the wide adoption of social media, especially Twitter, for computational social sciences research, there are many problems that stand in the way of making accurate assessments from this data. The data that is obtained from social media sites does not represent the activity of people in the real world (Duggan and Brenner 2013). The objective of this thesis is to provide tools and techniques to identify and mitigate the biases that occur in social media in order to provide researchers with a more unbiased view for their research.

## 1.1 Problem Statement

While social media is an amazing resource for societal research, there may exist biases on the social media platforms. These biases are introduced during data collection. Researchers should account for them in their study, *i.e.*, whether a representative dataset is obtained for planned computational social science research. If the goal

---

<sup>2</sup><http://www.nytimes.com/interactive/2012/10/28/nyregion/hurricane-sandy.html>

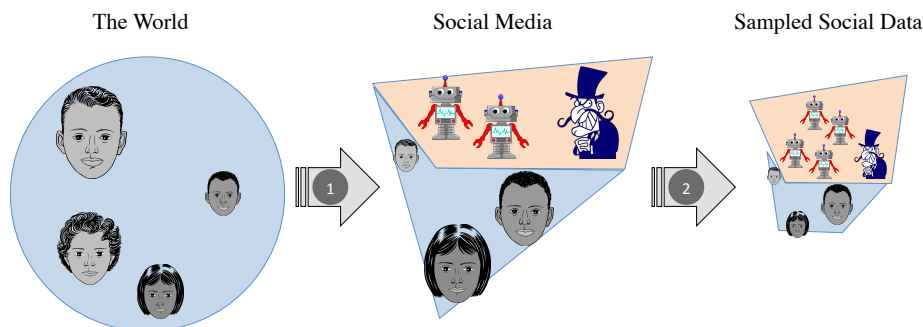
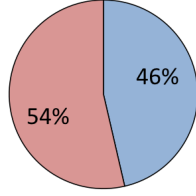


Figure 1: Overview of the Social Data Collection Process: Human Behavior Becomes Data upon Which Research Is Conducted. Humans Generate Data on Social Platforms (“1”) Which Is Then Collected by Researchers in Order to Answer Questions about Their Behavior (“2”). At Both Points, Nonrepresentative Sampling Procedures Are in Place That Will Skew the Results Taken by Researchers on Sampled Social Data.

of computational social science is to study society at scale, then the data studied must provide an accurate reflection of society. In this work I study whether the data that comes from Twitter, the model organism, is representative of society. More specifically, I first study whether or not the sampled data that researchers often use for their research is representative of the full, unsampled data on Twitter. I next study the ways in which Twitter itself is not representative of society. I identify ways in which the data can be biased, and introduce computational techniques for detecting and mitigating these biases. In this work, I focus on bias that arises from sampling strategies on social media and potential sources of data bias, and present ways of detecting and mitigating bias in order to draw credible conclusions about society from sampled and limited data. Both sampling mechanisms in data disclosure, as well as the implicit sampling mechanisms for users on a social media site are focused upon in this dissertation.

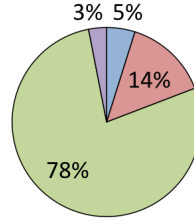
**Twitter Usage at ICWSM (2010 - 2017)**

■ Use Twitter ■ Do Not Use Twitter



**Data Sources in Twitter Papers**

■ Firehose ■ Rest APIs ■ Streaming APIs ■ Unclear



(a) Statistics of Papers That Use Twitter (b) Statistics of Sources in Twitter Papers.  
N = 843. N = 391.

Figure 2: Breakdown of Papers That Use Twitter (Left), and among Them Who Use the Different APIs as Data Sources (Right). Twitter Is a Major Source for Social Media Research, with 46% of All Papers Using It in Their Findings. Overall,  $46\% \times 78\% = 36\%$  of All Papers Use the Streaming API. “unclear” Refers to Papers That Did Not Properly Disclose How They Obtained Their Tweets.

#### 1.1.1 The Scale of the Problem

In our discussion of data collection bias, one natural question which may arise in this work is the emphasis on Twitter’s APIs. Why are they significant? A study performed in 2014 found that among all of the papers published at The International Conference on Weblogs and Social Media (ICWSM), a preliminary conference in the area of social media mining, in 2013, almost half of them used Twitter to obtain some results for their research.

In this thesis we complement their analysis by analyzing the papers from all of the ICWSM conferences held from 2010 - 2017, inclusive. We downloaded 843 papers, including long papers, short papers, and demos. First, we counted the number of papers that used Twitter to obtain results. This was done by using a regular expression. The case-insensitive regular expression “`\bTwitter\b`” was used. to count the number

of papers. Empirically, we discovered that papers that mention Twitter at least 5 times tend to use it in their analysis. Applying this rule to all of the papers from ICWSM, we find that 46% of all papers use Twitter in some capacity for their research. This supports the analysis performed by Tufekci 2014.

Furthermore, we broke down the Twitter-based papers based upon *how* they collected their Twitter data. We used regular expressions to break the papers into one of four categories: Firehose, Streaming/Sample APIs, REST APIs, or Unclear. Unclear means that the paper did not specify how they got to their data well enough for us to classify it into one of the other three categories. To do this, we employed regular expressions. The breakdown of the papers is shown in Figure 2b. We find that 78% of these papers, or 36% of *all* papers at ICWSM, use the Streaming or Sample APIs.

This analysis is provided to give an understanding of the scale of data collection bias, a prominent theme of this dissertation. While we focus on ICWSM for this analysis, it is not limited to this one conference. Twitter data is also used widely in other major academic venues such as the World Wide Web (WWW) Conference, and the Computer-Supported Cooperative Work and Social Computing (CSCW) conference. Perhaps more importantly, it is also used widely by advertising agencies and government and private-sector analysts across the world.

Malicious user bias is another problem that deserves intense study to due the prevalence of malicious users. Bots alone contribute to a large amount of the bias on social platforms. Twitter, Inc. claims that bots account for 8.5% of the accounts on the site are automated.<sup>3</sup> Other claims are larger, with one study noting that

---

<sup>3</sup><https://qz.com/248063/twitter-admits-that-as-many-as-23-million-of-its-active-users-are-actually-bots/>

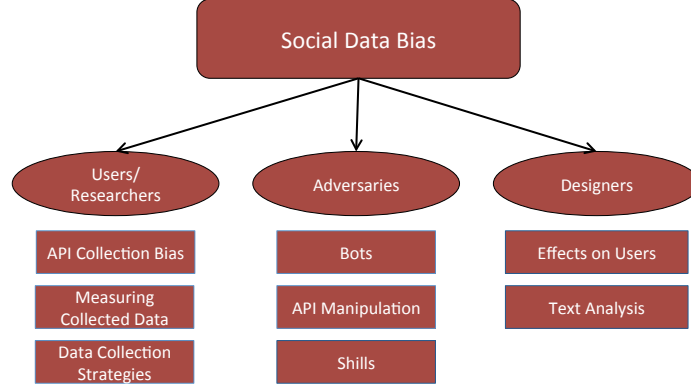


Figure 3: An overview of the topics in this dissertation.

approximately 50% of the accounts created in 2014 were bots.<sup>4</sup> In our own dataset collected from the Firehose (Morstatter et al. 2013), we note that 34.5% of all of the accounts from 2011 have been suspended.<sup>5</sup> These accounts can add a significant amount of bias to any dataset collected from social media.

## 1.2 Problems Addressed in this Thesis

The theme of this dissertation is addressing pitfalls that befall social data research. Under this umbrella, we provide tools and techniques to detect, assess, and mitigate several different types of pitfalls that can occur in social data. The pitfalls discussed in detail in this dissertation can be seen in Figure 1. The pitfalls shown here are major pitfalls, and cover a wide range of the research done using social media data; however, they are not comprehensive. We leave out pitfalls, for example ethical issues related to

---

<sup>4</sup><https://blogs.wsj.com/digits/2014/03/21/new-report-spotlights-twitters-retention-problem/>

<sup>5</sup>Accounts are usually suspended for bot behavior, but there are other reasons that account for a small percentage of these suspended users. These reasons include sharing copyrighted content and hate speech.



handling social data (Zimmer 2010), and the notion that individuals behave differently on online platforms than they would in the real world (Wilson et al. 2009). In the related work chapter, we will discuss other types of pitfalls, such as those introduced in Olteanu et al. 2016. In this thesis, we focus on the following pitfalls for social data analysis:

- **Bias from Researchers.** This pertains to biases from researchers’ assumptions about the validity of their data. When collecting data from social media, one cannot be sure if that data actually represents the true population of the site. I address this question by comparing data obtained through Twitter’s sampled APIs with Twitter’s full dataset. I am the first to address this problem in social data.

Confirming that bias can exist in the way that data is distributed is an important one, but comparing the data from the full dataset is not tractable for most researchers due to the prohibitive cost. Thus, I develop ways for researchers and practitioners to assess the amount of bias in their dataset, as well as to collect data in such a way that the amount of bias is mitigated.

- **Bias from Adversaries.** Bias is not just a factor of bizarre blackbox sampling strategies employed by social data APIs. There are malicious actors who seek to inject bias into the data one receives in order to change the statistics of the site. This is often done in political contexts to inflate the perceived popularity of a candidate or message,<sup>6</sup> but it’s also very effective in other domains like advertising. It is important that those studying social media data accurately

---

<sup>6</sup><https://www.pastemagazine.com/articles/2017/06/how-the-trump-russia-data-machine-games-google-to.html>

identify as many of these malicious actors as possible, not just so that we can remove them, but also so that we can get a better understanding of their agendas. The way social media data is generated provides unique vulnerabilities that can be exploited by malicious actors. I introduce a new attack vulnerability for social data. I also demonstrate simple strategies for identifying users who are using this attack to increase the prevalence of their messages.

Specialized users on social media can introduce bias by skewing the perception of popularity of an idea, candidate or product. Shills are highly specialized and trained to combat opposing opinions. In concluding this section, we discuss how shills can be detected based upon their online behavior and show how they differ from other classes of malicious users.

- **Perceptual Biases from Social Media Designers.** Bias in social media data does not stop at the way data is distributed, or how that distribution is manipulated by nefarious actors. There are also many perceptual biases that can occur in social media data. One example is that the way that data is produced on social media may not be perceived the way that the author intended. This can have real impact on the way that this data is perceived and thus analyzed. I will introduce one type of bias that is unique to the way data is analyzed: that the intended meaning behind the emojis used by the authors of social media posts varies by the platform the author used to write the post.

By addressing these issues, the contribution of the dissertation can be summarized as tools and techniques to identify and remove social data bias. While these biases occur from different perspectives, and can be introduced at different steps in the research process, they all have the side effect of introducing artifacts into the data. These artifacts then will pollute the conclusions taken from social media data. By

applying the techniques in this dissertation, we allow those who use social media data as part of their research to obtain more credible and generalizable results from this social data.

In this dissertation, I will outline the steps that I have taken to address each of these issues. First, in Chapter 2 I will address other work that has been done to address the problem of social data bias and how it supports and furthers our findings and conclusions. Next, in Chapter 3 I introduce our strategy for measuring and detecting data collection bias from social data APIs like the Twitter Streaming API. In Chapter 4 I will introduce a method for measuring the amount of bias that is present in a dataset collected from these APIs, and present methodologies for collecting data from these outlets in such a way that the amount of bias is minimized. Chapter 5 discusses the vulnerabilities of APIs to malicious users, and discusses the ways to detect these users. It then focuses on detecting bots that attempt to pollute social data platforms and introduces the concepts of shills. Chapter 6 introduces conceptual biases that can be present in social media data, and focuses on one type: platform-specific emoji biases. It also introduces ways to correct for them. Finally, Chapter 7 concludes the thesis and discusses areas of future work.

## Chapter 2

### RELATED WORK

This chapter contains other works related to this dissertation. It is organized based upon the topics presented in the subsequent chapters. We begin by presenting a taxonomy of the different types of bias that can manifest in social data. We then continue to present the related work for each of the subsequent chapters in the dissertation.

		Source	
		Exogenous	Endogenous
Intent	Intentional	External influencers trying to skew perceptions. Astroturfing offline. Bots, spammers online.	“Gaming the system.” Gerrymandering offline. Manipulation online.
	Unintentional	“Weapons of math destruction,” cases where human bias is encoded into the data. Recidivism offline, demographic bias online.	Issues caused by the way in which data is collected. Enumerator bias offline. Sampling bias online.

Figure 4: Overview of the Two Dimensions along Which We Characterize Bias. “Intent” Refers to the Knowledge of the Actor about the Extent to Which They Are Biasing the Data. “Source” Refers to Whether the Bias Originates from Within or Outside of the System.

## 2.1 Dimensions of Social Data Bias

In this section we take a broader view to social data bias. We look at other work, not just limited to computer sciences literature, which addresses the issue of bias pertaining to social data. The discussion will be presented with two main dimensions of bias. The first dimension is intent, which is whether or not the data is being intentionally manipulated. I characterize the concerns about bias as coming from intentional skewing of the data by a body of people or a group. The second dimension of bias is whether it is endogenous or exogenous to the data. I view exogenous bias as that which comes from external factors and endogenous as that which occurs in the way the data is handled. A characterization of these dimensions is shown in Figure 4. Below, I will discuss each of the quadrants in detail as well as present research work that helps to further characterize these dimensions. Where appropriate, I will discuss how my work fits into the general theme of the quadrant.

### 2.1.1 Unintentional Exogenous Bias

This refers to the case where bias is unintentionally encoded into data by selection processes external to the system. These unintentional biases can be learned by machine learning algorithms, and perpetuate this same bias in their predictions. This is the type of bias that leads to machine learning algorithms that are more likely to predict that a black person will re-commit a crime (Angwin et al. 2016), that are less likely to predict a black person as the winner of a beauty contest,<sup>7</sup> or are more likely to

---

<sup>7</sup><https://www.theguardian.com/technology/2016/sep/08/artificial-intelligence-beauty-contest-doesnt-like-black-people>

predict Asians as “blinking” in photographs.<sup>8</sup> As Cathy O’Neil points out in her book, *Weapons of Math Destruction*, the common problems leading to these phenomenon are incorrect data and uninterpretable measures, statistics, or algorithms (O’Neil 2016).

The problem of recidivism prediction, or predicting whether a prisoner will reoffend, is one area that has received a lot of attention in terms of fairness. Some public prison systems have hired private companies to algorithmically determine if a person will re-commit. This has been shown to overpredict African Americans as likely to reoffend at a higher rate than whites (Angwin et al. 2016). This problem was underscored by Richard Berk (Berk 2011b), who discusses the legal and data-related problems pertaining to treating false-positives and false negatives with even care. This issue has inspired new algorithms to address these issues. One example is a model, created by Berk, to account for the asymmetric needs of parole officers (Berk 2011a). For example, Cynthia Rudin’s group (Zeng, Ustun, and Rudin 2016) employed a “Supersparse Linear Model” which has two interesting attributes that can be used for this problem. First, it allows the user to specify the true positive rate and false positive rate for each class, allowing to better control the tolerance for false positives. Secondly, the weights learned by the linear model are *small*, and *interpretable*. This allows for decision makers such as judges to understand *why* the model is making the prediction it is making. In the area of reinforcement learning, work has been done to ensure that the actions taken by the agent are “fair” (Jabbari et al. 2016). Their evaluation of fairness employs a definition of fairness that says that an agent should never prefer an action if a more rewarding action is available (Joseph, Kearns, Morgenstern, and Roth 2016). In future work, they show that this can learn at least

---

<sup>8</sup><http://content.time.com/time/business/article/0,8599,1954643,00.html>

as fast as existing reinforcement learning processes (Joseph, Kearns, Morgenstern, Neel, et al. 2016). This has real-world implications for things such as hiring processes.

Beyond recidivism prediction, there are a number of other areas in which bias present in the data can be amplified by machine learning processes. One example is word embeddings. Word embeddings offer a way to represent instances, namely words, through “contexts.” These contexts are learned based upon the proximity of words in a corpus. In addition to this they also often semantic properties that allow users to quickly understand certain aspects of the dataset. One example of these properties is analogies, where simple vector arithmetic is used to find relationships among the words. One famous example of this property is “king” - “man” + “woman” = “queen,” where taking the vector for king and subtracting that of man and adding the one for woman will yield a vector that is close to the word “queen.” It has been found word embeddings trained on real world data reveal unsavory properties about society. For example, Bolukbasi et al. 2016 found that “Computer Programmer” - “man” + “woman” = “Homemaker.” They claim that this finding reflects sexist undertones in the text upon which the word embeddings are trained. To get around this issue, the authors proposed an approach to find dimensions of the model that are correlated with gender, and then skew them such that words that should not be correlated with gender are not. This is done at a slight cost in terms of predictive accuracy of the model. This is a direct approach to debiasing the data. Similar approaches have been taken to debias the data. One example is in a series of papers authored by Wu, where bias is also mitigated through identifying correlated features that should not correlate with the predictive variable (Wu and Wu 2016; Zhang, Wu, and Wu 2016), such as race in the case of housing decisions.

As a final note, it is important to note that the traditional “machine learning”

interpretation of the word bias also fits in here. Many machine learning algorithms are formulated with the understanding that the data is not always centered, and account for this by adding a bias term to the formulation (Bishop 2006). This fits in this category because this type of bias is usually unintentional and occurs in the way the data is generated, not usually within the model.

### 2.1.2 Unintentional Endogenous Bias

This refers to problems that can exist when the mechanisms through which data is distributed cause the data to be collected in such a way that inferences drawn upon it are skewed. We further split this group into two subcategories, those where the data distribution mechanism is known and those where it is not.

In some cases, the way in which the data is generated is known. Despite this, the problem of how to accurately collect samples has a long history. For example, the method through which people are selected for polling is completely visible, but this has consistently caused problems for pollsters. This was true in the widely mispredicted election of Dewey and Truman (Mosteller 1949) and continues to elections today (Allcott, Gentzkow, et al. 2017). This is the case on many social media sites, such as those in which it is customary to create custom software to collect the data. These were commonly used in the early days of collecting data from the web (Boldi et al. 2004; Cho and Garcia-Molina 1999), and correspond to many web-scale datasets that are still in use today.<sup>9</sup> Many attempts have been constructed to combat this type of bias (Granovetter 1976; Ye, Lang, and Wu 2010; Borgatti, Carley, and Krackhardt

---

<sup>9</sup>Two example repositories built from such crawling strategies are the ASU Social Computing Data Repository (<http://socialcomputing.asu.edu/>), and the Stanford SNAP Repository (<https://snap.stanford.edu/data/#socnets>).



2006). In fact, it has been found that many network sampling approaches fail to yield samples that can recover the initial parameters of the underlying distribution (Ahmed, Neville, and Kompella 2013).

While these sampling approaches are important, a new type of sampling has arisen in recent years. Driven both by the abundance and speed of data produced on social media, some sites have acknowledged the use of collecting data for research and applications. The key difference is that instead of writing a crawler that collects the data, the data is provided by the social media site. This creates a sort of “black box” for data distribution. This is where the majority of my work is situated. Previously, researchers just used this outlet to collect data with the assumption that it would be provided in a statistically representative manner. My work initially challenged this assumption (Morstatter et al. 2013) by showing the different ways in which the data from these outlets could be biased. In light of these findings, we also created two approaches for correcting this type of bias (Morstatter, Pfeffer, and Liu 2014; Sampson et al. 2015).

### 2.1.3 Intentional Exogenous Bias

This section refers to the set of malicious actors who work in tandem to make their voice heard at disproportionate levels. This is not distinctive to social media. One real-world example of this astroturfing (Cho et al. 2011), where companies or political advocacy groups hire real people to go to a rally to lobby on their behalf.

This phenomenon is exacerbated in the online world where faces are not counted, but online accounts. This has caused a surge in bot accounts that create fake posts and connections in order to make certain users or topics seem more salient. Many

approaches have been proposed to address this problem (Chu et al. 2010; Lee, Eoff, and Caverlee 2011; Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011; Thomas, Grier, and Paxson 2012; Thonnard and Dacier 2011; Xie et al. 2008). My work is situated in how to identify bots in such a way that the most bots are removed (Morstatter, Wu, et al. 2016).

#### 2.1.4 Intentional Endogenous Bias

This refers to attempts by people or groups to “game the system.” Actors within a system can work in tandem to skew its results. This is becoming increasingly common, especially in online systems. One example is how government censors in China selectively remove content to allow for some contrary opinions, but ultimately stop people from gathering offline (King, Pan, and Roberts 2013, 2014). Recently, the administrators of Reddit<sup>10</sup> manually changed the content of some users posts to make it appear as if they did not favor one presidential candidate.<sup>11</sup> What these methods have in common is that malicious actors privy to the internal working of the system can exploit it to bias content in their favor. Examples of this phenomenon are abundant offline. For example, gerrymandering is a common practice used to draw district lines in such a way as to prefer an outcome for one party (Issacharoff 2002).

Recently, the technique for skewing bias has taken been discovered on social media. In my work I identify a method through which malicious bots can time their tweets

---

<sup>10</sup><http://www.reddit.com>

<sup>11</sup><https://www.cnet.com/news/reddit-ceo-admits-to-editing-user-comments-amid-pizzagate-malarkey/>

so that they are 82x more likely to be selected by Twitter’s APIs (Morstatter, Dani, et al. 2016).

Having discussed this quadrant of different social biases, we now turn our attention to specific areas that are

## 2.2 Sampling Bias in Social Media Data

Twitter’s Streaming API has been used throughout the domain of social media and network analysis to generate understanding of how users behave on these platforms. It has been used to collect data for topic modeling (Hong and Davison 2010; Pozdnoukhov and Kaiser 2011), network analysis (Sofean and Smith 2012), and statistical analysis of content (Mathioudakis and Koudas 2010), among others. Researchers’ reliance upon this data source is significant, and these examples only provide a cursory glance at the tip of the iceberg. Due to the widespread use of Twitter’s Streaming API in various scientific fields, it is important that we understand how using a sub-sample of the data generated affects these results.

From a statistical point of view, the “law of large numbers” (mean of a sample converges to the mean of the entire *population*) and the Glivenko-Cantelli theorem (the unknown distribution  $X$  of an attribute in a population can be approximated with the observed distribution  $x$ ) guarantee satisfactory results from sampled data when the randomly selected sub-sample is big enough when samples are taken *i.i.d.*. From a network algorithmic (Wasserman and Faust 1994) perspective, the question is more complicated. Previous efforts have delved into the topic of network sampling and how working with a restricted set of data can affect common network measures. The problem was studied earlier in (Granovetter 1976), where the author proposes

an algorithm to sample networks in a way that allows one to estimate basic network properties. More recently, Costenbader and Valente 2003 and Borgatti, Carley, and Krackhardt 2006 have studied the effect of data error on common network centrality measures by randomly deleting and adding nodes and edges. The authors discover that centrality measures are usually most resilient on dense networks. In Kossinets 2006, the authors study global properties of simulated random graphs to better understand data error in social networks. Leskovec and Faloutsos 2006 proposes a strategy for sampling large graphs to preserve network measures.

We compare the datasets by analyzing facets commonly used in the literature. We start by comparing the top hashtags found in the tweets, a feature of the text commonly used for analysis. In Tsur and Rappoport 2012, the authors try to predict the magnitude of the number of tweets mentioning a particular hashtag. Using a regression model trained with features extracted from the text, the authors find that the content of the idea behind the tag is vital to the count of the tweets employing it. Tweeting a hashtag automatically adds a tweet to a page showing tweets published by other tweeters containing that hashtag. In Yang et al. 2012, the authors find that this communal property of hashtags along with the meaning of the tag itself drive the adoption of hashtags on Twitter. De Choudhury et al. 2010 studies the propagation patterns of URLs on sampled Twitter data.

Topic analysis can also be used to better understand the content of tweets. Kireyev, Palen, and Anderson 2009 drills the problem down to disaster-related tweets, discovering two main types of topics: informational and emotional. Finally, Yin et al. 2011; Hong et al. 2012; Pozdnoukhov and Kaiser 2011 all study the problem of identifying topics in geographical Twitter datasets, proposing models to extract topics relevant

to different geographical areas in the data. Joseph, Tan, and Carley 2012 studies how the topics users discuss drive their geolocation.

Geolocation has become a prominent area in the study of social media data. In Wakamiya, Lee, and Sumiya 2011 the authors try to classify towns based upon the content of the geotagged tweets that originate from within the town. De Longueville, Smith, and Luraschi 2009 studies Twitter’s use as a sensor for disaster information by studying the geographical properties of users tweets. The authors discover that Twitter’s information is accurate in the later stages of a crisis for information dissemination and retrieval.

## 2.3 Assessing the Impact of Data Collection Bias

Our related work for this portion of the dissertation falls into three areas: one concerning work previously done on Twitter’s Streaming API, another concerning research done to verify other black box systems on the web, and finally another area devoted to previous evidence of bias found on Twitter’s Streaming API.

### 2.3.1 Work with the Streaming API

Twitter’s Streaming API has been used throughout the domain of social media and network analysis to generate understanding of how users behave on these platforms. It has been used to collect data for topic modeling (Hong and Davison 2010; Pozdnoukhov and Kaiser 2011; Shamanth Kumar et al. 2012), network analysis (Sofean and Smith 2012), and statistical analysis of content (Mathioudakis and Koudas 2010), among

others. Researchers’ reliance upon this data source is significant, and these examples only provide a cursory glance at the tip of the iceberg.

### 2.3.2 Bias in Black Box Systems

The topic of assessing the results from a black box system is related to our work. Another web sciences black box that has been studied is Amazon’s Mechanical Turk (AMT). In Crump, McDonnell, and Gureckis 2013, the authors assess the representativeness of the users on AMT, and provide example tasks showing areas where these users give good performance. In a similar vein, Snow et al. 2008 proposes a method to correct for response bias in the results obtained from AMT. In the area of social media research, this topic has also been studied from the perspective of link propagation. In De Choudhury et al. 2010, the authors analyze the effect data sampling has on the way link propagation is perceived. Specifically, the authors study URLs shared on sampled Twitter data.

### 2.3.3 Bias in Twitter Data

There are many potential areas of bias in Twitter. One possible area of bias in Twitter data comes from the demographic makeup of the users on the site, often referred to as “participation bias.” In Mislove et al. 2011, the authors use last names of users to estimate their race, and find that the ethnicity/race distribution of Twitter users diverges widely from U.S. Census estimates. Duggan and Brenner 2013 finds similar results, and also finds that adults aged 18-29, African Americans, and urban residents are over-represented on the site.

The bias we focus on in much of this dissertation is concerned with sample bias from Twitter’s APIs. The work performed in Morstatter et al. 2013 compared four commonly-studied facets of the Streaming API and Firehose data, looking for evidence of bias in each facet. They obtained widely different results across facets. First, they studied the statistical differences between the two datasets, using correlation to understand the differences between the top  $n$  hashtags in the two datasets. They find bias in the occurrence of the top hashtags for low values of  $n$ .

The authors compared topical facets of the text by extracting topics with LDA (Blei, Ng, and Jordan 2003), where they found similar evidence of bias. The authors discover that the topics extracted through LDA are significantly different than those extracted from the gold standard Firehose data. The other facets compared in the data were the networks extracted from the dataset. Here, the authors extracted the User  $\times$  User retweet network from both sources and compare centrality measures across the two networks. They find that, on average, the Streaming API is able to find the most central users in the Firehose 50% of the time. The final facet they compare is the distribution of “geotagged” tweets. Here, they find no bias and that the number of geotagged tweets from the Streaming API is over 90% of that in the Firehose.

## 2.4 Mitigating Data Collection Bias

The predictive power of social media services, such as Twitter, has been used to effectively track and predict the spread of disease (Achrekar et al. 2011; Garcia-Herranz et al. 2014; Gomide et al. 2011). Other efforts have shown promising results by using social media to discover a wide range of collective knowledge such as real-time political polling (Ceron et al. 2014; Tumasjan et al. 2010) and the potential success of movies

at box-office (Asur and Huberman 2010; Lu, Wang, and Maciejewski 2014). However, there are very few standards governing how data from social media is gathered and how research and predictions should be approached (Madlberger and Almansour 2014). A number of studies have shown that the method used for sampling can introduce various forms of bias which introduce error into results and remain largely unnoticed (Gonzalez-Bailon et al. 2014; Morstatter et al. 2013). Very little research has gone into methods for correcting for bias. Morstatter et al. proposed using bootstrapping (Efron and Tibshirani 1994) and secondary data sources to obtain a confidence interval for the occurrence rate of a hashtag between the two sources. Such a statistical difference could be used as a red flag for the presence of bias in the stream (Morstatter, Pfeffer, and Liu 2014).

Several relevant works have attempted to uncover the underlying mechanisms with which Twitter disseminates tweets through its various streaming APIs. In 2013, Morstatter et al. tested the differences between the public Twitter streaming API, which is commonly used by researchers but will only return up to 1% of all available data, and the prohibitively priced “firehose” stream, which offers all of the available data to those willing to pay for it (Morstatter et al. 2013). This work uncovered a number of anomalies such as varying top hashtags, differing topic distributions, and varying network measures when using the same searches on both the paid and free services. They explain that according to the “law of large numbers” if the sample is truly random then it should relate well with the results from the population, in this case the “firehose” results (Morstatter et al. 2013). These differences indicate that the streaming API introduces some form of bias when determining how to limit the results sent to a stream (Morstatter, Pfeffer, and Liu 2014).

The possible causes of this sampling bias have been a continuing source of inquiry.



Joseph, Landwehr, and Carley 2014 used five independent streaming sources the differences between multiple streaming results taken at similar but varying time windows. They used a set of keywords and usernames including stop words such as “the” and “i” and words that they specifically invented for the experiment. In order to determine if starting time had an impact on the results they staggered each start by a fraction of a second. After running these tests multiple times, they showed that, when using the same keywords across each stream, 96% of unique tweets were captured by all streams (Joseph, Landwehr, and Carley 2014). Since the experiment used stop words that undoubtedly make up a significant portion of English tweets, a random distribution method would have given results that varied wildly across the separate streaming connections. This is strong proof that the sampling method used by Twitter is highly deterministic.

Kergl et al. found further evidence of non-random sampling. In earlier forms of the Twitter streaming API, the three data streaming levels which provided 1%, 10%, and 100% of Twitter data were named “Spritzer”, “Gardenhose”, and “Firehose” respectively. These streams, unlike the filter-based stream, provide data from the entire body of current tweets. Through analysis of the unique tweet IDs provided by Twitter, the authors discovered that the unique IDs included data, such as the timestamp when the tweet was created, the data center that created it, and other information related to the underlying technical infrastructure of the Twitter data centers. Analysis of the timestamp in particular showed that, in the limited streams, the timestamps only fell over a specific interval of milliseconds directly related to the percentage of sample coverage specified by the service level (Kergl, Roedler, and Seeber 2014). Though no similar timestamp-based sampling method appears to be used in the filtered search stream, the non-random nature of the filtered data, in

addition to the use of simple deterministic methods in past dissemination schemes, indicates that there are underlying artifacts in the filtered stream infrastructure as well that may be adding bias to gathered samples.

The 1% boundary has proven to be a significant hindrance for applications and research that require as close to a complete set of data as possible. These include mission critical response situations, such as those necessary for emergency response and monitoring applications (S Kumar et al. 2011; Vieweg et al. 2010), as well as any form of research that is highly affected by sample bias. As a direct result of the high cost of the “firehose” service, many users have attempted to develop novel solutions for gathering data to improve either the size and coverage of the dataset (Li, Wang, and Chang 2013) or the overall quality of results for a smaller sample (Ghosh et al. 2013). Li, Wang, and Chang 2013 proposed a system that uses “automatic keyword selection” to determine the best search terms to use for a given search through a keyword cost-based system. Using this method improved topic coverage from 49% of targeted tweets obtained through human-selected keywords to 90% in the automatic system. Ghosh et al. took an alternate approach and attempted to improve the quality of their sample by gathering topic-based twitter data from experts and comparing the “diversity, timeliness, and trustworthiness” to a larger sample of community tweets of the same topic. While the expert-based search did show an improvement in all of these areas, they cautioned that crowd-based data could not be entirely discounted as it captured other significant properties such as the flow of conversation in the topic that is otherwise ignored by experts (Ghosh et al. 2013).

## 2.5 Identifying Bots in Social Media Data

Bot detection approaches in general try to build a classifier that labels a given user as a bot or not. In the first part of this section we introduce different detection methods and group them based on features they extract from the data to train the classifier. Then we discuss techniques of ground truth acquisition, their distinctions, pros and cons from both the perspective of convenience and resulting data quality. Finally, we enumerate evaluation mechanisms that have been used for this task.

Recent statistics show that more than 50% of Twitter accounts are not human users. Social network administrators are well aware of these harmful activities and try to delete these users using their suspension/removal systems. By one estimate 28% of accounts created in 2008 and half of the accounts created in 2014 have been suspended by Twitter.<sup>12</sup> What is not well taken care of is the role of bots in facilitating these malicious activities. In one study, 145,000 accounts survived for months without detection. Today, 16% of spammers in Twitter are bots (Grier et al. 2010).

Although social network spam detection approaches are still insufficient, bot detection in social networks has received wide attention from the research community in recent years (Cook et al. 2014; John et al. 2009; Kanich et al. 2008; Stone-Gross et al. 2011). Methods proposed to solve this problem mostly observe it as a classification task which extracts features from the user in order to classify it as a bot or a human. A categorization of the features used in these classifiers is as follows:

- *Content of posts and messages:* In using content, the goal is to find properties that show the difference between what is published by a bot and what is published by a normal user. The text sent by each bot is significantly different from another

---

<sup>12</sup><http://blogs.wsj.com/digits/2014/03/21/new-report-spotlights-twitters-retention-problem/>

- bot but they all tend to use URLs to link to the content they promote (Xie et al. 2008). Although the content is specific to each bot, they all have high tendency in using URLs for promotional purposes. Furthermore, extracting the sentiment of tweets (Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011; Ratkiewicz, Conover, Meiss, Goncalves, Patil, et al. 2011), language and length of messages (Thonnard and Dacier 2011), number of URLs (Chu et al. 2010; Lee, Eoff, and Caverlee 2011; Wang 2010), number of mentions and similarity between all pairs of tweets posted by a user (Lee, Eoff, and Caverlee 2011), and originality of tweets (Wang 2010) are all examples of using content.
- *Profiles and activities:* Automation involved in generating bot accounts results in profile properties that contain detectable patterns. Bots that were working together towards political disruption in 2011 Russian parliamentary election were found to have similar email addresses and account creation times (Thomas, Grier, and Paxson 2012).

Bots mostly tweet using automatic devices (Chu et al. 2010), have a shorter life time (Lee, Eoff, and Caverlee 2011), short subscriber details (Cook et al. 2014), and use hijacked IP addresses (Thomas, Grier, and Paxson 2012). In an effort to escape from suspension systems, the controller of bots spreads the machines from which he launches his attacks geographically (Kanich et al. 2008).

Even when profile information is not available, the username alone can be a bot indicator. The length of the screen name (Lee, Eoff, and Caverlee 2011) and the distance of the unigram/bigram distribution from verified names (Lee and Kim 2014) can differentiate bots from normal users. Matching how well the screen name matches human typing patterns can differentiate automated users (Zafarani and Liu 2015).

- *Network structure and connections:* To boost their desired effect, Twitter bots follow normal users hoping to be followed back. To this aim they show mass following and unfollowing behaviors and thus their connections show different characteristics. The number of followers, number of friends, ratio of the the friends to followers, percentage of bidirectional friends, and the standard deviation of unique numerical IDs of followers and friends have been all used to represent this (Lee, Eoff, and Caverlee 2011).

In order to build a classifier, we need to obtain a gold standard dataset: a set of users and their bot/human label to evaluate the results. Of course, this information does not come affixed to each user when the researcher collects the dataset. Thus, the onus is on the researcher to collect these labels. There are three main approaches to this:

- *Manual annotation:* As in most other classification problems, manual labeling can be to obtain ground truth. Although this method has been used widely in this field (Chu et al. 2010; Cook et al. 2014; Grier et al. 2010; Ratkiewicz, Conover, Meiss, Goncalves, Patil, et al. 2011; Xie et al. 2008), we still have the challenge of how to choose annotators and how many annotators are sufficient in order to achieve reliable labels. Furthermore, this method does not scale as it takes time to recruit users and funds to pay them.
- *Suspended users lists:* This method leverages the social networking site itself to obtain the labels. The researcher simply observes the users and sees which ones get suspended or removed by the site. This approach is simple, but the concern remains that the reason of suspension and deletion is not usually indicated in such lists. Thus, many suspension lists include users who violated other rules

and regulations of the site. Methods proposed in John et al. 2009; Lee and Kim 2014; Thomas, Grier, and Paxson 2012 use these lists.

- *Honeypots:* A newer ground truth acquisition method is based on using honeypots, bots created by the researcher to lure other bots. Honeypots show non-human behaviors and can be made in groups to increase their effect by connecting and interacting with each other without intervention in activities of normal users. Due to the fact that they are designed in a way that any normal user can immediately tell they are bots, any user in the network that connects to a honeypot will be considered as a bot. By using this method, a researcher can gather a large set of bots active in a network with high confidence. This method has been applied (Lee, Eoff, and Caverlee 2011; Thonnard and Dacier 2011).

When building a classifier, the proposed method should be evaluated by means of an evaluation metric. The most widely used metrics in bot detection are accuracy and precision (Chu et al. 2010; Lee, Eoff, and Caverlee 2011; Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011; Thomas, Grier, and Paxson 2012; Thonnard and Dacier 2011; Xie et al. 2008). Although they can reach extremely high precision, few report their recall. This drawback will have a significant effect as most of the bots will remain undetected (low recall) at the expense of precision.

## 2.6 Detecting Shills in Online Social Networks

The problem of detecting shills online falls closely to three problems: crowdturfing detection, bot detection, and detecting online activists. We organize the discussion of our related work along these themes.

Crowdturfing is the process of maliciously using crowdsourcing systems in order to spread fake reviews. This is close to our problem as it involves the mass proliferation of opinion by paid workers. In Wang et al. 2012, the authors performed an analysis on different crowdsourcing which harbor malicious tasks such as crowdsourcing fake reviews, and spreading disinformation about competitive brands. Lee et al. Lee, Tamilarasan, and Caverlee 2013 furthered this line of research by proposing a machine learning framework to detect workers on crowdsourcing sites that may facilitate crowdturfing operations. Other researchers try to find evidence of crowdturfing reviews by analyzing the text of the social media sites (Ott, Cardie, and J. Hancock 2012; Ott, Cardie, and J. T. Hancock 2013). While these are close to our problem, the key difference is that these algorithms focus on review spam and the nature of the unskilled workers in order to make these classifications. Our problem is different because these shills are highly trained and provided with diverse “talking points” when generating their own content.

Another related task to shill detection is bot detection. Bot detection, shill detection, and crowdturfing all involve the use of a “controller” who directs the content produced by the controllee (Conover et al. 2011), which is a similar structure to the behavior of shills. Furthermore, bots are able to manipulate the discussion on a site by mass-posting on certain topics in order to get those topics to trend on the social networking site (Ferrara, Varol, Davis, et al. 2016). Furthermore, other work in bot detection uses similar feature sets such as language (Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011; Thonnard and Dacier 2011), and the user’s network properties (Zafarani and Liu 2015; Thomas, Grier, and Paxson 2012). Another area where our work dovetails with bot detection is ground truth curation. Both our project and bot detection have the problem that a user can rarely definitively be

proven to be a in the positive class. Instead, both problems rely on heuristics to yield ground truth. Heuristics used in bot detection commonly include manual annotation of user accounts (Chu et al. 2010; Cook et al. 2014), observing the social sites to see which users the site bans for bot behavior (John et al. 2009; Lee and Kim 2014), and creating honeypots (Lee, Eoff, and Caverlee 2011; Morstatter, Wu, et al. 2016) to lure bots into following them. We utilize the first approach as it is the most feasible to bot detection. We train the users using the set of criteria described previously, and ask them to label the data.

Our study can be compared with studies on the political nature of social media. The detection of shills is similar to the identification of central political activists (Morales et al. 2015) in that both partially rely on the content expressed by key individuals in topics pertaining to their political representatives’ respective agendas (Ranganath et al. 2016). Although we do indicate that the responsibility of influencing others lies in a few key individuals, we are not studying polarization of political opinions (Morales et al. 2015). Furthermore, our end goals differ in the specificity. While most try to identify promoters on social media (Li et al. 2014), promoters are a different type of user in the sense that they do not hide their identity to be seen as regular users. More recently, Ferrara, Varol, Menczer, et al. 2016 propose a dynamic time warping algorithm to detect users behaving in tandem on a social media site.

Our goals differ from other politically-motivated studies of social media. For example, Morales et al. 2015 exploits the fact that political polarization tends to form around those with extreme opinions on political issues and created a model with the purpose of identifying these individuals. Our work differs from theirs in that although both project are trying to find key users in the center of political conversations, we are only trying to find paid political advocates out of these users. Another similarity



with other existing projects is the concept of finding “hidden” political activists. For example, Li et al. 2014 find hidden campaigns on social media. A key difference when is the target audience. Another major difference lies in the approach: Li et al. 2014 heavily leveraged the network in their approach. Since Reddit does not have a network, we are unable to apply this approach in our classification and instead rely heavily upon natural language processing tools when extracting features.

There exist other political analysis projects that targeted a variety of audience groups due to their holistic nature (Conover et al. 2011). In contrast, our targeted audience group consists of individuals who are interested in the subject after SuperPAC announced its involvement. While our purpose does not involve scoring each individual on a set scale as possible skill, the process of denoting an user as a skill does slightly involve such rating, which is heavily used in Wong et al. 2013.

## 2.7 Perceptual Bias in Social Data

There are hundreds of different cognitive biases recognized by the social sciences. Many of these are catalogued in Baron 2000, and a very comprehensive list can be found on the “List of cognitive biases” Wikipedia page.<sup>13</sup>

In this section we discuss the related work as it pertains to perceptual bias from three different perspectives. First, since our solution heavily relies on word embeddings to create the emoji mapping, we enumerate some recent work on word embeddings. Next, we discuss other resources for informal text and continue to discuss other work that has been done on emoji analysis. Finally, we discuss other work that has been done in the context of platform-specific emoji rendering.

---

<sup>13</sup>[https://en.wikipedia.org/wiki/List\\_of\\_cognitive\\_biases](https://en.wikipedia.org/wiki/List_of_cognitive_biases)

### 2.7.1 Developing An Emoji Mapping with Word Embeddings

One of the first word embedding algorithms was the Neural Network Language Model (Bengio et al. 2006). Currently, one of the most famous word embedding algorithms, Word2Vec (Mikolov et al. 2013), has risen to prominence. This algorithm works embedding words that appear next to each other in the text next to each other in the embedding. Word2Vec provides two approaches to solving this problem: the “continuous bag of words” (CBOW) and the “skip-gram” (SG) architecture. While skip-gram has been shown to better solve analogies (Mikolov, Yih, and Zweig 2013), we compare with both approaches.

The simple rules combined with other constraints has been shown to create powerful embeddings that work in many different settings. For example, they have been used to produce state-of-the-art performance in the areas of syntactic parsing (Socher et al. 2013), named entity recognition (Dhillon, Foster, and Ungar 2011), antonym detection (Ono, Miwa, and Sasaki 2015), sentiment analysis (Maas et al. 2011; Tang, Wei, Qin, et al. 2014; Tang, Wei, Yang, et al. 2014), and machine translation (Zou et al. 2013). Furthermore, researchers have tried to go beyond using vectors to model just words, and have used them to model sentences, paragraphs, and documents (Le and Mikolov 2014).

### 2.7.2 Linguistic Resources for Informal Text

Before emojis were commonplace, there has been a long history of trying to better represent and understand text, both formal and informal. One of the most influential resources is WordNet (Miller 1995), which represents words not only by

their definitions, but also provides a graph of the relationship between the words. Extensions to this approach abound, but perhaps the most relevant one to our work is SentiWordNet (Baccianella, Esuli, and Sebastiani 2010), which considers the sentiment of the words when building the resource. In the context of informal text, SlangSD (Wu, Morstatter, and Liu 2016) provides a sentiment resource for mapping slang words to sentiment scores by leveraging Urban Dictionary’s data. Crowdsourcing has been used to extract the emotional meanings for words (Mohammad and Turney 2013).

### 2.7.3 Emoji Analysis

As emojis have become an important tool that help people communicate and express their emotions, the study of emojis as they pertain to sentiment classification and text understanding is attracting attention (Barbieri et al. 2016; Barbieri, Ronzano, and Saggion 2016; Eisner et al. 2016; Hallsmar and Palm 2016; Hu et al. 2013; Kelly and Watts 2015; Novak et al. 2015). Hu et al. 2013 proposes a unsupervised framework for sentiment classification by incorporating emoticon signals. Hallsmar and Palm 2016 investigates the feasibility of an emoji training heuristic for multi-class sentiment analysis on Twitter with a Multinomial Naive Bayes Classifier. Eisner *et al.* learn emoji representation by running Skip-gram on descriptions of emojis provided in the Unicode standard. The resulting emoji representation along with the word embedding from Google News are used to perform sentiment analysis and the results show that emoji representation can improve sentiment analysis. Instead of using Unicode description, Barbieri, Ronzano, and Saggion 2016 learns the vector skip-gram model for twitter emojis using tweets, which also demonstrate the ability of Emojis in improving sentiment analysis. Novak et al. 2015 and Barbieri et al. 2016 analyze

the sentiment of emojis with respect to tweet corpus of different languages, position of emojis in text, etc. They find that there is no significant difference for emoji usage in different *languages*, and people are more likely to express positive sentiments when putting emojis at the end of their text. We build on this work by studying the difference for emoji usage in different *platforms*. Kelly *et al.* shows that emojis can be used as appropriations which can help facilitate communications using interview data. Cappallo, Mensink, and Snoek 2015 propose a framework called *Image2Emoji*, which combines visual and text information to guide the process of emoji scoring. Lu et al. 2016 explores the ubiquitous usage pattern of emojis by people from different cultures. They find that theses pattern can be applied to improve user experience qualities for input methods and understand user preferences. Wijeratne et al. 2016 use emojis to build a sense inventory.

#### 2.7.4 Platform Specific Emoji Rendering

The aforementioned studies on emoji analysis ignore the fact that the same emoji unicode has different emoji images on different platform. Thus, the sentiment or semantic meanings of the same emoji may be perceived differently for people using different platforms and thus cause misunderstanding. Therefore, recently, there are researchers paying attention to this issue (Miller et al. 2016; Tigwell and Flatla 2016). Miller et al. 2016 show that emoji misinterpretation exists within and across platforms, from both semantic and sentiment perspectives. The analysis is based on a survey to collect people’s feedback of sentiment scores and semantic meaning on different rendering of emojis, which does not consider the context of emojis. Similarly, Tigwell and Flatla 2016 also explore platform-dependent emoji misinterpretation problem

in <http://on.wsj.com/1Qo215n>. They design a questionnaire to collect user’s sentiment feedback on 16 emojis from Android and iOS platform, and compute a valence-arousal space to guide sentiment analysis. Different from existing approaches exploring platform specific emoji rendering problems, we use real world data to verify the existence of emoji ambiguity and provides a mapping-based solution to identify the intended emoji from original posts.

### DETECTING SOCIAL DATA COLLECTION BIAS

When we use social media data, we want the data we collect to be representative of the problem we wish to study. Unfortunately, this is often not the case. As we will demonstrate in this chapter, there are many ways in which the data we collect from Twitter is not representative of Twitter itself, let alone any singular problem we wish to study.

Because Twitter is so prominent for research, we need to make sure that the data collected by researchers is “fit for purpose.” Social media data is often difficult to obtain, with most social media sites restricting access to their data. Twitter’s policies lie opposite to this. The “Twitter Streaming API”<sup>14</sup> is a capability provided by Twitter that allows anyone to retrieve at most a 1% sample of all the data by providing some parameters. According to the documentation, the sample will return at most 1% of all the tweets produced on Twitter at a given time. Once the number of tweets matching the given parameters eclipses 1% of all the tweets on Twitter, Twitter will begin to sample the data returned to the user. The methods that Twitter employs to sample this data is currently unknown. The Streaming API takes three parameters: keywords (words, phrases, or hashtags), geographical boundary boxes, and user ID.

One way to overcome the 1% limitation is to use the Twitter Firehose—a feed provided by Twitter that allows access to 100% of all public tweets. A very substantial drawback of the Firehose data is the restrictive cost. Another drawback is the sheer

---

Portions of this chapter have been published at ICWSM 2013 (Morstatter et al. 2013).

<sup>14</sup><https://dev.twitter.com/docs/streaming-apis>



### 3.1 A Dataset for Detecting Bias

From December 14th, 2011 - January 10th, 2012 we collected tweets from the Twitter Firehose matching any of the keywords, geographical bounding boxes, and users in Table 1. During the same time period, we collected tweets from the Streaming API using TweetTracker (S Kumar et al. 2011) with exactly the same parameters. During the time we collected 528,592 tweets from the Streaming API and 1,280,344 tweets from the Firehose. The raw counts of tweets we received each day from both sources are shown in Figure 6. One of the more interesting results in this dataset is that as the data in the Firehose spikes, the Streaming API coverage is reduced. One possible explanation for this phenomenon could be that due to the Western holidays observed at this time, activity on Twitter may have reduced causing the 1% threshold to go down.

One of the key questions we ask is how the amount of coverage affects measures commonly performed on Twitter data. Here we define coverage as the ratio of data from the Streaming API to data from the Firehose. To better understand the coverage of the Streaming API for each day, we construct a box-and-whisker plot to visualize the distribution of daily coverage, shown in Figure 7. In this period of time the Streaming API receives, on average, 43.5% of the data available on the Firehose on any given day. While this is much better than just 1% of the tweets promised by the Streaming API, we have no reference point for the data in the tweets we received.

The most striking observation is the range of coverage rates (see Figure 7). Increase of *absolute* importance (more global awareness) or *relative* importance (the overall number of tweets decreases) result in lower coverage as well as fewer tweets. To give



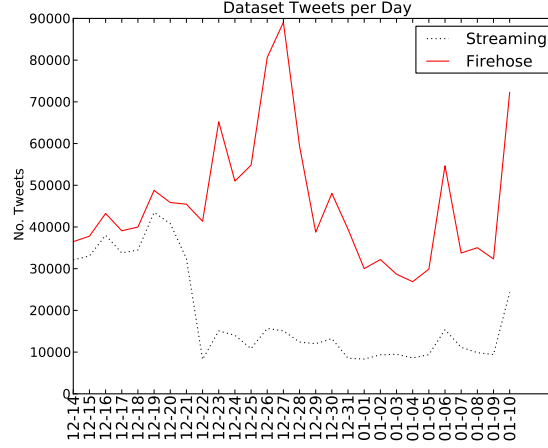



Figure 6: Raw Tweet Counts for Each Day from Both the Streaming API and the Firehose.

Table 1: Parameters Used to Collect Data from Syria. Coordinates Below the Boundary Box Indicate the Southwest and Northeast Corner, Respectively. The Last Keyword Is the Arabic Word for “Syria.”

Keywords	Geoboxes	Users
#syria, #assad, #alep- povolcano, #alawite, #homs, #hama, #tar- tous, #idlib, #damas- cus, #daraa, #aleppo, #houla, #سوريا	 (32.8, 35.9), (37.3, 42.3)	@SyrianRevo

the reader a sense for the top words in both datasets, we include tag clouds for the top words in the Streaming API and the Firehose, shown in Figure 5.

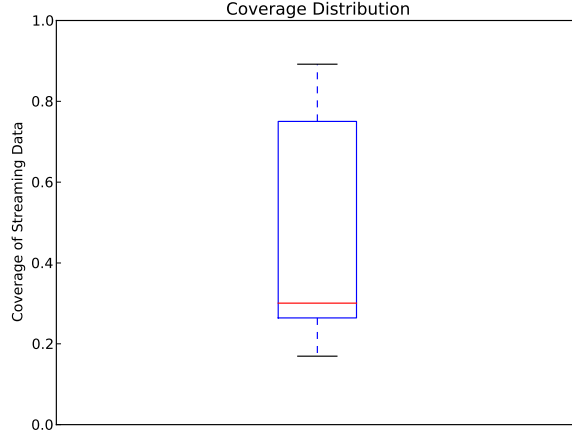


Figure 7: Distribution of Coverage for the Streaming Data by Day. Whiskers Indicate Extreme Values.

### 3.2 Statistical Measures

We investigate the statistical properties of the two datasets with the intent of understanding how well the characteristics of the sampled data match those of the Firehose. We begin first by comparing the top hashtags in the tweets for different levels of coverage using a rank correlation statistic. We continue to extract topics from the text, matching topical content and comparing topical distribution to better understand how sampling affects the results of this common process performed on Twitter data. In both cases we compare our streaming data to random datasets obtained by sampling the data obtained through the Firehose.

#### 3.2.1 Top Hashtag Analysis

Hashtags are an important communication device on Twitter. Users employ them to annotate the content they produce, allowing for other users to find their tweets and to facilitate interaction on the platform. Also, adding a hashtag to a tweet is

equivalent to joining a community of users discussing the same topic (Yang et al. 2012). In addition, hashtags are also used by Twitter to calculate the trending topics of the day, which encourages the user to post in these communities.

Recently, hashtags have become an important part of Twitter analysis (M. Efron 2010; Tsur and Rappoport 2012; Recuero and Araujo 2012). For both the purpose of community formation and trend analysis it is important that our Streaming dataset convey the same importance for hashtags as the Firehose data. Here we compare the top hashtags in the two datasets using Kendall’s  $\tau$  rank correlation coefficient (Agresti 2010).

#### Kendall’s $\tau$ of Top Hashtags

Kendall’s  $\tau$  is a statistic which measures the correlation of two ordered lists by analyzing the number of concordant pairs between them. Consider two hashtags, #A and #B. If both lists rank #A higher than #B, then this is considered a concordant pair, otherwise it is counted as a discordant pair. Ties are handled using the  $\tau_\beta$  statistic as follows:

$$\tau_\beta = \frac{|P_C| - |P_D|}{\sqrt{(|P_C| + |P_D| + |T_F|)(|P_C| + |P_D| + |T_S|)}} \quad (3.1)$$

where  $P_C$  is the set of concordant pairs,  $P_D$  is the set of discordant pairs,  $T_F$  is the set of ties in the Firehose data, but not in the Streaming data,  $T_S$  is the number of ties found in the Streaming data, but not in the Firehose, and  $n$  is the number of pairs in total. The  $\tau_\beta$  value ranges from -1, perfect negative correlation, to 1, perfect positive correlation.

To understand the relationship between  $n$  and the resulting correlation,  $\tau_\beta$ , we construct a chart showing the value of  $\tau_\beta$  for  $n$  between 10 and 1000 in steps of 10.

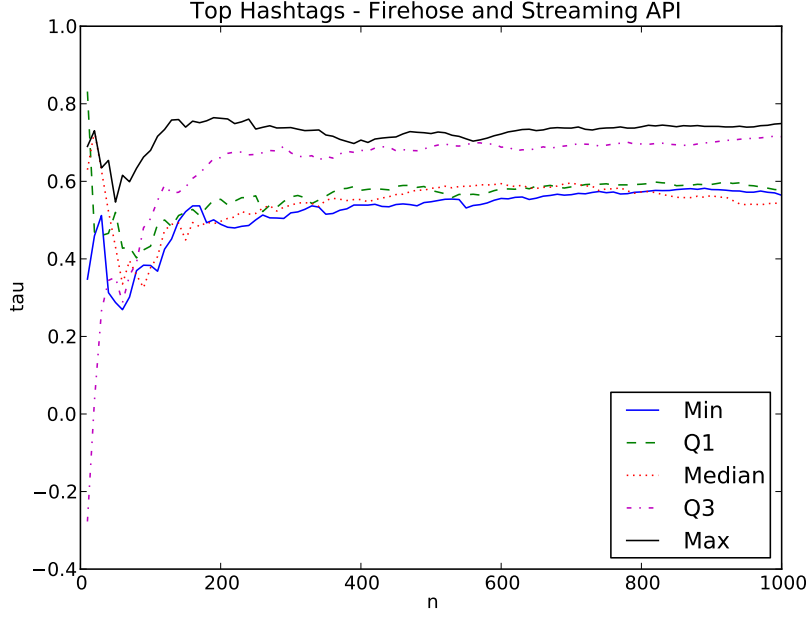


Figure 8: Relationship between  $n$  - Number of Top Hashtags, and the Correlation Coefficient,  $\tau_\beta$ .

To get an accurate representation of the differences in correlation at each level of Streaming coverage, we select five days with different levels of coverage as motivated by Figure 7: The minimum (December 27th), lower quartile (December 24th), median (December 29th), upper quartile (December 18th), and the maximum (December 19th). The results of this experiment are shown in Figure 8. Here we see mixed results at small values of  $n$ , indicating that the Streaming data may not be good for finding the top hashtags. At larger values of  $n$ , we see that the Streaming API does a better job of estimating the top hashtags in the Firehose data.

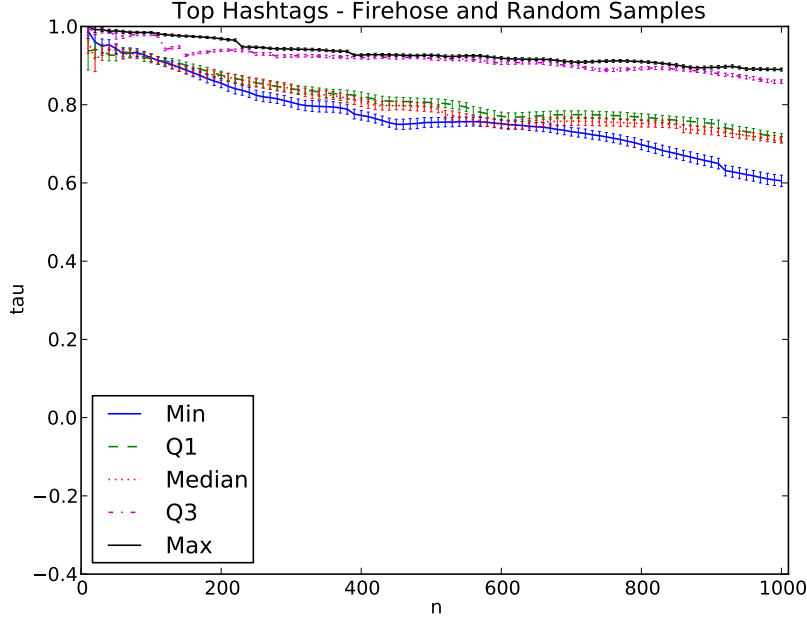


Figure 9: Random Sampling of Firehose Data. Relationship between  $n$  - Number of Top Hashtags, and  $\tau_\beta$  - The Correlation Coefficient for Different Levels of Coverage.

### Comparison with Random Samples

After seeing the results from the previous section, we are left to wonder if the results are an artifact of using the Streaming API or if we could have obtained the same results by any random sampling. Would we obtain the same results with a random sample of equal size from the Firehose data, or does the Streaming API's filtering mechanism give us an advantage? To answer this question we repeat the experiments for each day in the previous section. This time, instead of using Streaming API data, we select tweets uniformly at random (without replacement) until we have amassed the same number of tweets as we collected from the Streaming API for that day. We repeat this process 100 times and obtain results as shown in Figure 9. Here we see that the levels of coverage in the random and Streaming data have comparable  $\tau_\beta$  values for large  $n$ , however at smaller  $n$  we see a much different picture. The

random data gets very high  $\tau_\beta$  scores for  $n = 10$ , showing a good capacity for finding the top hashtags in the dataset. The Streaming API data does not consistently find the top hashtags, in some cases revealing reverse correlation with the Firehose data at smaller  $n$ . This could be indicative of a filtering process in Twitter’s Streaming API which causes a misrepresentation of top hashtags in the data.

### 3.2.2 Topic Analysis

Topic models are statistical models which discover topics in a corpus. Topic modeling is especially useful in large data, where it is too cumbersome to extract the topics manually. Due to the large volume of tweets published on Twitter, topic modeling has become central to many content-based studies using Twitter data (Kireyev, Palen, and Anderson 2009; Pozdnoukhov and Kaiser 2011; Hong et al. 2012; Yin et al. 2011; Chae et al. 2012). We compare the topics drawn from the Streaming data with those drawn from the Firehose data using a widely-used topic modeling algorithm, latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003). Latent Dirichlet allocation is an algorithm for the automated discovery of topics. LDA treats documents as a mixture of topics, and topics as a mixture of words. Each topic discovered by LDA is represented by a probability distribution which conveys the affinity for a given word to that particular topic. We analyze these distributions to understand the differences between the topics discovered in the two datasets. To get a sense of how the topics found in the Streaming data compare with those found with random samples, we compare with topics found by running LDA on random subsamples of the Firehose data.

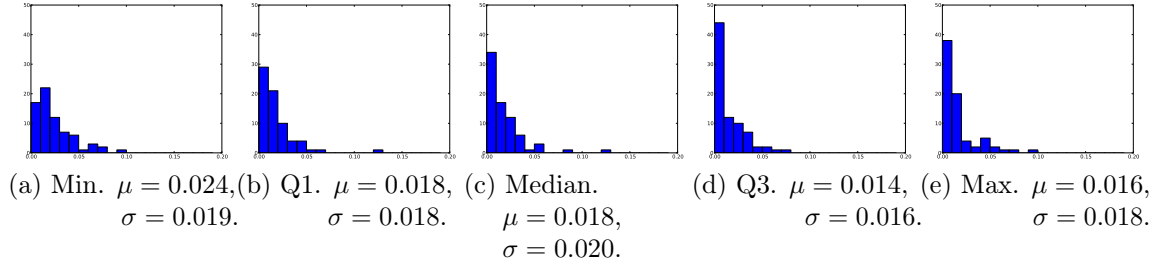


Figure 10: The Jensen-Shannon Divergence of the Matched Topics at Different Levels of Coverage. The X-Axis Is the Binned Divergence. No Divergence Was  $> 0.15$ . The Y-Axis Is the Count of each Bin.  $\mu$  Is the Average Divergence of the Matched Topics,  $\sigma$  Is the Standard Deviation.

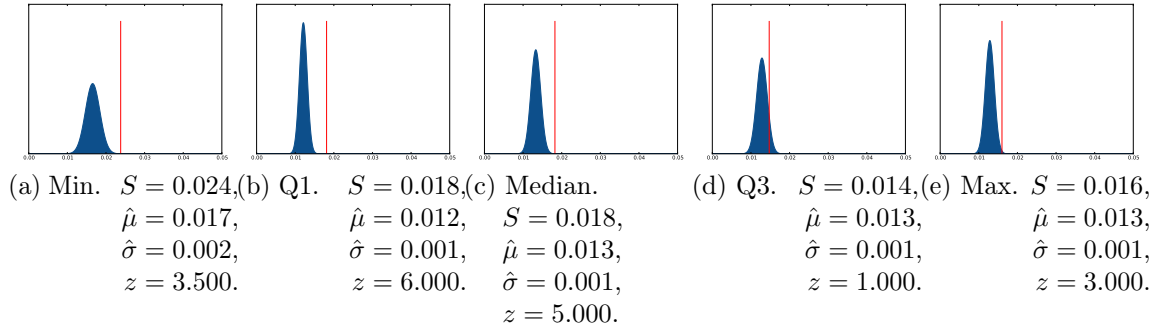


Figure 11: The Distribution of Average Jensen-Shannon Divergences in the Random Data (Blue Curve), with the Single Average Obtained through the Streaming Data (Red, Vertical Line).  $Z$  Indicates the Number of Standard Deviations the Streaming Data Is from the Mean of the Random Samples.

## Topic Discovery

Here we compare the topics generated using the Firehose corpus with those generated using the Streaming corpus. LDA takes, in addition to the corpus, three parameters as its input:  $K$  - the number of topics,  $\alpha$  - a hyperparameter for the Dirichlet prior topic distribution, and  $\eta$  - a hyperparameter for the Dirichlet prior word distribution. Choosing optimal parameters is a very challenging problem, and is

not our focus. Instead we focus on the similarity of the results given by LDA using identical parameters on both the Streaming and Firehose corpus. We set  $K = 100$  as suggested by Dumais et al. 1988 and use priors of  $\alpha = 50/K$ , and  $\eta = 0.01$ . The software we used to discover the topics is the *gensim* software package (Rehurek and Sojka 2010). We select the same days as we did for the comparison of Kendall’s  $\tau$ .

## Topic Comparison

To understand the differences between the topics generated by LDA, we compute the distance in their probability distribution using the Jensen-Shannon divergence metric (Lin, Jan). Since LDA’s topics have no implicit orderings we first must match them based upon the similarity of the words in the distribution. To do the matching we construct a weighted bipartite graph between the topics from the Streaming API and the Firehose. Treating each topic as a bag of words, we use the Jaccard score between the words in a Streaming topic  $T_i^S$  and a Firehose topic  $T_j^F$  as the weight of the edges in the graph,

$$d(T_i^S, T_j^F) = \frac{|T_i^S \cap T_j^F|}{|T_i^S \cup T_j^F|}. \quad (3.2)$$

After constructing the graph we use the maximum weight matching algorithm proposed in Galil 1986 to find the best matches between topics from the Streaming and Firehose data. After making the ideal matches, we then compute the Jensen-Shannon divergence between the two topics. Treating each topic as a probability distribution, we compute this as follows:

$$JS(T_i^S || T_j^F) = \frac{1}{2}[KL(T_i^S || M) + KL(T_j^F || M)], \quad (3.3)$$

where  $M = \frac{1}{2}(T_i^S + T_j^F)$  and  $KL$  is the Kullback-Liebler divergence (Cover and Thomas 2006). We compute the Jensen-Shannon divergence for each matched pair and plot



a histogram of the values in Figure 10. We see a trend of higher divergence with lower coverage, and lower divergence with higher coverage. This shows that decreased coverage in the Streaming data causes variance in the discovered topics.

### Comparison with Random Samples

In order to get additional perspective on the accuracy of the topics discovered in the Streaming data, we compare the Streaming data with data sampled randomly from the Firehose, as we did earlier to compare the correlation. First, we compute the average of the Jensen-Shannon scores from the Streaming data in Figure 10,  $S$ . We then repeat this process for each of the 100 runs with random data, each run called  $x_i$ . Next, we use maximum-likelihood estimation (Casella and Berger 2001) to estimate the parameters of the Gaussian distribution from which these points originate,  $\hat{\mu} = \frac{1}{100} \sum_{i=1}^{100} x_i$ , and  $\hat{\sigma} = \sqrt{\frac{1}{100} \sum_{i=1}^{100} (x_i - \hat{\mu})^2}$ . Finally, we compute the  $z$ -Score for  $S$ ,  $z = \frac{S - \hat{\mu}}{\hat{\sigma}}$ . This score gives us a concrete measure of the difference between the Streaming API data and the random samples. Results of this experiment, including  $z$ -Scores are shown in Figure 11. Nonetheless, we are still able to get topics from the Streaming API that are close to those found in random data with higher levels of coverage. A threshold of *3-sigma* is often used in the literature to indicate extreme values Guthrie 2010, Section 6.3.1. With this threshold, we see that overall we are able to get significantly better topics with the random data than with the Streaming API on 4 of the 5 days.

### 3.3 Network Measures

Because Twitter is a social network, Twitter data can be analyzed with methods from Social Network Analysis (Wasserman and Faust 1994) in addition to statistical measures. Possible 1-mode and 2-mode networks are:  $User \times User$  retweet networks,  $User \times Hashtag$  content networks,  $Hashtag \times Hashtag$  co-occurrence networks. For the purpose of this article we focus on  $User \times User$  retweet networks. Users who send tweets within a certain time period are the nodes in the network. Furthermore, users that are retweeted within this time period are also nodes in this network, regardless of the time their original tweet was tweeted. The networks created by this procedure are directed and not symmetric by design, however, bi-directional links are possible in case  $a \rightarrow b$  and  $b \rightarrow a$ . We ignore line weight created by multiple  $a \rightarrow b$  retweets and self-loops (yes, some user retweet themselves). For the network metrics, the comparison is done on both the network, and the node levels. Networks are analyzed using ORA (Carley et al. 2012).

#### 3.3.1 Node-Level Measures

The node-level comparison is accomplished by calculating measures at the user-level and comparing these results. We calculate three different *centrality measures* at the node level, two of which—Degree Centrality and Betweenness Centrality—were defined by Freeman as “distinct intuitive conceptions of centrality” (Freeman 1979, p. 215). Degree Centrality counts the number of neighbors in unweighted networks. In particular, we are interested in In-Degree Centrality as this reveals highly respected sources of information in the retweet network (where directed edges point to the

Table 2: Average Centrality Measures for Twitter Retweet Networks for 28 Daily Networks. “All” Is All 28 Days Together.

Measure	$k =$	Top- $k$ ( <i>min-max</i> )	All
In-Degree	10	4.21 (0–9)	4
In-Degree	100	53.4 (36–82)	73
Potential Reach	100	59.2 (32–83)	80
Betweenness	100	54.8 (41–81)	55

source). Betweenness Centrality (Freeman 1979) identifies brokerage positions in the Twitter networks that connect different communities with each other or funnel different information sources. Furthermore, we calculate the *Potential Reach* which counts the number of nodes that are reachable in the network weighted with the path distance. In our Twitter networks this is equivalent to the inverse in-distance of reachable nodes (Sabidussi 1966). This approach results in a metric that finds sources of information (users) that potentially can reach many other nodes on short path distances. Before calculating these measures, we extract the main component and delete all other nodes (see next sub-section). In general, centrality measures are used to identify important nodes. Therefore, we calculate the number of top 10 and top 100 nodes that can be correctly identified with the Streaming data. Table 2 shows the results for the average of 28 daily networks, the *min-max* range, as well as the aggregated network including *all* 28 days.

Although, we know from previous studies (Borgatti, Carley, and Krackhardt 2006) that there is a very low likelihood that the ranking will be correct when handling networks with missing data, the accuracy of the daily results is not very satisfying.

When we look at the results of the individual days, we can see that the matches have, once again, a broad range as a function of the data coverage rate. In Borgatti, Carley, and Krackhardt 2006 the authors argue that network measures are stable for denser networks. Twitter data, being very sparse, causes the network metrics’ accuracy to be rather low in the case when the data sub-sample is smaller. However, identifying  $\sim 50\%$  key-players correctly for a single day is reasonable, and accuracy can be increased by using longer observation periods. Even more, the Potential Reach metrics are quite stable for some days in the aggregated data.

### 3.3.2 Network-Level Measures

We complement our node-level analysis by comparing various metrics at the network level. These metrics are reported in Table 3 and are calculated as follows. Since retweet networks create a lot of small disconnected components, we focus only on the size of the largest component. The size of the main component and the fact that all smaller components contain less than 1% of the nodes justify our focus on the main component for this data. Therefore, we reduce the networks to their largest component before we proceed with the calculations. To describe the structure of the retweet networks we calculate the clustering coefficient, a measure for local density (Watts and Strogatz 1998). We do not take all possible triads of directed networks into account, but treat the networks as undirected when calculating the clustering coefficient.  $D_{in} > 0$  shows the proportion of nodes in the largest component that are retweeted and  $\max(D_{in})$  shows the value of the highest unscaled In-Degree value, i.e., number of unique users retweeting the same single user. The final three lines of Table 3 are network centralization indexes based on the node-level measures

that have been introduced in the previous paragraph. Freeman 1979 describes the centralization  $C_X$  of a network for any given metric as the difference of the value  $C_X(p^*)$  of the most central node to all other node values compared to the maximum possible difference:

$$C_X = \frac{\sum i = 1n[C_X(p^*) - C_X(p_i)]}{\max \sum i = 1n[C_X(p^*) - C_X(p_i)]} \quad (3.4)$$

High centralization indicates a network with some nodes having very high node-level values and many nodes with low values while low centralization is the result of evenly distributed node-level measures.

We do not discuss all details of the individual results but focus on the differences between the two data sources. First, the coverage of nodes and links is similar to the coverage of tweets. This is a good indicator that the sub-sample is not biased to the specific Twitter user (e.g. high activity). The smaller proportion of nodes with non-zero In-Degree for the Firehose shows us that the larger number of nodes includes many more peripheral nodes. A low Clustering Coefficient implies that networks are hierarchical rather than interacting communities. Even though the centralization indexes are rather similar, there is one very interesting result when looking at the individual days: The range of values is much higher for the Streaming data as a result of the high coverage fluctuation. Further research will analyze whether we can use network metrics to better estimate how sufficient the sampled Streaming data is.

### 3.4 Geographic Measures

The final facet of the Twitter data we compare is the geolocation of the tweets. Geolocation is an important part of a tweet, and the study of the location of content

Table 3: Comparison of Network-Level Social Network Analysis Metrics.

Metrics	Firehose		Streaming API	
	avg.day	28 days	avg.day	28 days
nodes	6,590	73,719	2,466 (37.4%)	30,894 (41.9%)
links	10,173	204,022	3,667 (36.0%)	76,750 (37.6%)
$D_{in} > 0$	25.1%	19.3%	32.4%	20.5%
$max(D_{in})$	341	2,956	167.3	1,252
main comp.	5,609	70,383	2,069	28,701
main comp. %	84.6%	95.5%	82.5%	92.9%
Clust.Coeff.	0.029	0.053	0.033	0.050
$DC_{in}$ Centr.	0.059	0.042	0.085	0.043
$BC$ Centr.	0.010	0.053	0.010	0.050
$PReach$ Centr.	0.130	0.240	0.156	0.205

and users is currently an active area of research (Cheng, Caverlee, and Lee 2010; Wakamiya, Lee, and Sumiya 2011). We study how the geographic distribution of the geolocated tweets is affected by the sampling performed by the Streaming API.

The number of geotagged tweets is low, with only 16,739 geotagged tweets in the Streaming data (3.17%) and 18,579 in the Firehose data (1.45%). We notice that despite the difference in tweets collected on the whole we get 90.10% coverage of geotagged tweets. We start by grouping the locations of tweets by continent and can find a strong Asian bias due to the boundary box we used to collect the data from both sources, shown in Table 1. To better understand the distribution of geotagged tweets we repeat the same process, this time excluding tweets originating in the boundary box set in the parameters. After removing these tweets, more than 90% of geotagged Tweets from both sources are excluded from the data and the Streaming coverage

Table 4: Geotagged Tweet Location by Continent. Excluding Boundary Box from Parameters.

Continent	Firehose	Streaming	Error
Africa	156 (5.74%)	33 (3.10%)	-2.64%
Antarctica	0 (0.00%)	0 (0.00%)	$\pm 0.00\%$
Asia	932 (34.26%)	321 (30.11%)	-4.15%
Europe	300 (11.03%)	139 (13.04%)	+2.01%
Mid-Ocean	765 (28.12%)	295 (27.67%)	-0.45%
N. America	607 (22.32%)	293 (27.49%)	+5.17%
Oceania	54 (1.98%)	15 (1.41%)	-0.57%
S. America	3 (0.11%)	2 (0.19%)	+0.08%
Total	2720 (100.00%)	1066 (100.00%)	$\pm 0.00\%$

level is reduced to 39.19%. The distribution of tweets by continent is shown in Table 4. Here we see a more even representation of the tweets’ locations in Asia and North America.

### 3.5 Discussion

In this chapter we asked whether data obtained through Twitter’s sampled Streaming API is a sufficient representation of activity on Twitter as a whole. To answer this question we collected data with exactly the same parameters from both the free, but limited, Streaming API and the unlimited, but costly, Firehose. We provide a methodology for comparing the two multifaceted sets of data and results of our analysis.

We started our analysis by understanding the coverage of the Streaming API data, finding that when the number of tweets matching the set of parameters increases, the Streaming API’s coverage is reduced. One way to mitigate this might be to create

more specific parameter sets with different users, bounding boxes, and keywords. This way we might be able to extract more data from the Streaming API.

Next, we studied the statistical differences between the two datasets. We used a common correlation coefficient to understand the differences between the top  $n$  hashtags in the two datasets. We find that the Streaming API data estimates the top hashtags for a large  $n$  well, but is often misleading when  $n$  is small. We also employed LDA to extract topics from the text. We compare the probability distribution of the words from the most closely-matched topics and find that they are most similar when the coverage of the Streaming API is greatest. That is, topical analysis is most accurate when we get more data from the Streaming API.

The Streaming API provides just one example of how sampling Twitter data affects measures. We leverage the Firehose data to get additional samples to better understand the results from the Streaming API. In both of the above experiments we compare the Streaming data with 100 datasets sampled randomly from the Firehose data. We compare the statistical properties to find that the Streaming API performs worse than randomly sampled data, especially at low coverage. We find that in the case of top hashtag analysis, the Streaming API sometimes reveals negative correlation in the top hashtags, while the randomly sampled data exhibits very high positive correlation with the Firehose data. In the case of LDA we find a significant increase in the accuracy of LDA with the randomly sampled data over the data from the Streaming API. Both of these results indicate some bias in the way that the Streaming API provides data to the user.

By analyzing retweet  $User \times User$  networks we were able to show that we can identify, on average, 50–60% of the top 100 key-players when creating the networks based on one day of Streaming API data. Aggregating some days of data can increase



the accuracy substantially. For network level measures, first in-depth analysis revealed interesting correlation between network centralization indexes and the proportion of data covered by the Streaming API.

Finally, we inspect the properties of the geotagged tweets from both sources. Surprisingly, we find that the Streaming API almost returns the complete set of the geotagged tweets despite sampling. We attribute this to the geographic boundary box. Although the number of geotagged tweets is still very small in general ( $\sim 1\%$ ), researchers using this information can be confident that they work with an almost complete sample of Twitter data when geographic boundary boxes are used for data collection. When we remove the tweets collected this way, we see a much larger disparity in the tweets from both datasets. Even with this disparity, we see a similar distribution based on continent.

Overall, we find that the results of using the Streaming API depend strongly on the coverage and the type of analysis that the researcher wishes to perform. This leads to the next question concerning the estimation of how much data we actually get in a certain time period. We suggest that we found first evidence in different types of analysis that can help us to estimate the Streaming API coverage. Uncovering the nuances of the Streaming API will help researchers, business analysts, and governmental institutions to better ground their scientific results based on Twitter data.

Looking forward, we hope to find methods to compensate for the biases in the Streaming API to provide a more accurate picture of Twitter activity to researchers. Provided further access to Twitter’s Firehose, we will determine whether the methodology presented here will yield similar results for Twitter data collected from other domains, such as natural disaster, protest, and elections.

### ASSESSING AND MITIGATING SOCIAL DATA COLLECTION BIAS

In this chapter we introduce two approaches to addressing the problem of overcoming social data bias. The first is based upon identifying bias in a previously-collected social media dataset. The second is based upon collecting social media data in such a way that the resulting dataset contains less data collection bias than it would have if the data had been collected in another way.

#### 4.1 Assessing Bias in Previously Collected Datasets

Twitter is a microblogging site where users share short, 140-character messages called “tweets”. Twitter has become one of the largest social networking sites in the world with 240 million users who publish 500 million tweets each day<sup>15</sup> from their web browsers and mobile phones. Due to Twitter’s massive size and ease of mobile publication, Twitter has also become a central tool for communication during protests (Campbell 2011) and disasters (*Humanitarianism in the Network Age* 2012). This has caused an immense push from both the computer and social science research communities who have used data from the site for wide applications of social media research from predicting users’ location (Cheng, Caverlee, and Lee 2010) or to characterize the life cycle of news stories (Castillo et al. 2014).

---

Portions of this chapter have been published at WWW 2014 (Morstatter, Pfeffer, and Liu 2014) and Hypertext 2015 (Sampson et al. 2015).

<sup>15</sup><https://blog.twitter.com/2013/new-tweets-per-second-record-and-how>

One way to get around the 1% limit is to purchase the Twitter Firehose, a feed offered by Twitter that allows for access to 100% of all of the public tweets posted on the site. Simply comparing the results from the Streaming API with the Firehose is one way to verify the results of the sampled service. This is the approach taken in Morstatter et al. 2013. Unfortunately, verifying results using the Firehose is not an option for most researchers as access to Twitter’s Firehose is restrictively expensive and access is limited to users with a business agreement with Twitter<sup>16</sup>.

In this chapter, the word “bias” only pertains to sample bias. We say that a hashtag is “biased” if the relative trend is statistically significantly over-represented or underrepresented in contrast to its true trend on Twitter. In particular, we are looking for particular time periods of bias in the Streaming API. Based on Twitter’s documentation, the sample size is determined by the volume of a query at a point in time. There are times when the sample is representative, and times when it is biased. We try to find time periods where the data from the Streaming API is biased, meaning not an accurate representation of the true activity on Twitter.

The focus of this chapter lies in finding and assessing an alternative method for bias detection in the Streaming API. Our goal is to detect the bias automatically, using methods that are openly available to researchers. We begin this process by discussing the related work, which includes a discussion of other work that discovers bias in the Streaming API. We continue to introduce and vet another data source, the Sample API<sup>17</sup>, and propose a methodology that utilizes this data source to show a user when there is bias in the results of their Streaming API query. Finally, we show that the Streaming API gives nearly the same results to identical queries originating

---

<sup>16</sup><https://dev.twitter.com/discussions/2752>

<sup>17</sup><https://dev.twitter.com/docs/api/1/get/statuses/sample>

from different points around the world, and to identical queries started at different points in time during the overlap of their execution.

#### 4.1.1 Discovering Bias in the Streaming API without the Firehose

With work showing evidence that the Streaming API is biased, researchers must be able to tell whether their Streaming API sample is biased. Vetting their dataset using the methodology proposed with the Firehose is prohibitive for many reasons. We propose a methodology that can give an indication of bias for a particular hashtag.

In this section, we investigate whether another open data source, Twitter’s Sample API, can be used to find bias in the Streaming API. We show that using the Sample API, one can accurately detect bias in the Streaming API without the need of the prohibitive Firehose. We focus on alternative methods to help the user understand *when* their data diverges from the true activity on Twitter. We continue to show that not only can one find the bias using this method, but that these results are consistent regardless of when and where the Streaming API query was issued.

#### Validation of the Streaming API

We define the true trend of a particular hashtag,  $h$  as a function,  $f(h)$ . We define the trend of  $h$  as it is conveyed through the Streaming API as  $t(h)$ . To make this estimation, we consider another freely-available open Twitter data source, the Sample API. Unlike the Streaming API, the Sample API takes no parameters and returns a 1% sample of all of the Tweets produced on Twitter. Here, we empirically assess the Sample API’s ability to return a truly random sample and continue to build a

Table 5: Data Collected to Test Bias Detection Approach Introduced in This Chapter.

Data Source	Keywords	No. Tweets
Firehose (Gnip)	syria	214,383
Sample API	N/A	734,172

Table 6: Significance Levels of  $\tau_\beta$  Statistic for Top  $K$  Hashtags, Sample API vs. Firehose. All Lists of Size Greater Than 40 Had  $p$ -Values  $< 10^{-6}$ .

Top- $k$	$\tau_\beta$	$p$ -value
10	0.988826	0.000069
20	0.778663	0.000001
30	0.655072	0.000001
40	0.549759	0.000002
50	0.604880	$< 10^{-6}$
...	...	...
450	0.476931	$< 10^{-6}$

framework to compare the Streaming API data with the Firehose using the Sample API as a proxy. We call the trend from the Sample API  $s(h)$ . Given the sparsity of the Sample API, it is likely impossible to find the real trend  $f(h)$  from  $s(h)$ , however  $s(h)$  can be used as an indicator to understand when  $t(h)$  is biased.

### Vetting the Randomness of the Sample API

Given the evidence of bias that was observed in the Streaming API, we must proceed with caution before using the Sample API as a surrogate gold standard. We begin our assessment of the randomness of the Sample API by collecting data from the feed. We collect all of the tweets available through the Sample API on 2013-08-30 from 17:00 - 21:00 UTC, and post-filter them by the keyword “syria”. Simultaneously, we collect all of the tweets matching the keyword “syria” from the Gnip Twitter feed.

The Gnip<sup>18</sup> feed is another outlet for Firehose data, it also provides 100% of the publicly-available Tweets on Twitter. We report the keywords and the collection volume for this dataset in Table 5.

To verify the validity of this source we compare the ranked list of top hashtags in both sets. We first plot Kendall’s  $\tau_\beta$  score of the Sample API against the Firehose. Kendall’s  $\tau_\beta$  calculates the number of concordant and discordant pairs between two ranked lists. This score gives us a sense of whether the frequency of hashtags coming through the Sample API is the same as the Firehose. We plot the average and standard deviation of 100 perfectly randomly sampled datasets of the same size as the Sample API against the Firehose. A plot of the rank correlation in the top hashtags from the Firehose and the Sample API is shown in Figure 12.

Overall we see that the shape of the trend of the Sample API closely resembles that of the random samples, a promising sign that the Sample API is unbiased. However, we still see that the  $\tau_\beta$  values occasionally fall outside of one standard deviation of the distribution of random samples. We perform a statistical test to ensure that the Sample data and Firehose data are not independent. To perform the statistical test we calculate two-sided significance level from the Kendall  $\tau$  statistic, which tests the following hypothesis:

$H_0$  – The top  $k$  hashtags in the Firehose data and the top  $k$  hashtags in the Sample API data are independent,  $\tau_\beta = 0$ .

The results of this experiment at varying levels of  $k$  are shown in Table 6. In all cases we are able to reject  $H_0$  with a 95% confidence level. The strong correlation between the top hashtags and the strong similarity between the two distributions demonstrates that the Sample API is representative of the true activity on the Firehose.

---

<sup>18</sup><http://gnip.com/>

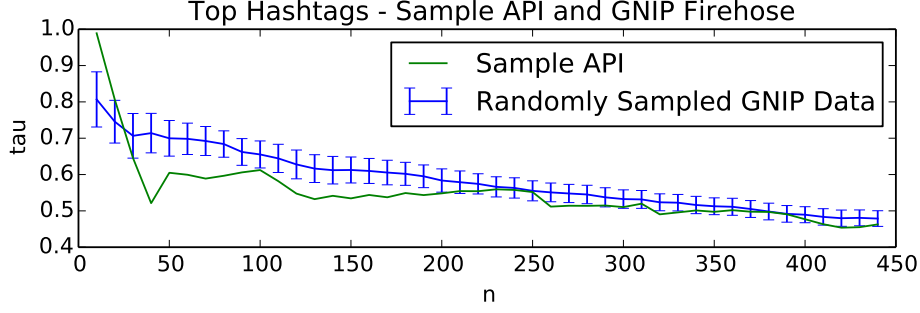
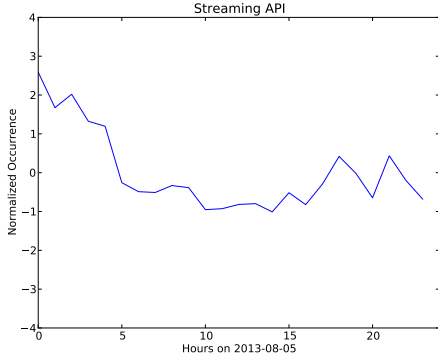
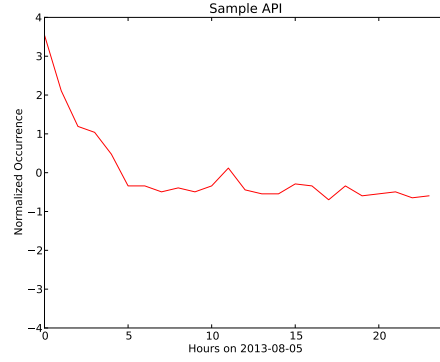


Figure 12: Rank Correlation of Sample API and Gnip Firehose. Relationship between  $n$  - Number of Top Hashtags, and  $\tau_\beta$  - the Correlation Coefficient for Different Levels of Coverage.

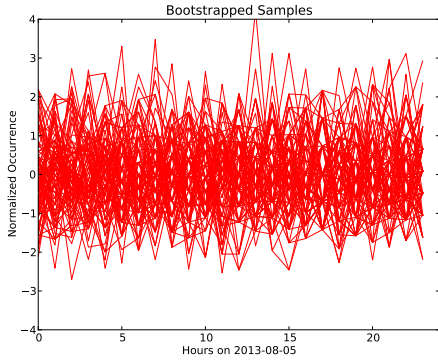
With the revelation that the Sample API is a random sample of the Firehose, one might be tempted to simply use this as a replacement for the Streaming API, post-filtering the Sample API's data to suit their needs. While this is correct from the perspective of data sampling, this approach will lead to a massive loss in data. For example, assume that a query matches 10% of the data on the Firehose. Filtering the Sample API will give us a dataset that is 0.1% of the size of the Firehose, but unbiased. The Streaming API's sometimes-biased sampling method will give us a 1% sample of the Firehose that is 10x larger than the Sample API. This may be counterintuitive as larger samples have a greater chance of being unbiased, however since the issue resides in the sampling methodology of the Streaming API a sample of any size from this source has the potential to be biased. The size difference between the two sources cannot be ignored. Instead, we will use the Sample API to identify periods of bias in the Streaming API.



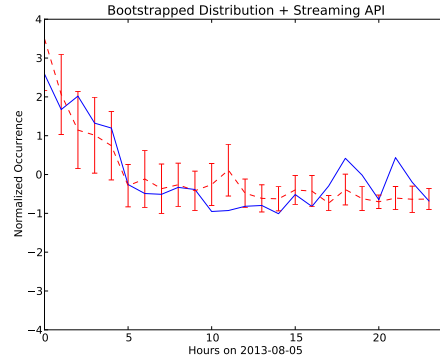
(a) Streaming API Results. Trendline for “#believemovie” over one day.



(b) Sample API Results. Trendline for “#believemovie” over one day.



(c) Trendlines from 100 bootstrapped samples of the Sample API.



(d) Bootstrapped average and  $\pm 3$  standard deviations overlaid with Streaming API.

Figure 13: Figures Outlining Different Steps of the Process for Finding Bias in the Streaming API.

### Finding Bias in the Trend of a Hashtag

Now that we know that the Sample API gives us an unbiased picture of Twitter’s Firehose, we can continue to construct a framework that incorporates this source to find bias. Herein, we propose a methodology that finds the bias in the Streaming API and reports to the user collecting the data when there is likely bias in the data.

With only one unbiased view from the Sample API, it is difficult to understand



what we should expect from our Streaming API data. When the results from both sources match there is clearly no problem. When there is a difference how do we know if the relative error between the Sample API and the Streaming API at one time step is significant or if it is just a small deviation from a random sample? To better understand the Sample API’s response, we bootstrap (B. Efron 1987) the Sample API to obtain a confidence interval for the relative activity for the hashtag at a given time step.

We begin by normalizing both the Sample API and Streaming API time series. This is done by calculating the mean, and standard deviation of each of the counts in the time series. Finally, we normalize each point by its standard score, which is calculated as:

$$Standard\_Score(t_i) = \frac{t_i - \mu_T}{\sigma_T}, \quad (4.1)$$

where  $\mu_T$  and  $\sigma_T$  are the mean and standard deviation of all of the time periods in the time series, respectively, and  $t_i$  is an individual time series point. This is done to ensure that the distribution of points from both time series is  $\mathcal{N}(0, 1)$ . We create 100 bootstrapped samples for each hashtag. We then extract the time series data from each sample and normalize them as we did before. This gives us a distribution of readings for each time period in the dataset. Next, we compare this distribution to the normalized time series from the Streaming API to detect the bias. We take the sample mean and sample standard deviation of this distribution at each point  $t_i$  as  $\mu_i^b$  and  $\sigma_i^b$ . Borrowing the threshold used in control charts (Ryan 2011), we say that any Streaming API value at time  $t_i$  that is outside of  $\pm 3\sigma_{t_i}$  is biased.

We show a full example of our method in Figure 13. We enumerate the process for a single hashtag, “#believemovie” on August 5th, 2013. We choose this hashtag as it is one of the most frequent hashtags on this day. The process begins with the

time series data for this hashtag from both the Streaming and Sample APIs, shown in Figures 13a and 13b, respectively. Looking at the two figures, we immediately see a difference in the trends of this hashtag from the two sources. To obtain a confidence interval on the difference of the two sources, we create 100 bootstrapped samples. The time series extracted from these samples are shown in Figure 13c. Finally, taking the mean and 3 standard deviations of the bootstrapped samples at each time point, we obtain the confidence intervals seen in Figure 13d. We make several observations from Figure 13d. First, a spike that occurs between hours 10 and 11 is underrepresented in the Streaming data. Also, the spikes that appear after hour 16 are both over-represented in the Streaming API. Due to the nature of our bootstrapping method, these observations are all statistically significant at the 99.7% confidence interval.

#### Signal Usability Under Sparsity

One potential drawback of our method lies in the sparsity of the Sample API. Accounting for only 1% of the Firehose, the “long tail” of hashtags will largely be ignored by the Sample API. This is problematic for researchers who wish to verify their Streaming API query’s results when their query is focused upon hashtags that do not see a lot of activity as a fraction of the entire activity on Twitter. One counterargument is that these kinds of queries will likely not eclipse the 1% threshold, and in general will be unbiased.

One observation we make is that the times where our bootstrapping method will be futile is in times where there is data for the hashtag from the Streaming API and no data from the Sample API. In such cases, a bootstrapping approach will give

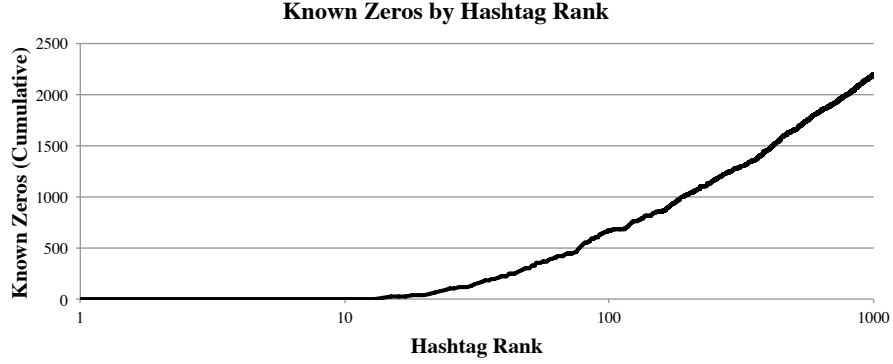


Figure 14: Cumulative Known Zeros Ranked by Hashtag Popularity. We See That the Most Popular Hashtags Have Relatively Few Points of Missing Data, While the Less Popular Hashtags Have Many More. To Help the Reader Separate the Higher-Ranked Hashtags’ Missing Values, We Plot the  $x$ -Axis on a Log Scale.

us a degenerate distribution with mean 0, not allowing for meaningful comparison between the sources. We test the sparsity of the Sample API by finding “known zeros”, hashtags seen in the results of the Streaming API query, but not in the Sample API for a particular time unit.

Figure 14 shows the number of known zeros in the top 1,000 hashtags, with each hashtag ordered by frequency. Here, we see that the first hashtags are nearly perfect, with a total of only 4 known zeros in the top 10 hashtags. However, as we continue down the list, we begin to see more and more known-zeros. While this method helps researchers to find bias in their Streaming API queries, there are still many hours for many hashtags where no claim can be made about the validity of the data.

#### 4.1.2 Geographic and Temporal Stability of Queries

In addition to tackling the bias problem, we also analyze the stability of the results when the data are collected in different geographic areas, and for queries started at

different times. Do identical Streaming API queries started at different times get different responses during the overlap of their execution? Do identical Streaming API queries get different responses if they are executed from different geographical regions? To ensure that the results obtained in this chapter hold for researchers outside of the US, we assess whether Twitter issues the same results to identical queries.

To answer these questions, we collected data from the Streaming API in both the United States (USA) and Austria (AT) with the following scheme: every 20 minutes, we start a query that lasts for 30 minutes. For example,  $query_1^{USA}$  and  $query_1^{AT}$  collect tweets from 00:00 - 00:30 UTC,  $query_2^{USA}$  and  $query_2^{AT}$  from 00:20 - 00:50 UTC, and  $query_3^{USA}$  and  $query_3^{AT}$  from 00:40 - 01:10 UTC, and so on. Each query is configured with exactly the same parameters. In structuring our queries this way, we can control both for time, and for location. By looking at the 10-minute overlaps in the adjacent within-country queries (i.e. all  $query_i^C$  and  $query_{i+1}^C$ ), we can gain an understanding of whether identical queries started at different times get the same results. By looking at entire queries across countries (i.e.  $query_i^{USA}$  and  $query_i^{AT}$ ), we can understand whether identical queries started at the the same time from different locations get the same results.

The data we collected spans from 2013-10-20 06:20 UTC - 2013-10-22 22:20 UTC. Each query starts exactly on the 20-minute interval and lasts for exactly 30 minutes. In this way, we collect 194 datasets in total from each country. In the between-country case, we compare the entire dataset as both queries are running in both locations. In the between-time case, we only compare the 10-minute overlaps between  $query_i^C$  and  $query_{i+1}^C$ .

Table 7: Number of Comparisons, Median, Average, and Standard Deviation of Twitter ID Jaccard Scores across All Comparisons. Because the Temporal Comparisons Are between Query, We Have One Less Than in the Geographic Comparison.

Comparison	N	Median	Mean	$\sigma$
Geographic Comparison	194	0.976	0.941	0.092
USA Time Comparison	193	0.996	0.995	0.003
Austria Time Comparison	193	0.996	0.942	0.186

### Between-Country Results

To compare the datasets, we calculate the Jaccard score of the Tweet IDs from  $query_i^{USA}$  and  $query_i^{AT}$ . We then take the median, average, and standard deviation of these Jaccard scores. These results are reported in the first row of Table 7. Here, we see a very high average and a very low standard deviation between the Jaccard scores, indicating that the results from these two queries are nearly identical. These results indicate that no preference was given to the queries originating in the United States. We are hopeful that researchers outside of the US will obtain similar results.

### Between-Time Results

To compare the datasets, we fix the country  $C$  and calculate the Jaccard score of the Tweet IDs from  $query_i^C$  and  $query_{i+1}^C$ . The results for the USA and Austria queries are shown in rows 2 and 3 of Table 7, respectively. In the case of the USA, we see even stronger results, with an extremely high average and an extremely low standard deviation. Here, we can see that the overlapping times receive practically the same dataset in all cases. In the case of the Austrian datasets, we see that there is a wider distribution of Jaccard scores between query windows, however we continue

to see an extremely high mean, which gives us confidence in the coverage in these results.

#### 4.1.3 Discussion

In this chapter we ask how to find bias in the Streaming API without the need for costly Firehose data. We test the representativity of another freely-available Twitter data source, the Sample API. We find that overall the tweets that come through the Sample API are a representative sample of the true activity on Twitter. We propose a solution to harness this data source to find time periods where the Streaming API is likely biased. Finally, we show that results obtained through the Streaming API for a given time period see a significant amount of overlap between queries generated from both the United States and from Austria, and between queries started at different points in time.

One question that arises is how to integrate the results of our framework into an individual’s research. One solution is to focus research efforts on time periods where no (or little) bias is found in the dataset. Another potential solution is to purchase Twitter data from a reseller such as Topsy<sup>19</sup> or Gnip, but only paying for the biased time periods. A third solution is to incorporate other forms of social media such as blogs. This allows for a multifaceted view of the data, and can give the researcher more depth in times where their Twitter data may be of question. One way to incorporate these other views is to cross-reference users from Twitter with other social media outlets, one such solution has been proposed in Zafarani and Liu 2013.

We have studied the feasibility of our method with the sparse signal that we get

---

<sup>19</sup><http://www.topsy.com>

from the Sample API. Overall, we find that this method can be used when the query issued by the researcher receives a lot of attention from Twitter users. We also find that this method is less useful when the query receives less data. One potential way to alleviate this problem is to use other bootstrapping methods such that proposed in Kirk and Stumpf 2009, which takes into account neighboring values to compute the values. Future work will attempt to find bias in sparse data scenarios, and adapting these methods to the speed and ephemerality of Twitter data.

## 4.2 Mitigating Social Data Collection Bias

In the previous chapter we discussed how we could identify bias in a previously collected dataset. We showed how a statistical methodology could enable a researcher to identify and remove bias in a dataset. However, this has the unsavory side effect of asking a researcher to ignore certain parts of their dataset. This is not ideal for many researchers who need to use as much data as possible in order to collect statistically valid samples.

In this chapter we present a new approach for collecting social media data that enables us to collect a representative sample from a social media site. This is based upon reparameterizing the queries that we send to the APIs in order to maximize the amount of data that we are able to obtain, and minimize the amount of sampling bias. This is based upon our finding earlier that the amount of bias is a function of how the sampling rate that Twitter imposes upon us.

Herein, we try to optimize the amount of data we get from the Streaming API. The Twitter Streaming API uses a mechanism called “limit track” to inform the requester of the number of tweets that were not delivered to them due to exceeding the 1% rate

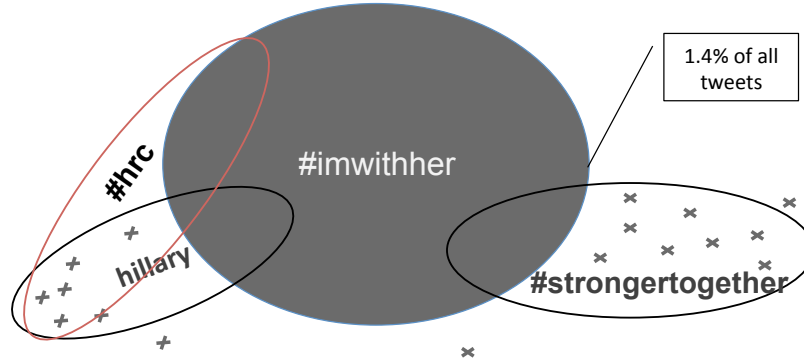


Figure 15: The General Idea behind the Splitting Mechanisms Proposed in This Section. The Grey Areas Denote the Relevant Tweets That We Want to Collect. If We Just Have One Crawler Tracking “#imwithher,” Then We Get the Entirety of the Grey Area in the Circle but Lose the Grey “x”s. By Creating Additional Crawlers, We Can Get a Greater Number of Those “x”s, in Turn Getting a Larger Sample of the Data.

limit. The limit track data is provided periodically along with tweets delivered in a particular stream. Unfortunately, if the limit track value provided by Twitter is not a reliable measurement, then it becomes significantly more difficult to determine the overall sample population, and, as a result, the level of bias in the sample remains unknown.

#### 4.2.1 Designing Novel Crawling Approaches

The main idea behind all of the methods in this chapter is that they are based upon splitting the parameters into multiple crawls. By doing this we should be able to get more tweets than with one crawler alone. This will be confirmed through experimentation in a subsequent section. There are many ways to split the parameters in order to achieve this task. We will propose two and investigate their usefulness.

The idea behind these approaches is that each individual crawler may be able to



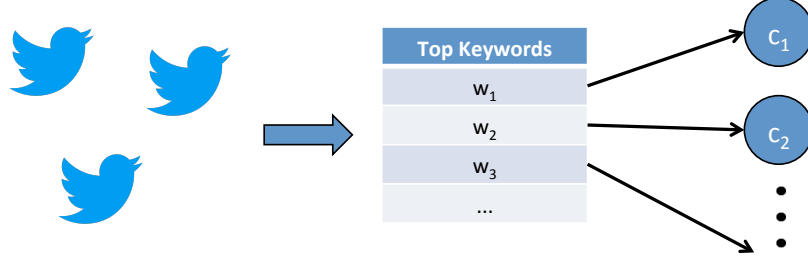


Figure 16: Overview of The Round Robin Splitting Approach.

give us more data than any one crawl. This is depicted in Figure 15, where we see that we are able to capture more the grey (relevant) area with additional crawlers.

#### 4.2.2 Round Robin Splitting

The round robin method of stream keyword splitting is an effective baseline for other splitting methods as it is a straightforward method that requires very little additional processing power. Sampling the amount of tweets gathered and missed at each split level requires running one baseline stream that contains all selected keywords as well as  $k$  additional streams that contain the keywords split between each stream. While it is possible to sample all split levels simultaneously, the number of required accounts for a test of this type is  $x_k = \frac{k(k+1)}{2}$  where  $k$  is the number of split levels and  $x$  is the number of accounts. Sampling all splits up to a split level of 7 would require 28 separate account which is unfeasible for our purposes. Additionally, the processing power required to maintain the set of unique tweet IDs for each stream becomes problematic very quickly. Alternatively, using a single baseline stream that contains all keywords and comparing the results to each split level independently requires a much lower number of accounts,  $x_k = k + 1$ .

Both of the methods introduced in this chapter consist of a “priming stage.” The

priming stage consists of a 15 minute run with a single crawler. This is done to get a first set of tweets in order to make the splitting. All splits after that are performed on the tweets collected in the previous iteration. This is shown on the left of Figure 16. At the completion of the priming stage, the word pairs, from the most frequently occurring to the least frequently occurring, are assigned to streams in a round robin fashion. Each split level runs for 30 minutes before resetting all streams, including the baseline stream, and beginning the next split level. Resetting the baseline stream is key to analyzing each stream level in this method as it allows a comparison of the difference between a single stream and multiple split streams over a given window of time and thereby making it unnecessary to maintain all data for every level of split at once.

The graph shown in Figure 17 show that we were able to eclipse the limit track maximum by 12 splits at which point we were able to gather six times as many unique tweets containing proper keywords than was possible with a single stream. Reaching 20 split levels nearly doubled the number of unique tweets gathered over the maximum indicated by Twitter. Constructing round robin splits can be found in Algorithm 1.

---

**Algorithm 1:** Round Robin Splits Construction

---

```

input : graph  $G$ , num_clusters  $k$ 
output : array of lists
for node  $v$  in  $G(V, E)$  do
    | keywordList +=  $v$ 
end
sortedList := Sort(keywordList by occurrence rate);
for word, index in sortedList do
    | listNum := index %  $k$ ;
    | assign word to split list  $k$ 
end

```

---

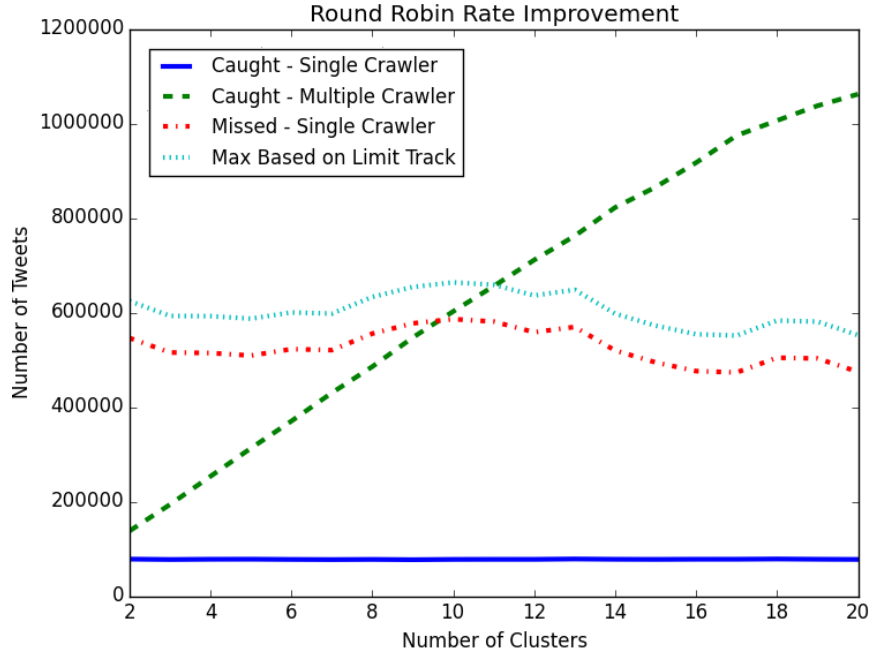


Figure 17: Round Robin Splitting Based on Word Co-Occurrence Tends to Show a Steady Rate of Gain as Additional Crawlers Are Added.

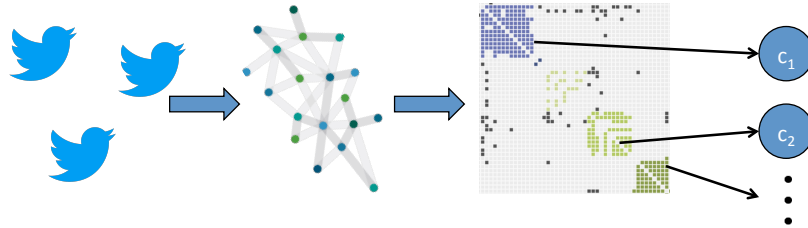


Figure 18: Overview of the Spectral Clustering Splitting Approach.

#### 4.2.3 Spectral Splitting

Spectral clustering, an extension of  $K$ -Means clustering which performs “low-dimension embedding” (Ng, Jordan, and Weiss 2001), directly leverages the word occurrence graph. This clustering method allows us to define a number of clusters,  $K$ , and the spectral clustering algorithm will incorporate the nuances of the similarity

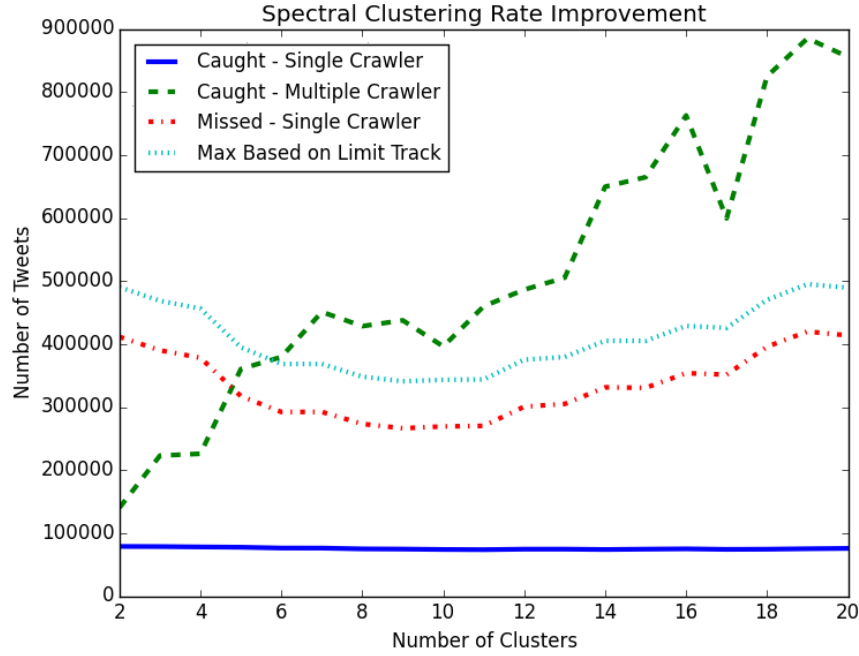


Figure 19: The Results of Spectral Clustering Show an Increase in the Overall Sample Coverage. However, the Clustering Creates Unbalanced Splits Where One Stream, While Still a Good Cluster, May Contain Significantly More Words Than Others. The Lack of Balance Manifests through Instability in the Rate of Gain from Each Additional Crawler.

between the items in order to improve cluster results. Like most clustering algorithms, spectral clustering does not make any guarantee on the size of each cluster.

This approach is based upon performing a spectral clustering of the keyword co-occurrence graph, as shown in Figure 18. The main idea behind this method is that we want to reduce the redundancy of the data collected by each crawler. As intuition, imagine if after we performed a split, each cluster retrieved exactly the same data. That would not be beneficial as we are not gaining any new data and not learning anything new about the core set of hashtags we wish to study. Thus, to improve the performance of our approach we want each cluster to get as much unique data as

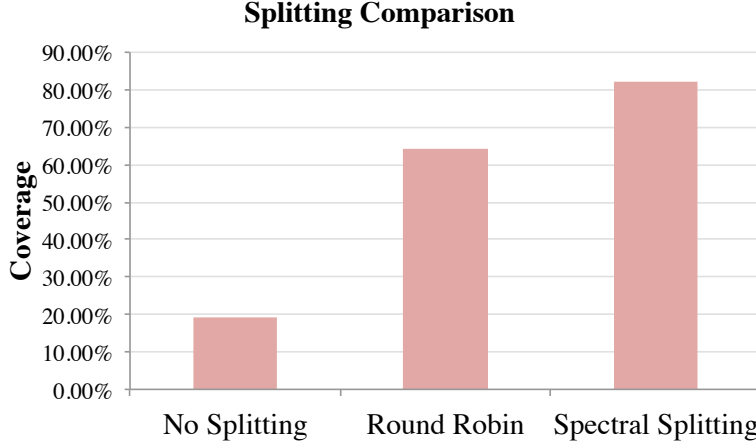


Figure 20: Results of Different Methodologies at the 3-Split Level.

possible. By grouping keywords that tend to occur together in clusters, we are likely to minimize the redundancy of the data that each cluster provides.

The approach is simple. First, we use the “priming step,” explained above, to get a set of tweets. Next, we build a word-word co-occurrence graph based upon how the co-occur in tweets. This is a weighted graph where nodes are words, and edges are the number of times that the pair of words co-occur in the corpus of tweets. Next, we run spectral clustering on this graph to yield a series of clusters. These clusters become the parameters for the crawl in the next iteration of the crawler.

The results of this approach can be seen in Figure 19. Clustering based on word occurrence quickly passes the Twitter limit with only 6 streams active but shortly thereafter struggles to gain much ground. Wild fluctuation can be observed between each split level and while there is overall growth it is possible to gather a smaller sample with a larger split level. Such inconsistencies were not observed in Figure 17 further indicating how detrimental sensitivity to cluster size is when considering methods for gathering tweet samples. Spectral clustering based splits can be found in Algorithm 2.

---

**Algorithm 2:** Process for Constructing Spectral Clustering Splits.

---

```
input : graph G, num_clusters k
output : array of lists
matrix A := [];
for node v in G(V, E, W) do
| wordIDs[v] := unique ID
end
/* create affinity matrix for spectral clustering */
for word pair (v1,v2) in G(V, E, W) do
| Construct symmetric matrix A for all words such that v1,v2 := E.weight;
end
labels := the result of Spectral Clustering with k clusters and the matrix A;
for label, cluster in labels do
| splitLists[cluster].append(label)
end
```

---

Table 8: Sample Coverage by Approach. “1-Split” Is the Amount of Data That We Would Get with No Splitting Approach.

	1-split	2-split	3-split	4-split
Round Robin	19.02%	50.54%	82.58%	64.34%
Spectral Clustering	19.02%	28.95%	78.63%	82.08%

#### 4.2.4 Performance of Splitting Approaches

We have proposed two methods for splitting keywords in social media data queries and shown preliminary results for how they perform on Twitter. We will now compare these methodologies in order to show how they perform side-by-side.

A comparison of each splitting method can be seen in Figure 20. These results are the average coverage rates related to the total sample size estimated from the limit track over 20 trials. Though the Twitter limit track is not a perfect indicator of the total population of data available from a complete set, it is used in this comparison as a relative measure as opposed to an absolute. Given additional crawlers, it is very

likely that the limit track would again be eclipsed. Each splitting method was able to produce a sample significantly closer to the total population as estimated by Twitter and always many times larger than a single stream.

### BIAS FROM MALICIOUS ACTORS

Bias does not originate purely from data collection. There are many actors in the real world that attempt to skew the contents of social media. This is done to make certain ideas, products, or candidates appear to be more popular than they truly may be. The ways in which they inject this bias into social media data varies. In this chapter we will introduce three such methods and ways to detect them. First, we will introduce bots. Bots are automated accounts that work together at a large scale to skew the content of a site in their favor. By automatically posting a large number of social media messages, they can easily make their message seem more prominent than it actually is. Bots generally work together, with thousands of millions of accounts working in tandem to promote a message.<sup>20</sup> However, the decisions that a social media site makes when distributing data can enable them to further promote their messages. In this chapter we will discuss how bots can take advantage of Twitter's data distribution mechanisms to promote their messaging. Finally, not all users that attempt to skew the perceptions of social media users are bots. Many are sophisticated users who are trained to promote pointed content online. We will investigate one specific case of these users, "shills," and discuss possible ways to detect them.

---

Portions of this chapter have been published at WWW 2016 (Morstatter, Dani, et al. 2016) and ASONAM 2016 (Morstatter, Wu, et al. 2016).

<sup>20</sup><https://krebsonsecurity.com/2011/12/twitter-bots-drown-out-anti-kremlin-tweets/>



## 5.1 Identifying Bots: The Importance of Recall

Social media is an important part of every day life. With billions of users producing and consuming information every day, it is a natural extension that people turn to this medium to read and disseminate news. With so many people turning to social media, malicious users like bots have begun using these sites to sway the discussion (Ratkiewicz, Conover, Meiss, Goncalves, Patil, et al. 2011; Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011; Thomas, Grier, and Paxson 2012). Bots, social media accounts that are controlled by software, have risen to prominence on social media. Bots cause users to lose trust that social media platforms can deliver news honestly, as they become suspicious that the stories they see at the top of their feeds were “pushed” there by manipulative bots. Furthermore, bots impinge the work researchers perform on social media as they can cause researchers to draw false conclusions about the populations under study. Researchers wish to understand human behavior through the lens of social media (Mejova, Weber, and Macy 2015), and this is often impinged by the wealth of content pollution created by automated social media accounts.

Bot detection is an important task in social media. Twitter, a popular social media platform, is plagued by automated accounts. One study has estimated that over half of the accounts on Twitter are not human<sup>21</sup>. More conservative studies estimate that the number of bots on Twitter lie somewhere between 5-9% of the overall population<sup>22</sup>. These bots, generate 24% of the tweets produced on Twitter<sup>23</sup>. While some bots

---

<sup>21</sup><http://blogs.wsj.com/digits/2014/03/21/new-report-spotlights-twitters-retention-problem/>

<sup>22</sup><http://www.nbcnews.com/technology/1-10-twitter-accounts-fake-say-researchers-2D11655362>

<sup>23</sup><https://sysomos.com/inside-twitter/most-active-twitter-user-data>

innocuously tweet the time of day or a record of historical events, the ones we detect in this study seek to actively sway the discussion on social media by promoting certain trends and retweeting particular users. These bots are considered malicious as they try to perturb the discussion on social media, swaying the discussion to the topics of their choosing by falsely amplifying the size of a topic or keyword.

Bots are not just a danger to the users of social media, but also to those that study it. By coordinating massive tweeting campaigns of particular hashtags, bots have been able to influence measures on Twitter, including the trending topics (Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011). This manipulation of Twitter also bleeds across into the analysis that is done on this particular platform. Bots can also influence statistics performed on Twitter data, such as the top hashtags and the most important users in the data. This means that bots not only have the potential to damage the active conversation on social media, but they can also cause additional destruction by harming the quality of the work done by researchers studying social media. Clearly, there is a need for us to remove bots from social media both for the benefit of users and researchers.

Current approaches to bot detection focus on precision (Lee and Kim 2014; Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011; Xie et al. 2008). Optimizing precision can be useful as it helps to avoid the possibility of a real user being deleted from the site, which may anger other real users or cause them to leave the site. However, this is one extreme. An algorithm that only considers precision will have many false-negatives, as it will only predict the most acute examples as bots, leaving many bots undetected. This is not ideal from a research perspective, where we wish to study human behavior on social media. Thus, we want to include recall as the objective of our bot detection model. However, this is another extreme. The major

concern with optimizing recall is that it is trivial to achieve a perfect score: simply marking every user as a bot will yield a recall of 100%. Thus, we propose to study the balance between these measures. While our model focuses on recall, we measure its performance using the  $F_1$  score.

We propose a model that can detect bots on social media with a focus on recall. We evaluate by taking into account both precision and recall to strike the balance between these important evaluation measures. In this way, we show that our method can help to remove the most bots, while simultaneously keeping the precision relatively high. By applying BoostOR, researchers can remove more bots from their social media dataset and focus on the messages produced by real users.

#### 5.1.1 Bot Definition

A bot is any automated account in an online social network. This definition differs slightly from the one found in Boshmaf et al. 2011 and Boshmaf et al. 2013 as they differentiate social bots from self-declared and spambots although socialbots and spambots may have the same structure and purpose.

#### 5.1.2 Problem Statement

Given a set of all of the users in a dataset,  $U$ , and a set of labeled bots  $b \subset U$ , identify a set of users  $u \subset U$  that maximizes the function:

$$F_1(u, b) = 2 \frac{PR}{P + R} \tag{5.1}$$

where  $P = \frac{|u \cap b|}{|u|}$  is the precision of the model, and  $R = \frac{|u \cap b|}{|b|}$  is the recall.

The main contributions of this section are as follows:

- We collect two bot datasets using state-of-the-art collection techniques, and test two labeling techniques. We publish this data to allow for reproducibility and encourage comparison with future work.
- We build a model that detects bots on Twitter with an emphasis on the  $F_1$  measure, which considers recall as well as precision. We compare this model to existing approaches for bot detection and find that the model achieves superior performance in this task, yielding high recall with only a minor loss in precision, which contributes to it achieving the best  $F_1$  score in all algorithms we studied.

### 5.1.3 Labeled Datasets for Bot Detection

Obtaining a social bot dataset can be difficult due to the challenge in obtaining definitive ground truth. While many approaches have been employed to obtain ground truth, we use two main approaches to label social media users as bots or humans: 1) observing whether the social media platform suspends these users and treating these suspended users as bots, and 2) creating “honeypot” bot accounts that to lure bots into following them, and then labeling these followers as bots.

We create two datasets to test the bot detection approaches. The labels for each dataset are obtained using different labeling approaches. Here, we enumerate the two datasets used in this study, along with describing the bot labeling approach<sup>24</sup>.

---

<sup>24</sup>Both datasets can be obtained from the following link: [http://www.public.asu.edu/~fmorstat/ASONAM\\_data.zip](http://www.public.asu.edu/~fmorstat/ASONAM_data.zip).

Table 9: Statistics of the Data Used for Bot Detection.

Property	Libya	Arabic Honeypot
Tweets	1,150,192	504,679
Retweets	576,167	220,500
Unique Users	94,535	6,285
Labeling Approach	Suspended Accts	Honeypots
Bot Ratio	7.5%	49.0%

### Arab Spring in Libya

From February 3<sup>rd</sup>, 2011 to February 21<sup>st</sup>, 2013, we collected Twitter data pertaining to Arab Spring activity in Libya. The data was collected from Twitter’s Streaming API<sup>25</sup>, a service which provides a stream of tweets matching a query (Kumar, Morstatter, and Liu 2014). The query we used to collect this data consisted of the following keywords: #libya, #gaddafi, #benghazi, #brega, #misrata, #nalut, #nafusa, #rhaibat, as well as a geographic bounding box around Libya<sup>26</sup>. Statistics on the dataset are shown in Table 19.

We obtained labels of whether a user is a bot or human by observing how Twitter handled these users. In February of 2015, we crawled each user in the dataset and observed the status of his Twitter account. We observed the user’s account status via the `statuses/user_timeline` API endpoint<sup>27</sup>. The status can take on one of three values:

1. **Active:** A user whose account is still open and available on the site.

<sup>25</sup><https://dev.twitter.com/streaming/reference/post/statuses/filter>

<sup>26</sup>Southwest Lng/Lat: 23.4/10.0; Northeast Lng/Lat: 33.0,25.0.

<sup>27</sup>[https://dev.twitter.com/rest/reference/get/statuses/user\\_timeline](https://dev.twitter.com/rest/reference/get/statuses/user_timeline)

2. **Deleted:** A user whose account has been deleted. A user can be deleted by violating Twitter’s policies. This is considered a permanent ban.
3. **Suspended:** A user whose account has been suspended for violating Twitter’s policies. This is considered a temporary ban, where the user can petition Twitter to have his account reinstated.

These labels were obtained by using Twitter’s APIs to crawl the dataset and inspecting the response code from the API. Through this process we discovered that **92.5%** of the users are active, **4.7%** are deleted, and **2.8%** are suspended. Because deleted and suspended have similar meanings, we consider both labels as a bot.

At first glance, we notice that the fraction of users identified as bots by this labeling technique is around 7.5%. This distribution gives a very conservative estimate of the number of bots on Twitter. This is a side effect of an industry approach which focuses purely on precision in order to avoid accidentally deleting some real users. We realize that this is not representative of the true distribution of bots on the site (Wei et al. 2015), and that many bots may have been overlooked by Twitter. With this in mind, we introduce our next dataset which focuses on detecting bots in the wild through honeypots which tweet specific content.

### Arabic Honeypot Dataset

This dataset consists of bots tweeting messages in Arabic. To collect this dataset, we construct a honeypot network. This network consists of 9 accounts controlled automatically by a single controller. Each account tweets messages containing Arabic phrases identified by a subject matter expert pertaining to a specific group of people. Each account also randomly follows other honeypots in our network. Since bots have

a lower chance of forming social ties (Thomas et al. 2011), we perform this random following process to lower the chance that our accounts are deleted by Twitter’s automatic account removal algorithm. Additionally, each honeypot can randomly retweet one of the other honeypots it follows in order to give that honeypot prominence on the network and lower its probability of being deleted due to Twitter’s policies.

All of the bots in our honeypot dataset behave identically, as dictated by the controller. Each bot collects tweets using the selected Arabic phrases from Twitter’s Streaming API. At random time intervals, the bot either copies the tweet, passing it off as its own, or retweets it from the user who originally posted it. The full logic for the honeypot controller is shown in Algorithm 3. Using a network of 9 honeypots, we consider all followers of our honeypots to be bots: due to the way our honeypots behave, no normal user would follow them (Lee, Eoff, and Caverlee 2011). With this approach we collected 3,602 active bot accounts who followed one of our honeypot accounts.

While the honeypot method yields a set of bot accounts, it does not give us a set of real users. This is because we only look at the bot followers to our honeypots, and we do not have ground truth labels for real users. In order to test our model, we need to collect a set of real users to use as negative training instances. To do this, we manually inspected a seed set of 10 users who also tweeted the same phrases. We did not extract verified accounts as these users are not normal users. They are often controlled by public relations firms and tweet on specific topics. We did this to ensure that the algorithm we train finds bot patterns in the text, and does not simply learn the difference in language distributions. Moving forward with the assumption that real users do not follow malicious bots, we use 1-link snowball sampling to collect their immediate network. We ensured that each user in the sample had fewer than

---

**Algorithm 3:** Logic for Honeypot Controller. The Controller Manages the Honeypot Accounts Used to Collect Bots in the Wild. All Randomly-Generated Numbers Mentioned in the Pseudocode Are Generated Uniformly at Random.

---

```

while True do
    Randomly choose one honeypot, h;
     $r \leftarrow$  random number  $\in [1, 10]$ ;
    if  $r = 1$  then
        | h retweets the most recent tweet from another randomly-selected
        | honeypot;
    end
    else
        Sample tweets from Twitter Streaming API for 20 seconds, filtering
        based on Arabic phrases, call sample set S;
         $r \leftarrow$  random number  $\in [1, 10]$ ;
         $s \leftarrow$  randomly-selected tweet from S;
        if  $r < 3$  then
            | h retweets s;
        end
        else
            | h copies s and tweets it word-for-word;
        end
    end
    Wait 10 seconds;
end

```

---

1,000 followers to make sure that we did not collect any celebrities (Zafarani, Abbasi, and Liu 2014). We also inspected each user in the sample to ensure that he tweeted one of the phrases at some point in his last 200 tweets. This approach yielded 3,107 real-world accounts, which helped us to maintain the same class distribution reported in other approaches (Lee, Eoff, and Caverlee 2011).

#### 5.1.4 Bot Detection Approaches

We propose a model for bot detection which considers recall in its formulation. As baselines we consider popular heuristics used to detect bot accounts. We also consider



a model based upon topic modeling. In this section we present the formulation for all of the bot-detection approaches we study. We begin by introducing the heuristics and baselines considered, and continue to introduce our model, *BoostOR*.

## Heuristics

Here we introduce a set of heuristics that are used to differentiate bot from non-bot users in social media. These heuristics are based upon state-of-the-art studies in bot detection on social media.

### Fraction of Retweets

This measures the number of times the user published a retweet divided by the number of tweets the user has published, calculated as:

$$Heuristic_{Retweet}(u) = \frac{|\{x|x \in tweets^u, x \text{ is retweet}\}|}{|tweets^u|}. \quad (5.2)$$

Whether a tweet is a retweet is determined by looking at the “retweeted\_status” field of the tweet’s data when returned through the API. If the field contains tweet information, we consider it to be a retweet. This measure was introduced in Ratkiewicz, Conover, Meiss, Goncalves, Flammini, et al. 2011, and hypothesizes that bots are unable to produce original content, so they rely on the retweet feature in Twitter in order to establish their presence.

### Average Tweet Length

A tweet’s text is limited to 140 characters, but it is possible that bots post fewer than that as they could just be promoting a URL or a hashtag (Lee and Kim 2014). To account for this, we introduce this heuristic which measures the average length of the user’s tweets. It is the sum of the characters in all of the tweets the user has published

divided by the number of tweets the user has published, formally:

$$Heuristic_{Length}(u) = \frac{\sum_i^{|tweets^u|} |tweets_i^u|}{|tweets^u|}, \quad (5.3)$$

where  $|tweets_i^u|$  is the length of user  $u$ 's  $i$ -th tweet, measured by the number of characters in the tweet.

### **Fraction of URLs**

This measures the number of times the user published a tweet containing a URL divided by the number of tweets the user has published, formally:

$$Heuristic_{URL}(u) = \frac{|\{x | x \in tweets^u, x \text{ contains URL}\}|}{|tweets^u|}. \quad (5.4)$$

Whether a tweet contains a URL is determined by looking at the “entities” field of the tweet returned by Twitter’s API. This measure has been studied previously (Ratkiewicz, Conover, Meiss, Goncalves, Patil, et al. 2011; Xie et al. 2008), and hypothesizes that bots are motivated to persuade real users into visiting external sites operated by their controller. In this way, this heuristic traps bots that are trying to promote URLs in their tweets.

### **Average Time Between Tweets**

It has been discovered that many bots tweet in a “bursty” nature (Lee and Kim 2014; Xie et al. 2008), publishing many of their tweets within a short amount of time. This behavior is measured as:

$$Heuristic_{Time}(u) = \frac{1}{|tweets^u| - 1} \sum_{i=2}^N (t_i - t_{i-1}), \quad (5.5)$$

where  $t_i$  is the time stamp of the  $i$ -th tweet in  $u$ 's timeline, sorted chronologically in ascending order (i.e.  $t_i \geq t_{i-1}$ ).

Bot accounts are created by malicious users to spread misinformation through their tweets. Thus, their content can be a strong indicator to expose such potential compromised accounts. The problem with using content for bot detection is that the raw text features are of high dimensionality and sparse, causing the “Curse of Dimensionality”. Inspired by the recent advances of topic modeling, we adopt latent Dirichlet allocation (LDA) to obtain a topic representation of each user. Since the raw features (words) can be viewed as being generated by a mixture of topics, and the bots are naturally more interested in certain topics, denoting each user as a distribution over different topics may help the learning process better identify them from regular accounts.

LDA is an unsupervised and probabilistic model that has been proven useful for extracting latent semantics of documents. The principle idea behind LDA is that it treats each document as a distribution over topics, and each topic as a distribution over the vocabulary in the dataset. LDA requires one parameter<sup>28</sup>,  $K$ , the number of topics in the corpus. From here, LDA learns two matrices:

1.  $\Phi$ : the *Topic*  $\times$  *Word* matrix. Each topic in LDA,  $\Phi_i$ , is a probability distribution over the entire vocabulary in the corpus. Thus,  $\Phi_i^j$  is the probability of word  $j$  occurring in topic  $i$ .
2.  $\Theta$ : the *Document*  $\times$  *Topic* matrix. Since each document is modeled as a distribution over topics, this each row,  $\Theta_i$ , contains the document’s distribution over all of the topics learned by LDA.

---

<sup>28</sup>We use the default hyperparameter value of  $\alpha = \frac{1}{K}$ , and  $\beta = \frac{1}{K}$ .

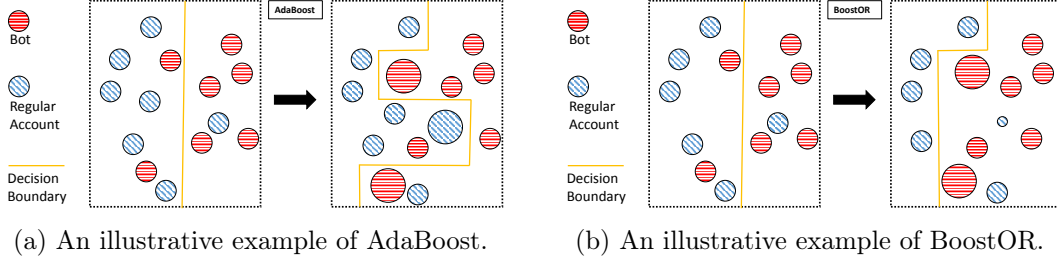


Figure 21: Illustration of the Models.

In this approach, we treat the user as a document. Each user’s document consists of the concatenation of all of the content of his tweets. We feed these documents into LDA with  $K = 200$ , obtaining  $\Phi$  and  $\Theta$  values for the corpus<sup>29</sup>. Since the  $\Theta$  matrix contains the affinity for each user to each topic, we can treat these as features to be fed into a classifier. We build an SVM classifier by directly using  $\Theta$  as the *Instance  $\times$  Feature* matrix, where the instances are users and the features are their affinities for the latent dimensions discovered by LDA.

### Supervised Model for Bot Identification

In this section, we will introduce how we optimize  $F_1$ , which includes both precision and recall, when exposing bots. Since bots are usually generated by different parties for different purposes, the discriminant characteristics of bots from different groups are unrelated. Such heterogeneity of bots makes it challenging to come up with a classifier. To this end, we formulate the problem as a boosting task. Boosting methods aim to achieve an optimal classifier through ensembling weak classifiers, which are

<sup>29</sup>We will discuss our selection of  $K$  in the experiments section.

more proper for this problem since different weak classifiers will focus on different bots.

First, we investigate whether boosting algorithms are directly applicable for bot detection. For generality, we selected AdaBoost for optimization. AdaBoost trains one weak classifier based on all training examples every iteration. In each iteration, the misclassified examples in the previous round are higher weighted in the next round. The final classifier is the weighted ensemble of all weak classifiers. Therefore, the key issues are how the weight are determined for different weak classifiers and training examples. Here we use  $\alpha$  to denote classifier weight, the weight of classifier  $t$  can be calculated as follows:

$$\alpha_t = -\frac{1}{2} \ln(\beta_t), \quad (5.6)$$

where  $\beta_t$  denotes the extent of the base learner deviates from the optimal solution.  $\beta_t$  can be directly calculated with training error  $\epsilon$  as follows:

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}. \quad (5.7)$$

As shown in Eq. 5.8,  $\alpha_t$  is greater than zero if  $\epsilon_t$  is less than 50%; while  $\alpha_t$  is less than zero if  $\epsilon_t$  is greater than 50%.  $\epsilon_t$  can be calculated through Eq. 5.8.

$$\epsilon_t = \frac{1}{\sum_{i=1}^m \mathbf{D}_t(i)} \sum_{i=1}^m \mathbf{D}_t(i) \mathbf{1}(h_i(\mathbf{v}_i) \neq y_i), \quad (5.8)$$

where the function  $\mathbf{1}(\cdot)$  equals one when the condition holds, and zero otherwise.

The second issue is to determine the weight of each training instance at each iteration. The instance weight is regulated depending on whether it is correctly classified in the previous round and the performance of the weak learner. As denoted in Eq. 5.9, when the weak learner's error rate is less than 50%, weights of misclassified users will increase while those of the rest decrease.

$$\begin{aligned}\mathbf{D}_{t+1}(i) &= \frac{\mathbf{D}_t(i)\exp(-\alpha_t y_i h_t(\mathbf{v}_i))}{Z_t} \\ Z_t &= \sum_{i=1}^m \mathbf{D}_t(i)\exp(-\alpha_t y_i h_t(\mathbf{v}_i)).\end{aligned}\tag{5.9}$$

The algorithm of AdaBoost for bot detection is illustrated in Algorithm 4. Note that  $m$  is the number of all users and  $k$  is the number of features. Since LDA is adopted to represent the user, here each user vector  $\mathbf{v}_i$  can be viewed as a probability distribution over  $k$  topics. The user labels are denoted by  $\mathbf{Y} = \{y_1, \dots, y_m\} \in \{-1, 1\}^{m \times 1}$ , where  $y_i = 1$  means that user  $i$  is a bot.

---

**Algorithm 4: AdaBoost for Bot Detection**

---

**Input:** The user-feature matrix  $\mathbf{V} \in \mathbb{R}^{m \times k}$ ,  
the user-label matrix  $\mathbf{Y} \in \{-1, 1\}^{m \times 1}$  and a weak learner  $h : \mathbf{V} \rightarrow \mathbf{Y}$   
the initial user weight vector  $\mathbf{D}_1 = (\mathbf{D}_1(1), \dots, \mathbf{D}_1(m))$   
the maximum number of iterations  $max_{iter}$ .

**Output:** The ensemble classifier  $H(\mathbf{v}) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(\mathbf{v}))$

**For**  $t = 1, \dots, max_{iter}$

Train weak learner using instance weight  $\mathbf{D}_t$  ;

Get the trained classifier  $h_t : \mathbf{V} \rightarrow \mathbf{Y}$  and training error  $\epsilon_t$  as Eq. 5.8;

Calculate the weight  $\alpha_t \in \mathbb{R}$  for  $h_t$  as Eq. 5.6;

Update user weight as Eq. 5.9;

**until** Convergence.

---

Since we aim to achieve a classifier which is sensitive to bots, we investigate how AdaBoost could be adapted for optimizing recall. As mentioned before, the weight of bots may be reduced if they are correctly classified. The reduction of weight leads subsequent weak classifiers to less focus on these bots, which is unfavourable. Therefore, we next investigate how the sensitivity to bots can be kept through regulating the weights of training instances. An intuitive solution is to avoid reducing weights of bots, which can be formulated as follows:

$$\mathbf{D}_{t+1}(i) = \mathbf{D}_t(i)\beta^{-y_i|h_i(\mathbf{v}_i - y_i)|}.\tag{5.10}$$

As shown in Eq. 5.10, if a regular user is predicted to be a bot, the corresponding weight will be multiplied by  $\beta^{|h_i(\mathbf{v}_i - y_i)|} \in (0, 1]$ , while the mislabeled bots still gain more weights. The loss between prediction and ground truth spans between the range of  $[0, 1]$ , and an exponential form ensemble predictor is adopted. This means that the training error after  $T$  iterations is bounded. We name the new boosting algorithm as Boosting through Optimizing Recall (*BoostOR*). The detailed algorithm is shown in Algorithm 4. Figure 21a illustrates how the example weight is updated in AdaBoost model. The mislabeled instances will gain a larger weight in the second round of iteration. While in *BoostOR*, the weight change depends on the labels of the user. If a bot is wrongly predicted, its weight is enlarged. mislabeled regular users are more often ignored.

Since the weight of regular users are reduced instead of removed, trivial solutions will not be easily achieved in real world data, where bots are the minority group. In next section, we empirically prove this with two real world Twitter datasets.

---

**Algorithm 5: BoostOR for Bot Detection**

---

**Input:** The user-feature matrix  $\mathbf{V} \in \mathbb{R}^{m \times k}$ ,  
the user-label matrix  $\mathbf{Y} \in \{1, -1\}^{m \times 1}$  and a weak learner  $h : \mathbf{v} \rightarrow y$   
the initial user weight vector  $\mathbf{D}_1 = (\mathbf{D}_1(1), \dots, \mathbf{D}_1(m))$   
the maximum number of iterations  $max_{iter}$ .  
**Output:** The ensemble classifier  $H(\mathbf{v}) = sign(\prod_{i=1}^{max_{iter}} (\beta_i^{-h_i(\mathbf{v})} - \beta_i^{-\frac{1}{2}}))$   
**For**  $t = 1, \dots, max_{iter}$   
Train weak learner using instance weight  $\mathbf{D}_t$  ;  
Get the trained classifier  $h_t : \mathbf{V} \rightarrow \mathbf{Y}$  and training error  $\epsilon_t$  as Eq. 5.8;  
Calculate the weight  $\alpha_t \in \mathbb{R}$  for  $h_t$  as Eq. 5.6;  
Update user weight as Eq. 5.10;  
**until** Convergence.

---

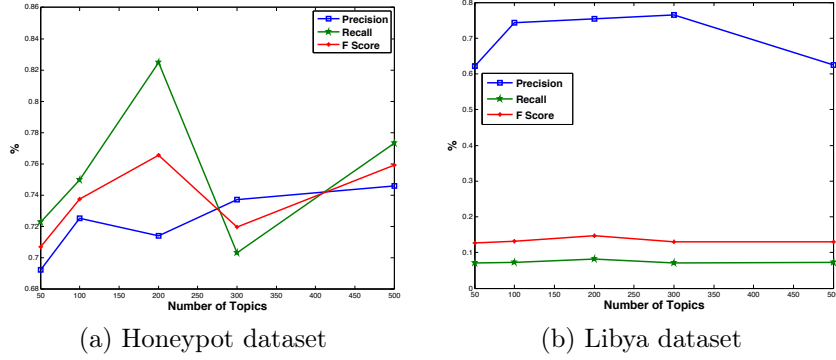


Figure 22: Performance of BoostOR Varying Number of Topics.

### 5.1.5 Experimental Results

In this section we empirically investigate the performance of *BoostOR* with respect to the ability to maximize the  $F_1$  score on bot detection. We perform two experiments: first to show the performance of different algorithms, and second to show how the parameters used in building the AdaBoost and BoostOR models can influence the results of these methods.

#### Bot Classification

We apply each heuristic, as well as AdaBoost and BoostOR to both the Libya dataset, shown in Table 12, and the Arabic Honeypot dataset, shown in Table 13. We find that both AdaBoost and BoostOR outperform the heuristics, with BoostOR performing the best on both datasets. Interestingly, we find that SVM underperforms, achieving a worse result than the heuristics when optimizing the  $F_1$  score.

One thing to note about the heuristics is that they conform to the class distribution. That is, when they achieve their maximum performance they are simply *always*



Table 10:  $Heuristic_{Time}$  Measure on the Arabic Honeypot Dataset.

Truth	Prediction	
	Bot	Human
	Bot	49.9%    50%
Human	0%	0.1%

Table 11:  $BoostOR$  Measure on the Arabic Honeypot Dataset.

Truth	Prediction	
	Bot	Human
	Bot	42.8%    7.7%
Human	14.8%	34.7%

predicting the user is a bot. In other words, the results of heuristic measures indicate that we should delete our entire dataset for both datasets. While this yields what seem to be competitive results, the implication of this approach is not reasonable. While the  $F_1$  score is useful to measure the performance, we need to dig deeper to understand the implication of these results.

To illustrate the difference in performance between the heuristics and our proposed model, we show a confusion matrix of the best-performing heuristic,  $Heuristic_{Time}$ , in Table 10 and a confusion matrix of  $BoostOR$  in Table 11 on the Arabic Honeypot dataset. First, we notice that the heuristic *never* misclassifies as a human as a bot. This is in line with the goals of the industry. However, we see in the  $BoostOR$  results in Table 11 that we achieve superior performance by lowering the false-negative rate of the model, which is in line with the goals of the researcher.

Table 12: The Precision, Recall and  $F_1$  Measure of Different Models on Libya Dataset.

Method	Precision	Recall	$F_1$
<i>Heuristic<sub>URL</sub></i>	6.74%	65.12%	12.21%
<i>Heuristic<sub>Retweet%</sub></i>	7.73%	53.63%	13.51%
<i>Heuristic<sub>Length</sub></i>	7.74%	53.63%	13.51%
<i>Heuristic<sub>Time</sub></i>	7.48%	99.89%	13.91%
<i>SVM</i>	29.24%	8.78%	13.53%
<i>AdaBoost</i>	75.25%	7.48%	13.61%
<i>BoostOR</i>	75.41%	8.14%	<b>14.69%</b>

Table 13: The Precision, Recall and  $F_1$  Measure of Different Models on Honeypot Dataset.

Method	Precision	Recall	$F_1$
<i>Heuristic<sub>URL</sub></i>	49.69%	96.39%	65.58%
<i>Heuristic<sub>Retweet%</sub></i>	50.05%	99.33%	66.56%
<i>Heuristic<sub>Length</sub></i>	50.00%	99.82%	66.63%
<i>Heuristic<sub>Time</sub></i>	49.99%	99.96%	66.65%
<i>SVM</i>	62.41%	62.52%	62.47%
<i>AdaBoost</i>	79.76%	72.41%	75.91%
<i>BoostOR</i>	71.42%	82.48%	<b>76.55%</b>

### Sensitivity of Topic Modeling

In previous experiments, the number of topics,  $K$ , of LDA is set as 200 for all methods. In this section, we study how the number of topics will influence the performance of the proposed model, BoostOR. The change of precision, recall and  $F_1$  score as a function of  $K$  is illustrated in Figure 22. On both datasets, the dimensionality spans from 50 to 500 and the variations of performance are observed.

As shown in the figures, the best performance is achieved when there are around 200 topics. When  $K$  deviates from the optimal value of 200, no matter increasing or decreasing, the performance decreases. Too many topics enable some redundant and

meaningless topics to exist, while some important topics may be neglected if the  $K$  is too small.

#### 5.1.6 Discussion

In this chapter we explore the problem of finding bots on social media. We begin by collecting two datasets, each with different labeling mechanisms. Then we continue to propose a bot detection method that optimizes the  $F_1$  score of the model, which considers recall in addition to precision.

The datasets we use to test this method were labeled through different processes. The first dataset we consider is labeled through using Twitter’s labeling processes of active/suspended/deleted users. The second dataset was obtained by building a honeypot network and collecting the users who follow our honeypot accounts. These datasets encompass different topics and were labeled using two different methods, which make them ideal for testing both the performance and generalizability of our model.

Next, we propose to optimize the  $F_1$  score of the bot detection problem through boosting a basic learner. In order to recall more bots from the dataset, the proposed BoostOR focuses more on the mislabeled bots and downweights mislabeled regular users. The theoretical analysis shows that the optimization can be done efficiently. Experimental results on two real world datasets show that our model outperforms the state-of-the-art bot detection techniques as well as heuristics which are often considered for this problem.

Future work seeks to further improve the recall of the bot detection task. Existing approaches which focus on precision could potentially be modified to improve recall.

We wish to exploit the sparsity of text in order to better separate bots from normal users. Additionally, we will continue to refine the process for collecting labeled social media datasets.

## 5.2 Tampering Bias in Social Data Streams

Social media has become a very active research area in recent years. One limiting factor for many researchers is the amount of data they are able to collect. Many social media sites strictly limit or outright forbid the collection of data on their sites for research purposes. One major social media site, Twitter, stands out among these sites for its willingness to share data with researchers. Twitter does this mainly through its real-time feeds, called the “Streaming” APIs. There are three main APIs that researchers can access directly: 1) the “Filter API”, which gives the user a directed sample based on the input of the user, 2) the “Sample API” which returns a random 1% sample of all public tweets generated on Twitter, and 3) the “Firehose” which yields 100% of all public tweets. While the Firehose seems like the obvious monetary cost of using this API as well as the server requirements for hosting the data prevent most researchers from being able to use this option. The Filter API and the Sample API can be used by researchers worldwide to download tweets in real-time, collecting at most 1% of all of Twitter’s public statuses, or “tweets” completely free of charge (Morstatter et al. 2013).

The Filter API and the Sample API are appealing to researchers, and it is important that researchers understand the underlying mechanisms of these APIs so that they can have confidence in the data collected through them. Biases introduced in the data collection process can propagate to the research results. For instance, bias has

been found in Twitter’s Filter API. By comparing the results of a Firehose crawl with a Filter API crawl, the authors of one study (Morstatter et al. 2013) discover that bias in the way tweets are sampled yields completely different statistics, *e.g.* the top hashtags in the data and the topics in the text.

While evidence of bias has been found in the Filter API, studies on the Sample API have been largely positive, with indications that this data outlet is unbiased. By empirically studying the data, one study (Morstatter, Pfeffer, and Liu 2014) found that the results were not statistically significantly different than the results of the Firehose. Another study (Kergl, Roedler, and Seeber 2014) discovered the underlying mechanisms behind how Twitter samples the data in the Sample API. First, they show the different components that make up the identification number (ID) for each and every individual tweet. A portion of this ID contains the millisecond-level timestamp when the ID was generated. The authors find that the Sample API selects the tweets based on this timestamp, with any tweet whose ID was generated in the millisecond range of [657, 666] to be selected by the Sample API. Twitter, assuming that the millisecond-level information of the ID’s timestamp is random, chooses this mechanism to distribute their tweets. This sampling mechanism seems reasonable. Humans don’t have millisecond-level control over the tweet’s timestamp. The muscle movements to click a mouse or tap a phone screen combined with the network delay make the timestamp nearly impossible for a human to predict at the millisecond-level granularity.

Unfortunately, many of the users of Twitter are not humans, but bots: accounts controlled by computers. Estimates of the number of bots on Twitter range from 5-9% (Elder 2013). Many of these bots have malicious intent: using their ability to generate massive amounts of noise to try to skew statistics of the data (*e.g.* sentiment

towards particular topics, top keywords). Bots have the ability to control the *exact* time their tweets are posted, allowing them to control (to a certain extent) the time that Twitter receives their tweet. If bots are able to control if their tweets appear in the Sample API, it could affect the integrity of this stream: bots could modify the statistics of this data stream, affecting both research and applications that depend on this stream.

In this chapter we evaluate two questions: 1) “can the millisecond-level information of the timestamps associated with tweet IDs be anticipated?”, and 2) “can we leverage this to build software which generates tweets that have a greater chance of being included in Twitter’s Sample API?”

### 5.2.1 Anticipating the Sample API

Here we investigate whether the tweet ID’s millisecond-level timestamp information can be predicted. In this first experiment we see how the millisecond-level information of the tweet ID corresponds to the millisecond at which the tweet was sent. To do this, we send tweets at each whole millisecond in the range  $[0, 990]$  in intervals of 10. For each tweet we send, we observe the millisecond-level information of the tweet’s ID. We extract this using the process outlined in Kergl, Roedler, and Seeber 2014, which identifies specific bits in the binary representation of the ID that contain this info. We repeat this experiment 6 times, and plot the results in Figure 23. We observe a strong linear relationship between the time the tweet is sent and the millisecond ID of the tweet ID. Furthermore, we find that the millisecond delta is consistent with an average delta of 171ms. Using this information we devise an approach that can

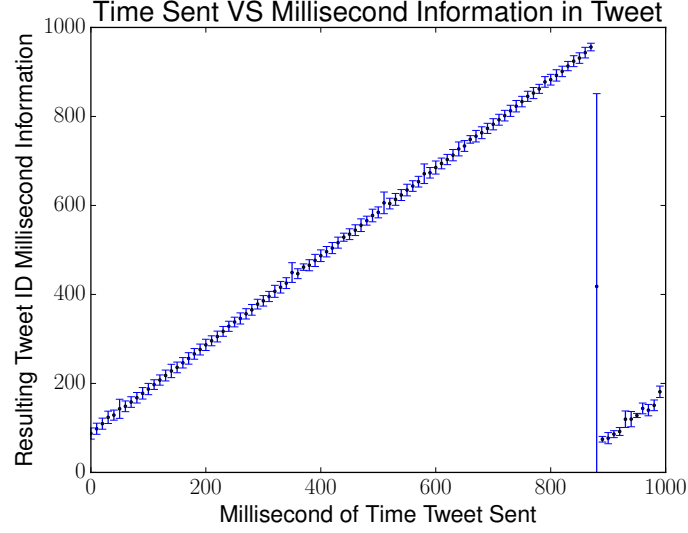


Figure 23: Millisecond of Publish Time from Tweet ID as a Function of the Millisecond When the Tweet Was Sent. The Large Error near the 850 Mark Is an Artifact of the Modulus Property of the Reading.

generate tweets with a greater probability of landing in the Sample API in the next section.

### 5.2.2 Manipulating the Sample API

In the previous section we showed empirically that the millisecond-level timestamp information of a tweet ID is a function of the time the tweet is posted. Using this insight, we design an algorithm that adaptively adjusts the millisecond lag in order to maximize the amount of tweets that appear in the Sample API. The pseudocode for this algorithm is shown in Algorithm 6. The essence of this algorithm is that it adaptively measures the network lag between when the tweet is sent and when the tweet is received by Twitter, measured by the ID of the tweet. We use a moving

average of the last  $w$  tweets to estimate the lag. Details on how to extract  $i$  from the tweet ID can be found in Kergl, Roedler, and Seeber 2014.

To measure the effectiveness of our algorithm, and by extension the possibility to bias the Sample API, we conduct 5 trial runs. In each experiment we produce 100 tweets. We use  $w = 3$  in our experiments. We set  $m = 661$  as it is the midpoint of  $[657, 666]$  and should give us the greatest chance of hitting the window where the Sample API selects its tweets. Also, we set  $\hat{\delta}_{init} = 491$  as it is the expected delta from 661 as we discovered in the previous section. We achieve  $82 \pm 9\%$  coverage, contrasted with  $1 \pm 0\%$  coverage when tweets are produced under normal conditions.

---

**Algorithm 6:** Maximize the Probability of a Post Being Selected by the Sample API.  $\mathbb{1}(\cdot)$  Is the Indicator Function.

---

**Data:**  $w$ : window size;  $m$ : target millisecond;  $\hat{\delta}_{init}$ : initial delta  
 $h \leftarrow$  empty list;  
 $target \leftarrow m - \hat{\delta}_{init}$ ;  
**while** *true* **do**  
    wait until the  $target$ -th millisecond of the next second;  
    post a tweet,  $t$ ;  
     $c \leftarrow$  current time in milliseconds;  
     $i \leftarrow$  time in milliseconds from tweet  $t$ 's ID;  
    append  $i - c$  to the end of  $h$ ;  
     $\hat{\delta} \leftarrow \frac{1}{w} \sum_{k=|h|-w}^{|h|} h[k] \bmod 1000$ ;  
     $target \leftarrow 1000 \cdot \mathbb{1}(m - \hat{\delta} < 0) + m - \hat{\delta}$ ;  
**end**

---

### 5.2.3 Discussion

Twitter's Sample API is a popular tool among researchers in the area of social media mining. The results of this chapter show that by timing the creation time of tweets, bots and spammers can have a large impact on the information that is provided



through the Sample API, effectively injecting bias into this data outlet. This is a major problem for users who wish to ensure that their data is a representative sample of the real activity on Twitter. The approach presented in this chapter is intended to give researchers an understanding that despite recent findings that say that the Sample API is unbiased, results should be taken with care when this data outlet is used in analysis. Furthermore, social media sites can use these results to improve the design of their APIs. API designers can surpass this issue by adding random bits to the timestamp portion of the ID, or by pre-generating IDs before distributing them.

### 5.3 Bias from Sophisticated Actors: The Case of Shills

There is ample evidence to suggest that social media acts as a powerful tool for campaigns. In the realm of politics, social media can spread political information (Tufekci and Wilson 2012), build political brands (Harfoush 2009), energize supporters (Juris 2012), and to reach and persuade potential voters (DiGrazia et al. 2013; Tumasjan et al. 2010). Politicians and political activists have employed many different social media techniques to spread their message. Most recently, both major parties' presidential campaigns used social media to spread information as well as to lambaste political opponents. What these techniques all have in common is that they are overt, where users of a social media site know who the political players are and can watch and contribute as their dialogue and messaging plays out online.

We live in an era where opinion manipulation is rampant online. There are many *covert* groups that attempt to sway the opinion of real people by manipulating and misrepresenting facts in order to make one side of a debate look more prominent than it actually is. There are many ways for this to manifest online. One is through the

presence of bots, automated accounts who push certain hashtags or opinions to the top of trending lists (Lee, Eoff, and Caverlee 2011; Chu et al. 2010). Because these accounts are automated, they do not contain rich original text like normal users, a weakness that can be leveraged to detect them in the real world. Another covert strategy is the deployment of “crowdturfers” (Lee, Tamilarasan, and Caverlee 2013). These are real people who are hired *en masse* by campaigns to generate text on their behalf. Because they are not experts on the topic they discuss, they can often be detected due to their novice nature (Ott, Cardie, and J. T. Hancock 2013). Both of these covert types of users can quickly spread opinions and propaganda quickly throughout a network.

Recently, a covert technique has gained prominence in swaying opinion in social media, “shills.” The first mention of online shills was in 1999 <http://partners.nytimes.com/library/tech/99/10/circuits/articles/14spin.html>. In 2016, they were used in the U.S. elections.<sup>30</sup> Shill accounts promote their campaign by directly engaging with those holding opposing opinions. These users complement the work done by bots and crowdturfers by suppressing and distorting opposing information on social media. Instead of creating new posts, these accounts usually do this in the form of directed replies to the opposition. This is done to delegitimize users who are spreading information that is contrary to the goals of their campaign. Shills directly engage with other users, so they must be able to directly address the points of a contrary opinion. This requires greater sophistication than bots can provide, and a deeper level of training than is provided to crowdsourced workers (Lee, Tamilarasan, and Caverlee 2013). Shills are professional users employed by campaign organizers

---

<sup>30</sup><http://www.thedailybeast.com/articles/2016/04/21/hillary-pac-spends-1-million-to-correct-commenters-on-reddit-and-facebook.html>.

who seed these users with talking points and facts and then ask them to go and engage with users holding differing opinions on social media sites. The requirement that shills be paid professionals means that there are relatively few of them in the wild; however, these users make a lot of noise.<sup>31</sup> Following the 1:10:89 rule for content creation on social media (Shirky 2008) we know that it takes very few users to affect change online.

Shills, while a small contingent of the site, can greatly affect a social media platform. If the users do not trust the content on a site to be from genuine users, they may go elsewhere to obtain trustworthy information. If shills are left to run amok, then their combative nature can prevent those with contrary opinions to theirs from posting, reducing the authenticity of discourse on the site. If shills have their way, they will substantially skew the content of a social media site, which skews statistics performed on data from that site. It is critical that we find these shills in order to ensure that the discourse that is carried out on social media is organic, free of paid promotional content for any idea or candidate.

We propose a framework to categorize shills in online social media. Focusing on one prominent social media site, Reddit, we develop a process that can describe the behavior of shills. We demonstrate its efficacy by showing that it can be used to differentiate shills from real users using real-world data.

### 5.3.1 Social Media Shills

The general term “shill” denotes an enthusiastic accomplice (Green 2011). The term initially arose to describe a person who helps a salesman by drawing excitement

---

<sup>31</sup>The shills in our study post 23% more content than other users.

for a product. Historically, when salesmen would come to a village to sell a product they would place a shill in the crowd. After the salesman had described the product, the shill would then speak up and loudly announce that the product had indeed worked as advertised. In order for the shill to be effective, it is paramount that the link between the salesman and the shill not be publicized: the shill must be perceived as an ordinary individual. In doing so, the unsuspecting audience believes that the product actually has merit, and will buy it hoping to have the same results.

This phenomenon has made its way into the online world (Steiglitz 2007), with modern shills acting on behalf of their employers to encourage certain opinions in social media. Online shills are users that are paid by a campaign or organization to help manage their image online. Shills combat opposing information that is spread on social media. This can be in the form of replies that prevent an opposing narrative from taking root and spreading on the site. Further, shills behave differently than true political supporters as they focus more on combating opposing opinion than spreading promotional or supportive content. “Correct the Record,” an organization which employs shills, states that it has “addressed more than 5,000 people that have personally attacked Hillary Clinton on Twitter.” Shills behave in a different manner from other covert users such as bots and crowdturfers, so it is unlikely that any existing approach will be effective in identifying them. We attempt to identify patterns that can be used to discover shills.

### 5.3.2 Collecting Data for Studying Shills

The data we collect pertaining to shill accounts comes from Reddit<sup>32</sup>, where we crawled and labeled users posting in a politically-active forum on the site. In this section we describe the process by which we collected the data, and annotated the users as shills. Finally, we provide some analysis on our labels to show that while we may never be certain if the user is a shill, that they are exhibiting the properties that we are looking for in shill detection.

Reddit is a large social media site where users share and comment on links. On Reddit, posts are submitted to “subreddits”, which are groups of posts organized around a common theme or interest. The scope of interest is initially set by the organizers, or “moderators”, of the specific subreddit, however the content and general discussion is provided by the people who read the subreddit, the “subscribers”. Users on Reddit refer to subreddits with the `/r/` prefix to denote that they are talking about a subreddit. For example, if a user were to mention the bernie sanders subreddit, they may include “`/r/sandersforpresident`” in their text. This convention is encouraged by the site, which automatically links to the subreddit when it is mentioned in this way.

Given shills’ nature, we hypothesize that shills most commonly post in politically-oriented subreddits. In the context of politics, subreddits exist at every granularity. The “`/r/politics`” subreddit being the most general, inviting users from any political orientation to post to the subreddit. Furthermore, there are entirely different subreddits specifically designed for political candidates (e.g. `/r/the_donald`, `/r/hillaryclinton`), as well as subreddits for specific interests (`/r/timetolegalize` for marijuana legalization, `/r/environmental_policy` for discussion regarding climate protection, etc.). We selected

---

<sup>32</sup><http://www.reddit.com>

the **/r/politics** subreddit to crawl due to its generality. Because this subreddit welcomes discussion from all political views, it contains many highly polarized groups each supporting different major candidates within. In addition, this is the most likely to invite shill activity: since people from all backgrounds are there they are more likely to reach audiences outside of the ones that the shill agrees with; and we are also more likely to find diverse opinion to trigger the shills into posting responses. Furthermore, **/r/politics** is one of the largest subreddits on Reddit with over 3 million subscribers, so shills may also be motivated to choose this subreddit for the large audience which will read their posts.

We collected posts from the **/r/politics** subreddit the beginning of April, 2016 through mid-June 2016. For all of the links submitted to the subreddit, we gathered all of the users who commented on or replied to any of them and collected the most recent 1,000 comments and replies from each user. This data was crawled by using Reddit’s REST API<sup>33</sup>.

After collecting the data, we assigned ground truth labels for the users: “shill,” and “not-shill.” We employed three human annotators to label the user accounts. We took a random sample of 185 users to annotate. In our sample, we only considered users that posted at least 10 times. For a given user, the annotator was provided with a text file containing at most the most recent 1,000 replies made by that user, with each reply occupying a single line of the file. Subreddit information, timestamps, and the user’s screen name are not provided. After reading all of the 1,000 replies by the user, the human then made the assignment based on the following criteria:

1. Did the user’s replies entirely, or almost entirely, support one candidate?
2. Did the user’s posts generally contain claims to support their arguments?

---

<sup>33</sup>Adding “.json” to the end of any Reddit URL yields JSON representation of that page.

### 3. Did the user explicitly mention a tie to any campaign?

It is important to note that for criterion 2 that the claims purported in the replies did not need to be backed by citations, nor did they need to be true. All that was required was that the user’s reply be supported by claims. If the annotator could answer “yes” to the first two criteria, and “no” to the third, then the annotator would mark this user as a shill. We took the intersection of the three sets of users identified as shills by each annotator.

The labeling process for this problem is very difficult as there is no underlying ground truth in the data. Despite our robust, multi-person approach to labeling shills, we still do not know if they are actually shills or just people who are extremely passionate about defending their candidate in online forums. Instead, we follow the labeling approach that is taken in bot detection research (Chu et al. 2010; Lee, Eoff, and Caverlee 2011), where we can at best be highly suspect that a user is a shill. Due to the large amount of time required to annotate each user, we limited our labeling to 185 users. The dataset will be shared upon request.

Statistics of our dataset are shown in Table 14. Surprisingly, the three annotators found a total of 17 shills within the dataset. This means that 9% of all of the users we labeled are shills. This number is strikingly high. We attribute this high number to the nature of the subreddit we crawled. /r/politics is very likely to accrue a large amount of diverse and argumentative political discussion, the likes of which are prime for shill activity. Furthermore, since /r/politics is so large, shills may deem it more important to address posts created in this venue. Almost assuredly, this number would be much smaller on another subreddit.

Table 14: Dataset Statistics

Property	Value
Users	9,379
Posts	6,754,832
Types	3,361,118
Tokens	120,341,645
Replies	6,754,832
Labeled	185
Labeled Non-Shills	168 (91%)
Labeled Shills	17 (9%)

### 5.3.3 Characterizing Shills

Shills are topic-focused, keeping their discussion limited to very specific political examples, and they are more interested in politically-oriented topics than non-shill users. Using these observations, we introduce a preliminary approach to characterize them based upon their topical engagement. We extract features to help a machine learning classifier differentiate shill accounts from real users.

### 5.3.4 Shill Characteristics

Many of the features we use are extracted with the help of Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003). LDA discovers underlying “topics” in a corpus. In addition to the corpus, the algorithm takes “ $K$ ,” the number of topics the algorithm is to learn. With this input, LDA learns the following:

1. Topics. These are represented as ranked lists of keywords over the vocabulary of the corpus, where the rank is determined by the words probability of occurring



in the topic. We represent these lists as  $\mathbf{T}$ , where  $\mathbf{T}_j^i$  is the probability of word  $i$  occurring in topic  $j$ .

2. **Document  $\times$  Topic affinity matrix.** In addition to the topics, the model also learns each document's affinity for each topic. LDA learns the probability that each document is generated from each of the  $K$  topics. This is represented as  $\mathbf{D}_j^d$ , the affinity of user  $d$  for topic  $j$ .

We run LDA over each dataset, treating users as documents. Each document's text is a bag of words containing all of the text from that user's posts.

Going forward we list the features based upon each of our aforementioned observations. We will describe the features from the perspective of one user,  $u$ .

1. **Content Features.** We extract features pertaining to the text that the user writes in their posts.
  - a) *Topic Affinities [K features]*. These consist of the  $\mathbf{D}^u$  topic affinity features from the results of LDA. This will help us to understand the degree to which a user focuses on specific topics.
  - b) *Affinity Entropy [1 feature]*. Shills are likely to be focused around a particular topic. Entropy (Gray 2011) is a measure of the randomness of a probability distribution, with a greater entropy indicating a greater degree of randomness. It is computed as:

$$h^u = - \sum_{k=1}^K \mathbf{D}_k^u \log_2 \mathbf{D}_k^u, \quad (5.11)$$

which yields a single feature measuring the users focus around any specific topic.

2. **Account Features** We extract features that represent both the user’s activity on the site, as well as how their content is perceived. These features are as follows:

- a) *Daily Post Rate [1 feature]*. The number of days in which the user has posted on the site, out of the total days they are present in the last 1,000 comments. *E.g.*, if the users last 1,000 comments appear during 230 unique calendar days, then their score will be 4.35.
- b) *Subreddit Entropy [1 feature]*. Reddit has natural mechanisms under which users can categorize their content, “subreddits”. Subreddits are boards where users post content pertaining to the underlying theme of the subreddit. We measure the frequency that each user posts in each subreddit, and then compute the entropy of the probability distribution. This measures the diversity of the subreddits that the user posts in. Treating the frequency of posting in each subreddit as a probability distribution, we compute the entropy, or focus, that the user has for the subreddits.
- c) *Hour Probability [24 features]*. This measures the probability that each user posts in a given hour of the day. It may be that hired shills work according to a schedule, and by calculating the probability at each hour we may be able to learn their schedule. This consists of one feature for each hour of the day.
- d) *Hour Entropy [1 feature]*. This is the focus of the probability distribution above. It is possible that the user may only log in for a brief period of time each day to do their shilling. To capture this, we measure the focus of the user around any hour of the day.

3. **Community-Based Content Features** The relative difference between the

topics they focus on compared to the rest of the users in the dataset may be equally important. In fact, in our investigation, we learned that shills tend to oppose opinions in social media, meaning that they may focus on different topics than those they are conversing with. Thus, we experiment with two feature groups that can provide information about how different the user is from the rest of the group.

a) *Focus Topic Significance* [ $K$  features]. This is measured as the  $z$ -score (Casella and Berger 2001) of the difference between the focus of the user on a topic against the focus of the community on the topic. For a topic  $k \in [1, K]$ , the feature is calculated as:

$$z_k^u = \frac{\mathbf{D}_k^u - \mu_k}{\sigma_k} \quad (5.12)$$

where  $\mu_k = \frac{1}{|U|} \sum_{j \in U} \mathbf{D}_k^j$  is the average focus of a user on topic  $k$ , and  $\sigma_k = \sqrt{\frac{1}{|U|} [\sum_{j \in U} \mathbf{D}_k^j - \mu_k]}$  is the standard deviation.

After feature extraction, each user is described by a total of:  $K$  topic affinity features + 1 topic entropy feature + 27 account features +  $K$  topic significance features =  $2K + 28$  features. In addition to measuring the performance of the classification algorithms on the model, we will also investigate the impact of  $K$  on the results.

An important note about this set of features is that while not all features are specifically designed to measure “shillness,” some are. Specifically, the “Focus Topic Significance” is designed to measure this. Consider the case of cross validation, which is what we use to select the models. If the shills in the training set have a particular relative topic affinity, then that feature maybe picked up by the classifier in order to identify further shills. We validate this finding in the feature importance section.

Table 15: Confusion Matrix Showing the Average Cosine Similarity between the Two Groups of Clusters.

	Shill	Not Shill
Shill	0.53	0.66
Not Shill	0.66	0.61

#### Feature Analysis: Clustering

We have collected a dataset of shills and non-shills and have proposed a set of features to formally describe these users. We investigate how different these two classes of users are. By setting  $K = 50$ , we investigate the separability of this dataset. We look at the clusters that are formed by the labels. We compute the average cosine similarity between all of the users in the shill class, and all of the users in the nonshill class, and the average distance between the two classes. The results of this measurement are shown in Table 15. Here we see that the within-cluster distance for each class is smaller than the between cluster distance. This means that the users in each cluster are more similar than the users between clusters, which is a promising indication that there is a difference between the two classes.

#### Feature Analysis: User Behavior

Analyzing the key difference that describe the shill behavior versus non-shill behavior was done using the non-parametric Wilcoxon rank-sum test (Mann and Whitney 1947). The Wilcoxon rank-sum test ranks all values in two different groups then finds whether one group ranking tends to be statistically different than the other group ranking. The most significant shift was captured in the samples between the

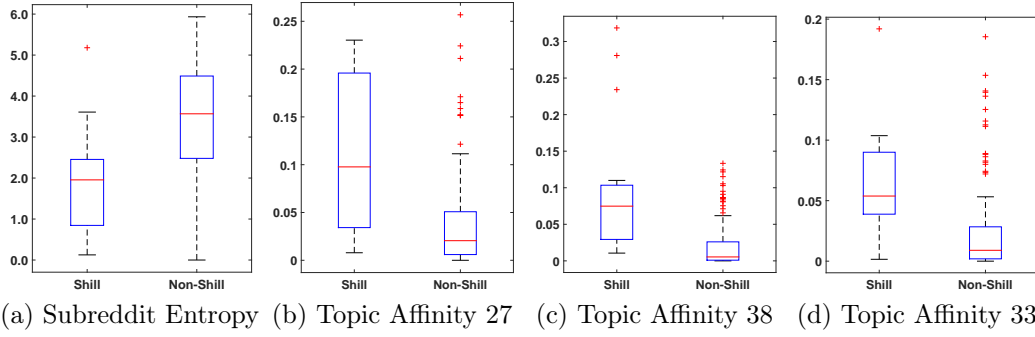


Figure 24: The Distribution of Feature Values for Features Identified as Most Important by the Wilcoxon Rank-Sum Test.

shill and the non-shill groups in four different features (Subreddit Entropy, Topic Affinity 38, 27, and 33) with  $\alpha < 0.005$ . Figure 24 provides a visual description of how shill behavior differs from non-shill. First, shills tend to demonstrate higher topic affinity than non-shills. Second, shills demonstrate less Reddit entropy than non-shills. Thus, inference into behavioral patterns given the statistical analysis leads to a conclusion that shills tend to focus on topics relating to specific keywords such as “Hillary”, “Clinton”, “Bernie”, and “Trump” discussions and seldom vary to other non-political topics. Going forward, we will propose a classifier to separate the two classes.

### 5.3.5 Identifying Shills

In this section we apply the data and feature extraction approach to build a classifier that can leverage the labeled data to differentiate these two classes. After building the classifier, we investigate how the classifier makes its decisions, what features it determines are most important, and the source of classification errors.

### 5.3.6 Skill Detection

We test various approaches in a standard machine learning framework. We use 10-fold cross validation. We report the average across the 10 runs. We compare several classifiers, organized into four groups:

- **Baselines.**
  - i. **Always-Skill:** Assigns the “skill” label to every user.
  - ii. **Random:** Randomly assigns a label to each instance following the distribution of labels in the training data.
  - iii. **Clinton:** Because we are studying pro-Clinton skills hired by Correct the Record, we see if simply mentioning the candidate reflects skill behavior. We denote any user that mentions “Clinton”, case insensitive, in over 10% of their posts as a skill.
- **Activists.** A classifier that strives to find activists in political campaigns (Ranganath et al. 2016). This is included to compare the task of skill detection to identifying political activists.
- **BoostOR.** A recent classifier that detects bots (Morstatter, Wu, et al. 2016), with code made available by the authors. This is to help understand the differences between skill detection and bot detection.
- **Skill Classifiers.** Classifiers trained with the features described in the “Characterizing Skills” section. We choose algorithms which produce interpretable models in order to estimate the importance of different features.

Using the classification setup described, we test several classifiers for their ability to differentiate the two classes. Due to the nature of the data, we do not evaluate

deep learning classifiers. The results are shown in Table 16. These results can be interpreted in several ways. First we see that the best classifier is Logistic Regression. When this classifier identifies a shill, it is correct slightly under half of the time. The recall is 45%, meaning that we detect just under half of the shills present in the data. This elucidates the difficulty of the problem. Shills are people writing original text; it is challenging to differentiate them from political enthusiasts.

Another result is the performance of the activist classifier proposed by Ranganath et al. 2016. While their proposed approach is very strong for identifying advocates, it does not perform well at identifying shills. This demonstrates that finding advocates is a fundamentally different problem; advocates display different behavioral patterns.

Table 16: Classification Results.

Classifier	Precision	Recall	F <sub>1</sub>
Random	0.050	0.050	0.050
BoostOR	0.183	0.150	0.165
Always-Shill	0.090	1.000	0.165
Clinton	0.112	1.000	0.201
Activists (Ranganath et al. 2016)	0.225	0.500	0.310
SVM	0.325	0.350	0.337
Decision Tree	0.335	0.550	0.417
Logistic Regression	0.475	0.450	<b>0.462</b>

## Sensitivity to $K$

Many of the features we investigate are based upon the topics generated by LDA. Thus far we have set the number of topics to  $K = 50$ . Here we investigate the classifier’s sensitivity to  $K$  by varying it and seeing the results.

We vary  $K$  from 5 to 500, and plot the results in Figure 25. We see that initially more topics are good, but after  $K = 50$ , more topics begin to add noise to the model and actually decrease performance. Informed by these results, we conduct all future experiments with  $K = 50$  topics.

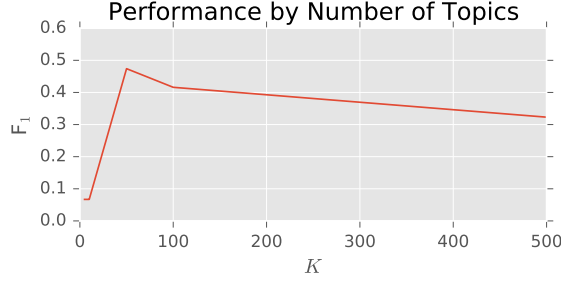


Figure 25:  $F_1$  As a Function of the Number of Topics Used to Create the Dataset. Here We See That the Performance of the Model Peaks at  $K = 50$  Topics.

### Feature Importance

In the above experiment, we evaluate several classification algorithms for their ability to discern between the two classes. We provided the features we used to make these classifications along with their justification. In this section we will rank the features based on their predictive power to understand what facets are actually contributing to the classification.

To rank the features we perform a *feature ablation* test, where we run the classification algorithm once for each feature to test its importance. In each run, we test the feature by assessing the  $F_1$  performance of the classifier with all features ( $p_1$ ) and then re-assessing the performance with that feature removed ( $p_2$ ). The difference in performance,  $p_1 - p_2$ , is used as that feature’s predictive power.

The most important features identified by this experiment can be seen in Table 17. These results show two important observations about skill behavior. First, skills tend to focus on topics differently than regular users. We know this because the top features used to identify them are topic focus significance scores, as well as the topic affinity entropy. This means that not only do skills focus on different topics than non-shills,



Table 17: Most Important Features in Our Dataset. These Are the Features Which Effected Performance by at Least 3%.

Feature	Top 5 Topic Words	$p_1 - p_2$
Subreddit Entropy	<i>(Not Applicable)</i>	0.194
Topic 38 Affinity	politicaldiscussion, sanders, clinton, bernie, vote	0.056
Topic 27 Affinity	politics, trump, hillary, bernie, people	0.039

but they also are focused on just a handful of topics. We know this because the topic entropy score helped to separate the two classes.

## Error Analysis

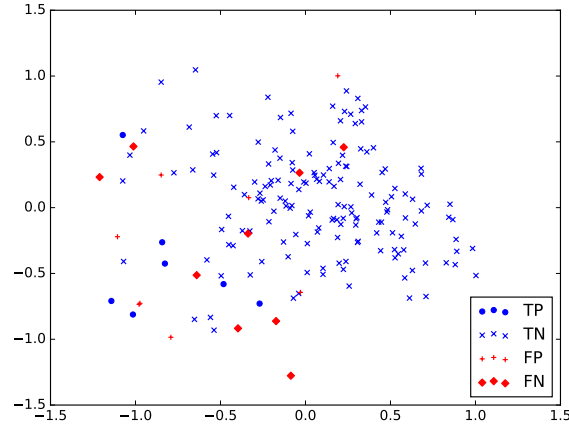


Figure 26: Low-Dimensional Embedding of the Features with Tags Comparing Their Real Label with That Assigned by the Classifier. TP, TN, FP, and FN Are “True Positives”, “True Negatives”, “False Positives”, and “False Negatives” Respectively.

To look at the differences between correctly-classified, and incorrectly-classified instances, we plot a low-dimensional embedding of the data. Using Principle Component Analysis (PCA), we extract the first two principle components of the data, and

plot each user according to their position along these two dimensions. Additionally, we tag each instance based upon how it was classified, as either a true positive, true negative, false positive, or false negative. This plot can be seen in Figure 26. Shills tend to appear in the lower-left corner of the figure. Furthermore, almost all of the true positives occupy this space with no true positive occurring outside of the lefthand side of the figure. That is not to say that the classifier does not attempt to predict shills outside of this region, however it does so incorrectly. False positives are present throughout the main cluster of users at the left side of the plot. This tells us that the rich and thought-out language used by shills enables them to camouflage themselves and avoid detection. Another important distinction is that the classifier finds many false positives in the bottom-left, meaning that the presence of shills with this feature value can cause the classifier into identifying real users as shills in these instances. Overall, the mixture of the users is a problem, with no clear separation based on the PCA position. This is likely due to the similarity between shills and activists.

### Identifying the Most Informative Shills

Our dataset contains very few positive instances, so the burden of providing information about the positive class falls upon just a few instances. As such, it is likely that there are a few users which help the classification. To measure the impact of each individual user, we repeat the experiment above, except this time we hold out a shill user from the dataset, and perform 10-fold cross validation again. We then compare the performance with the performance of when every user is present.

We found that for 5 users, holding them out individually will drop the  $F_1$  by more than 10.0%, meaning that their content strongly helps the classifier to identify

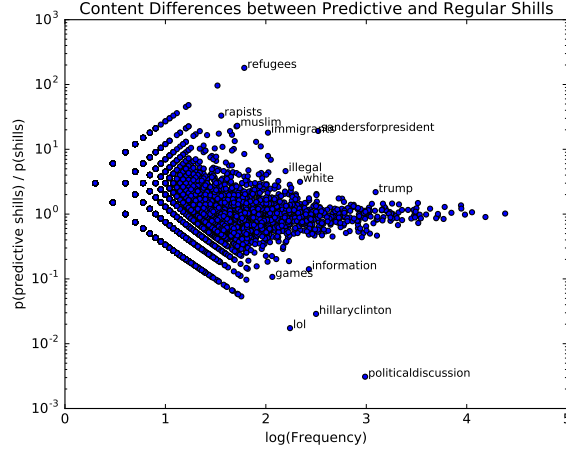


Figure 27: Differences in Word Choices between Normal Shills and the Shills That Add the Most Predictive Power in the Model. The  $X$ -Axis Is the Frequency of the Word in the Corpus, and the  $Y$ -Axis Is the Probability That a Predictive Shill Uses This Word, Divided by the Probability That a Regular Shill Uses It. Words above  $10^0$  Are Preferred by the Predictive Shills, and Words below This Line Are Preferred by Normal Shills. We Annotate Words That Are Used Significantly More Than the Other Class.

patterns within the shill class.<sup>34</sup> This indicates that individual users do make a large impact on the classification.

In light of these findings, it is natural to wonder what content drives these particular users to give such a strong description of this class. In order to see how these users are different from the rest of the shills. We plot the frequency of each word on the  $x$ -axis, and its relative prevalence among the “predictive shill” users on the  $y$ -axis in Figure 27. This plot shows us that the most predictive shills focus disproportionately on hot button issues such as immigration. Furthermore, the super shills also mentioned /r/sandersforpresident at a disproportionately higher rate than other shills, while other shills focused on mentioning /r/hillaryclinton, and more informal words such as

---

<sup>34</sup>If all 5 of these users are held out together, the total drop in  $F_1$  performance is 17.0%.

“lol” and “games” in their text.<sup>35</sup> While shills in general stay away from these words, these less-predictive shills use these words disproportionately more, meaning that they may be less trained than the more predictive shills.

### Comparing Shills with Other Covert Users

In the previous section we have proposed a method to identify shills. This method works primarily due to the way the shills are characterized, which enables their behavior to be captured by off-the-shelf algorithms. In this section, we investigate whether the patterns identified by our method are specific to shills, or if their behavior is also shared by other classes of covert users.

To make this comparison we test our framework at the task of bot detection. Using a sample of bot users identified in the complete dataset of Reddit comments,<sup>36</sup> in May 2015, we collected the entire history of these 43 users. In order to make these results comparable, we ensure the same class distribution as in the shill detection task, collecting non-bot users. We verified their status as humans by comparing their usernames with a list of known bots,<sup>37</sup> and manually verifying their text for the presence of bot activity (users posting repetitive, or nonsensical content were removed). After our preprocessing, we were left with 432 non-bot users.<sup>38</sup>

---

<sup>35</sup>In this case, the word “games” is used in the context of the “games” that other candidates are playing, and not in the context of video games.

<sup>36</sup>[https://bigquery.cloud.google.com/table/fh-bigquery:reddit\\_comments.2015\\_05](https://bigquery.cloud.google.com/table/fh-bigquery:reddit_comments.2015_05)

<sup>37</sup><https://www.reddit.com/r/autowikibot/wiki/redditbots>

<sup>38</sup>A list of users from both classes is available at <https://goo.gl/PPsR8E>.

Table 18: Results of Different Approaches on the Bot Detection Task. “Shills” Is the Logistic Regression Approach for Detecting Shills.

<b>Approach</b>	<b>Precision</b>	<b>Recall</b>	<b><math>F_1</math></b>
Shills	0.09	0.75	0.16
Always-Bot	0.09	1.00	0.17
BoostOR	0.68	0.58	0.63

Taking this list of users, we extract features from each user as described in the “Categorizing Shills” section. We use 10-fold cross validation. We report the average across the 10 runs. We use the classification approach identified as the best within the skill task, the Logistic Regression classifier. The results of this experiment are shown in the “Shills” row of Table 18. While our skill classifier is able to learn meaningful patterns to separate shills from non-shills, it cannot learn patterns to separate bots from non-bots.

To give a sense of the performance of the classifier, we compare with two other approaches. The first is a baseline that simply classifies every user as a bot (“Always-Bot” in Table 18). The other is a recent bot detection approach, BoostOR (Morstatter, Wu, et al. 2016). The skill classifier performs worse than a simple baseline approach on this task. BoostOR is able to significantly outperform these baselines, showing that this task is feasible when using an approach that is designed for bot detection. This shows that the skill classifier’s performance is not low due to the dataset being too difficult, but instead because the skill feature set is not fit for bot detection. This means that bot detection is a different problem from skill detection.

### 5.3.7 Discussion

In this chapter we present the novel problem of shills in online social media. We discussed how they can harm the authenticity of social media sites by injecting paid content that favors their cause. We discuss how shills operate, and how their behavior in the online space differs from similar tasks such as bot detection and finding covert campaigns in social media. To back these observations, we collected a dataset of users from /r/politics, a political Reddit forum, and labeled a selection of the users as shills or as people. We find that 9% of all of the users on this forum are shills, a strikingly high number. Based on our observations, we define a strategy to characterize these users in a way that allows for automated approaches to identify them. Next, we demonstrate the efficacy of this strategy by showing that we can build a classifier to separate these instances. Our  $F_1$  performance in this task is superior to several baselines. It is also superior to the state-of-the-art from activist detection, indicating that identifying shills is a different problem.

After building the classifier itself, we investigated how it makes its predictions. We found that the most discriminative feature was “Subreddit Entropy,” a feature that measure the diversity of Subreddits the user posts to. This is an important finding as it shows that shills are unable to behave as regular users due to the constraints imposed on their time while shilling for their cause on the site. We also found that the classifier learns a handful of topics that are focused upon by shill users. These topics are politically-oriented with words like “trump,” “sanderson,” and “law” ranking highly in these topics. When LDA is trained with 50 topics, it yields the best performance.

We observed that not only are some features very important in the classification, but so are some users. We found that for 5 of the 17 shills, removing one of them will

reduce the classifiers  $F_1$  performance by more than 10%. These users are providing a significant amount of information to the classifier. By plotting their content in relation to the content of other shills, we observe that they focus more on words pertaining to hot-button issues, such as immigration. Also, they tend to talk more about other candidates.

Shill detection is an important problem because shills can change the discourse in an online platform. Furthermore, shill detection is a difficult problem because shills are real people who can perform certain activities to evade detection systems. Future work seeks to improve the performance of the classification task. Other areas of work could be to use the shills, once discovered, to better understand the leanings of a political campaign. Also, shill detection may be more effective on social media sites that offer richer features, such as a social network. Furthermore, we noticed that shills who tend to focus on hot-button issues tend to give the most information to the classifier. Focusing on these users during data collection may yield better results.

### PERCEPTUAL BIAS IN SOCIAL DATA: CROSS-PLATFORM EMOJI MISINTERPRETATION

Perceptual bias refers to when an entity matches its stimuli to their own pre-conceived notions regarding the world (Gerber and Green 1999). This is usually considered from a human standpoint, where people match what they see to what they've experienced in the past. After people have formed a conclusion regarding an issue, they are less likely to process future information in an objective fashion. This is common in partisan politics, as evidenced in related work (Jerit and Barabas 2012). However, these perceptual biases can go beyond humans and can appear in machine learning applications. In this chapter we will demonstrate one example of how a perceptual bias can diminish the performance of machine learning models. We will show that how a prior assumption about the meaning of a specific aspect of the content of social media data, the emojis used by the users of a social media site, can cause a reduction in the accuracy of a common text analysis task, sentiment analysis.

Social media and web communication are a major part of every day life for most people. Sites like Facebook, Twitter, and WhatsApp all have hundreds of millions to billions of users who communicate on these platforms each and every day. While images and videos have become commonplace on these sites, text is still the predominant method of communication. This happens on our smartphones, tablets, and computers billions of times every day.

While communication through text has long been the norm, it still has many key issues that keep it from having the depth of face-to-face conversation. One of these



issues is the lack of emotional cues (Daft and Lengel 1986). When conversation is carried out through text, the lack of non-verbal cues removes key emotional elements from the conversation (Byron and Baldrige 2005). One solution to this problem is *emoticons*, which are combinations of standard keyboard characters to create facial representations of human emotion, e.g., :), :(, ^\_^, and :D. While widely used, there are only a limited number of character combinations that make a cogent representation of a human emotion, and the exact meaning of many emoticons can be ambiguous (Walther and D’Addario 2001). To provide for richer expression, “emojis” can convey a richer set of non-verbal cues.

*Emojis* are a set of reserved characters that, when rendered, are small pictograms that depict a facial expression, or other object. Unlike emoticons, these are not combinations of characters devised by the users, but instead single characters that are rendered as small pictures on the screen. There are currently over 1,800 different emojis defined by the Unicode specification, a number that grows with each iteration of the specification.<sup>39</sup> These emojis are either facial expressions (e.g., “grinning face,” 😊, character code U+1F600),<sup>40</sup> or ideograms (e.g., “birthday cake,” 🍰, character code U+1F382).

While emojis have allowed for increased expression of emotion through text, they have an inconsistency. That is, emojis are rendered differently on each platform. Different fonts display the same character according to a different style; similarly, each major platform has its own font to display these characters. With regular characters this is not crucial as each character has a predefined meaning. Emojis do not enjoy

---

<sup>39</sup><http://www.unicode.org/Public/emoji/3.0/emoji-data.txt>

<sup>40</sup>Unless otherwise noted, when an emoji appears inline, the depiction comes from Emoji One (<http://emojione.com/>), which is considered the standard.

this predefined definition, and these changes in rendering can have an impact on the way that the emoji is interpreted. Emojis are often small depictions of human faces, so slight variations can make the face look entirely different. This can cause a different interpretation of the text than was initially intended by the author for emotional interpretation to the text. This issue was raised in Miller et al. 2016, where human workers on a crowdsourcing platform rated the sentiment of emojis. The results of these ratings indicate that the same emoji can be perceived as positive on some platforms, while it can be perceived as negative on others. Miller’s finding is important, with repercussions for the 2 billion people who use a smartphone.<sup>41</sup> Furthermore, recent research suggests that emojis are replacing emoticons on social media sites such as Twitter (Pavalanathan and Eisenstein 2015). With so many people affected by this possibility for miscommunication, it is important that we study the implications and possible solutions to this problem.

In this chapter, we answer the following research questions:

**RQ1** *Does misinterpretation based upon emoji rendering occur in real world data?*

Miller et al. 2016 discovered this possibility for misinterpretation using surveys.

We assess if these phenomena appear in real world social media datasets.

**RQ2** *What is the scale of this misinterpretation?* If this misinterpretation manifests,

that does not necessarily mean that it affects a vast array of communication.

We measure the extent to which communication on one social network, Twitter, is affected by misinterpretable emojis.

**RQ3** *How can the problem of cross-platform emoji interpretation be addressed?* Using

our insights from the first two analytical portions, we construct a solution that produces a mapping of emojis from one platform to those on another.

---

<sup>41</sup><https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>

Table 19: Amount of Data Collected by Platform. The “Source(s)” Column Indicates the Applications We Chose to Represent Each Platform.

Platform	Source(s)	Tweets
Android	Twitter for Android	5,839,392
iOS	Twitter for iPad Twitter for iPhone iOS	12,850,344
Twitter	Twitter Web Client	1,562,655
Windows	Twitter for Windows Phone Twitter for Windows	114,175
<i>Total</i>		20,366,566

**RQ4** *Does correcting for emoji misinterpretation have a meaningful effect on analysis?*

We measure the usefulness of our mapping by applying it to a common text analysis task: sentiment analysis. We show that the performance is increased by mapping all tweets to a common emoji language.

## 6.1 Analyzing Platform-Specific Emoji Usage

Previous work by Miller et al. 2016, 2017 identified platform-specific emoji meaning by carrying out surveys with human participants on Amazon’s Mechanical Turk. While these insights are extremely useful, we must verify that these patterns truly occur in real-world data. In this section we outline our process for collecting an emoji dataset and measure the effect to which platform plays a role in the use of emojis.

### 6.1.1 Collecting a Platform-Specific Emoji Dataset

The dataset used in this work consists of social media posts collected from Twitter. Twitter is an attractive option for our analysis for several reasons. First, it is large. With approximately 500 million tweets each day, it is one of the largest social media sites. Also, because of its 140-character limit, users may be prone to use emojis because they can help the user to be more expressive within the restrictive character limit. Furthermore, Twitter, like many other social networking sites, is a place where people post using many different platforms. Additionally, the site makes the source of the post available as part of its metadata.

To collect the dataset used in this work, we manually identify a subset of emojis that have human faces or other emotional signals. This is to avoid emojis which lack a natural expression of emotion such as “helicopter” (U+1F681), and “mantelpiece clock” (U+1F570). The full list of codes used in the data collection will be released upon request. Using this list of emojis we query Twitter’s Filter API,<sup>42</sup> which takes as input a list of keywords to track, using our list of emojis. We tracked this data for 28 days, collecting a total of 20 million tweets.

Because the nature of this work is focused on the platform-specific nature of the emojis, we separate the tweets based on the platform from which they were posted. Twitter is an open platform, meaning that it has a fully documented and available API which any third party can use to make software to post to Twitter. While Twitter does not explicitly mention which platform the user used to write the tweet, they do provide details about the software used to author the tweet in their “source” field. This source is made available in the data that comes from their APIs. In some

---

<sup>42</sup><https://dev.twitter.com/streaming/reference/post/statuses/filter>

Table 20: Average Jaccard Coefficient of Emojis across Platforms. Random Indicates a Random Sample across All Tweets of the Size of the iOS Corpus.

	Android	iOS	Twitter	Windows	Random
Android	1.000	0.153	0.111	0.062	0.010
iOS	0.153	1.000	0.086	0.052	0.018
Twitter	0.111	0.086	1.000	0.047	0.005
Windows	0.062	0.052	0.047	1.000	0.002
Random	0.010	0.018	0.005	0.002	1.000

cases, the “source” is a clear indication of the platform because some software is only available on one platform. We use these when determining the platform from which the tweet was posted. We identify four platforms with distinct emoji sets according to Emojipedia:<sup>43</sup> iOS, Android, Windows, and Twitter. We select these because they are major platforms. There are two reasons behind this. First, the results obtained from these platforms will apply to more users on the social media site. Second, by choosing large platforms we can accrue a more sizable dataset, which will yield a more stable mapping. Statistics of the dataset, as well as the applications we selected to represent each platform, are shown in Table 19.

Now that we have collected an emoji dataset, we will continue to investigate the differences between the usage of emojis on different platforms. Towards answering **RQ1**, this analysis is performed from two perspectives: 1) the positioning of the emojis within a word embedding, and 2) the sentiment of the posts in which the emojis appear.

---

<sup>43</sup><http://emojipedia.org/>

### 6.1.2 Measuring Emoji Embedding Agreement

First, we investigate how consistent the embeddings of the emojis are across platforms. Word2Vec (Mikolov et al. 2013), a word embedding algorithm, learns a vector for each word in the vocabulary. For a given word, the vector is constructed based upon the neighboring words each time the given word is used. Thus, we employ this technique to measure how consistent the usage of emojis is across platforms.

First, we build a base word embedding by training a SkipGram Word2Vec model using the entire dataset with emojis removed.<sup>44</sup> Then we use this base embedding to train a platform-specific embedding by adding the emojis back in and updating the model with the new data. It is important to note that we do not update non-emoji words, we only update the vectors of the newly-added emojis. After following this process, we have 4 platform-specific emoji embeddings. To give a sense of how these deviate from general Twitter conversation, we also create a platform-agnostic word embedding by training a random sample of tweets of the size of the iOS platform.

Word embeddings have the property that words that are more semantically similar will be embedded closer together (Mikolov et al. 2013), where “closeness” is defined by cosine similarity. We compare the usage of the emojis on each platform by seeing the words that are embedded closest to each emoji. For each platform, we extract the closest 1,000 words to the emoji. To compare the differences across platforms, we compute the Jaccard coefficient between the top 1,000 on the first platform and the top 1,000 on the second. We compute the average Jaccard coefficient across all emojis.

The results of this experiment are shown in Table 20. The results indicate that the emojis are embedded next to very different words across models. The most

---

<sup>44</sup>This process is explained in greater detail in Section 6.2.1.

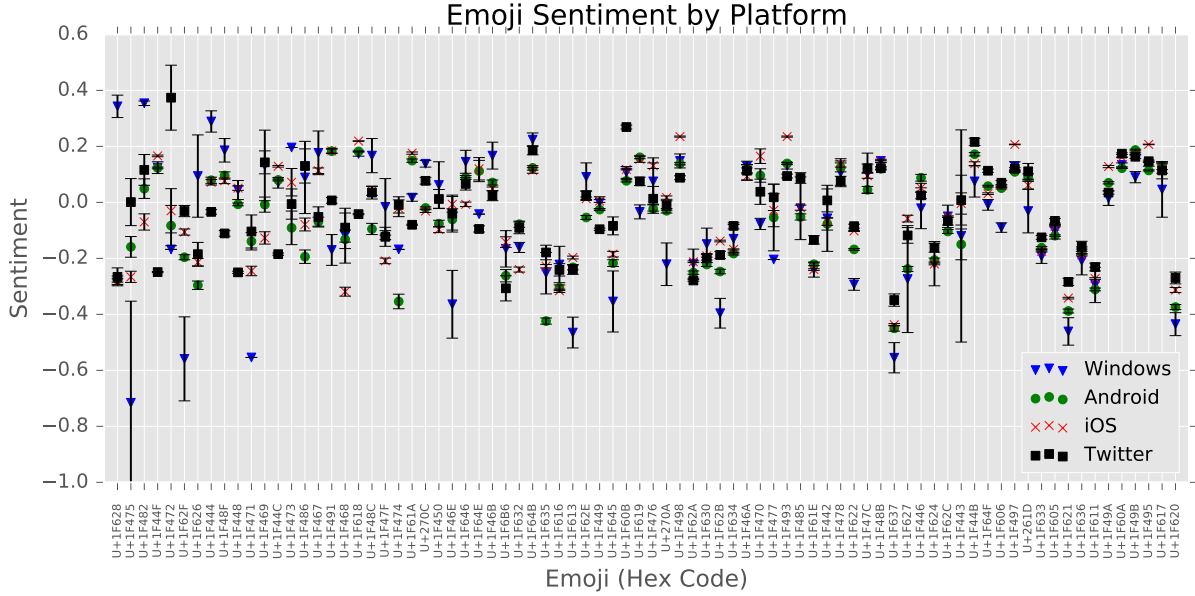


Figure 28: Average Sentiment for Each Platform by Emoji. Error Bars Indicate the Variance Calculated from the Bootstrapped Samples. A Sentiment Score of 0.0 Is “Neutral,” 1.0 Is “Perfectly Positive,” and -1.0 Is “Perfectly Negative.” the X-Axis Labels Indicate the Unicode Character Code of the Emoji.

agreeing platforms are iOS and Android, where an average of 153 words are common across the top 1,000 in the emojis. We also note that Windows has a much lower average agreement than other models. Finally, *all* platforms are extremely different from a random sample. This means that combining tweets from all platforms, as is done in many analytical tasks, will yield a significantly different representation than considering each platform individually.

While we have discovered that the emojis are used in different contexts across platforms, that does not necessarily mean that their meaning is perceived differently. To better answer this question, we assess the sentiment of the tweets in which the emojis occur.

### 6.1.3 Assessing Emoji Sentiment

Having collected an emoji dataset, we continue to see if the usage of the emojis is different across platforms. To measure the consistency of the meaning of the emoji, we perform sentiment analysis on the tweets containing the emoji. Using the Pattern library’s sentiment analysis tool,<sup>45</sup> we compute the average sentiment for each emoji on each platform. This is done by removing the emoji from the tweet and using the sentiment analysis tool to compute the sentiment score for the remaining text. Finally, we consider the possibility that each platform may have a different sentiment “bias,” that is the sentiment expressed on those platforms is different. For example, Windows Phone may be preferred by business users who are less likely to express negative sentiment in their posts. To account for this, we take the average sentiment across all tweets on the platform, and subtract that from the emoji’s score.<sup>46</sup> We then take the average sentiment across all tweets in which the emoji occurs and plot the average in Figure 28. Because we only have one corpus for each platform, we bootstrapped (Efron and Tibshirani 1994) the corpus to obtain confidence intervals. By sampling with replacement, we create 100 bootstrapped samples and reproduce the process above to understand the variance in the data. This is how we obtain the confidence intervals displayed in Figure 28.

The results of this experiment show several phenomena about the usage of emojis online. First, we see that in many of the emojis that there is a significant difference between the sentiment in at least two of the platforms. Among those, some even have









---

<sup>45</sup><http://www.clips.ua.ac.be/pages/pattern>

<sup>46</sup>All platforms have roughly the same sentiment bias of +0.20 with the exception of Windows Phone with a sentiment bias of +0.30.



Table 21: Example Emojis Provided to Illustrate the Difference in Meaning. The Names and Codes Are Official, Provided by the Emoji Unicode Standard.

Name	Code	Android	iOS	Twitter	Windows
Fearful Face	U+1F628				
Clapping Hands	U+1F44F				

*diverging* sentiment polarities. In these cases, if one were to read only the tweets from a particular platform, they would think that emoji has a completely different meaning than it does on another platform. To illustrate this point, we provide an example of the emoji differences. We include their official definition according to the Unicode standard,<sup>47</sup> and their depiction across the different platforms in Table 21. Take, for example, the “fearful face” emoji. In the case of this emoji, Figure 28 indicates that Android, iOS, and Twitter all have this emoji hovering at the roughly “neutral” area of the sentiment spectrum. Windows, however, is an extreme outlier with most users including this emoji in positive tweets. The intuition behind this phenomenon is demonstrated clearly by the “Fearful Face” emoji shown in Table 21, where Android, iOS, and Twitter display the emoji completely differently from Windows. We speculate that the difference in the way the emojis forehead is rendered could cause this difference in interpretation. Another difference that appears in the “Clapping Hands” emoji, this time with Twitter being the outlier. Android, iOS, and Windows all clearly show two hands with action lines indicating that they are moving together. Twitter, on the other hand, is less perceptible. Only one hand is clearly visible, and this could give the impression of a “slap” motion, yielding the more negative sentiment.

We have now conducted two experiments on the cross-platform use of emojis. These experiments both illustrate that the use of emojis is platform specific, answering

<sup>47</sup><http://unicode.org/emoji/charts/full-emoji-list.html>

**RQ1.** In the case of our word embedding experiment, we find that the words that neighbor a certain emoji are vastly different between platforms, indicating that they are used in different contexts. This point is furthered by our sentiment analysis experiment, where we found that some emojis have *significantly* different sentiment scores across platforms, confirming the results of Miller et al. 2016 on a large-scale, real-world dataset.

#### 6.1.4 The Scale of Misinterpretation

We continue to address **RQ2**, which involves measuring the scale at which emojis can contribute to miscommunication across platforms. We have used real-world data to show that this problem exists, however, we do not know the extent to which users are affected by this phenomenon. Is this a wide-reaching problem impinging most Twitter users or is it an esoteric issue restricted to the few users who happen to include emojis in their text?

Based on the results of our experiment, we find that 38.2% of all emojis yield a statistically significant sentiment difference between different platforms. While this is a minority, these emojis appear in 73.4% of all of all the tweets in our dataset. Since our dataset was collected using emojis, we need to leverage outside information to estimate the impact for all of Twitter. To estimate the fraction of tweets using these emojis, we use the Sample API,<sup>48</sup> which provides a 1% sample of *all* of the tweets on Twitter, irrespective of whether they contain an emoji. We collected data from the Sample API during the same time period we collected the emoji dataset. Through analyzing this data, we observe that of the 94,233,024 tweets we collected

---

<sup>48</sup><https://dev.twitter.com/streaming/reference/get/statuses/sample>

from the Sample API during this time period, 8,129,483 tweets (**8.627%**) use emojis that are prone to misinterpretation. In other words, 1 in every 11 tweets sent on Twitter contain an emoji that has a statistically significant interpretation to a user on a different platform.

These findings indicate that there is a need to disambiguate emojis across platforms. To this end, the rest of this chapter proposes a strategy for generating a mapping which can disambiguate emoji choice across platforms. Next, we evaluate this mapping to show that it can increase the performance of sentiment analysis tasks.

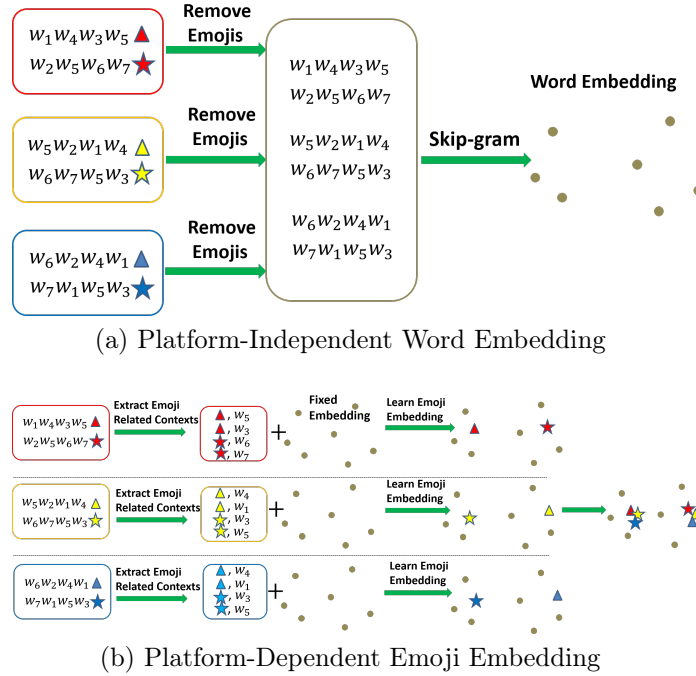


Figure 29: An Illustration of Platform-Dependent Emoji Mapping Construction. The Three Rectangles Blue, Green, Red Means Corpus of Three Platforms. Each Line Is a Sentence. Triangle and Star Denote Emojis. Red Triangle Means an Emoji in Red Platform While Yellow Triangle Means the Same Emoji in Yellow Platform.

## 6.2 A Cross-Platform Emoji Mapping Solution

The fact that the representations of emojis are different both from the perspective of the embedding as well as the sentiment suggests that a mapping may help us disambiguate emoji meaning across platforms. But how can we construct such a mapping? In this section, we describe our approach to constructing a platform-dependent emoji mapping. This is done towards the goal of answering **RQ3**. The goal of this mapping is to provide a translation from an emoji on one platform to its corresponding emoji on another. The purpose of the platform-dependent emoji mapping is to disambiguate platform-specific emoji interpretation. To construct the cross-platform emoji mapping, we need to understand the semantic meaning and the sentiment polarity of the emojis on each platform. We then can construct the platform-dependent mapping by identifying emojis that have the closest semantic and sentiment polarities across platforms. Thus, the key step is to learn the representation of emojis from the texts on each platform such that the representations capture the platform-dependent semantic meanings of emojis and also allow simple similarity matching to construct the mapping.

Recent advances in natural language processing suggest that word embeddings such as Skip-gram (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014), which represent each word as a low-dimensional distributional vector, are able to capture the semantic meanings of words. The low-dimensional vector representations also allow a similarity calculation using cosine or Eulidean distance, which eases the mapping construction. In addition, recent findings on emoji analysis (Barbieri, Ronzano, and Saggion 2016; Eisner et al. 2016) demonstrate that by treating each emoji as a word and performing Skip-gram on texts, the vector representation learned

by Skip-gram can capture semantic meanings and sentiment polarities of emojis, which improve the performance of sentiment classification. Therefore, word embedding is a natural choice to learn platform-dependent semantic meanings of words and emojis for mapping construction.

While word embeddings may be suitable for this task, there are two challenges which we must overcome in order to create the mapping. The first challenge is that we must find a way to map emojis from a source platform to their true equivalent semantic emoji on the target platform. The second challenge is that when we build our platform-specific embedding, the position of the *words* will change, as well as the emojis. Thus, we need to figure out a way to represent the words within a common space first, so that we can extend it to measure the emojis relative to the words. Our solution addresses both of these challenges.

In the subsequent sections we will introduce in detail how we leverage a word embedding approach to build a platform-dependent emoji embedding and how to use the embedding to construct the mapping.

### 6.2.1 Building the Embedding

For simplicity of explanation, let  $\mathcal{T}_p$ ,  $p = 1, \dots, P$  be the set of tweets from each platform. The process of building word and emoji embeddings are illustrated in Figure 29. As shown in Figure 29a, since the interpretation of words are platform-dependent, for each corpus  $\mathcal{T}_p$ , we first remove the emojis, which are denoted as  $\tilde{\mathcal{T}}_p$ . We then combine  $\tilde{\mathcal{T}}_p$ ,  $p = 1, \dots, P$ , as one large corpus  $\tilde{\mathcal{T}}$ . Then Skip-gram is applied on  $\tilde{\mathcal{T}}$  to learn word embedding  $\mathbf{W} \in \mathbb{R}^{K \times N}$ , where  $K$  is dimension of the vector representation and  $N$  is the size of the vocabulary without emojis. The advantages of combining  $\tilde{\mathcal{T}}_p$ ,

$p = 1, \dots, P$ , as one large corpus  $\tilde{\mathcal{T}}$  are two-fold: (i) we obtain a large corpus which allows us to train a better word embedding; and (ii) word embedding in each platform is the same, which satisfies the assumption that word interpretations are the same for each platform. After the platform-independent word embedding is learned, then we can learn the platform-dependent embeddings for emojis. The process is depicted in Figure 29b. For each corpus  $\mathcal{T}_p$ , as the process of Skip-gram, we first use a context window of size 5 to extract the neighboring words of emojis in the corpus. Let  $(e_i, w_j)$  denote a pair of neighboring words, where  $e_i$  means emoji  $i$  and  $w_j$  means word  $j$ . The extracted pairs are then put into the set  $\mathcal{P}_p$ . We then learn the representation of emoji  $e_i$  in corpus  $\mathcal{T}_p$  by optimizing the following problem:

$$\min_{\mathbf{e}_i^p} \sum_{w_j: (e_i, w_j) \in \mathcal{P}_p} \left( \log \sigma(\mathbf{w}_j^T \mathbf{e}_i^p) + \sum_{k=1}^N \log \sigma(-\mathbf{w}_k^T \mathbf{e}_i^p) \right) \quad (6.1)$$

where  $\mathbf{e}_i^p$  is the vector representation of  $e_i$  in corpus  $\mathcal{T}_p$ . Equation 6.1 is essentially the negative sampling form of the objective function of the Skip-gram approach, where we try to learn the emoji representation which is able to predict the neighboring words. Note that the difference with Skip-gram is that word embeddings  $\mathbf{W}$  is fixed across different corpus, we only learn  $\mathbf{e}_i^p$ . We do the same thing for each corpus, which gives us  $\mathbf{e}_i^p$ ,  $p = 1, \dots, P$ , i.e., the vector representations of the same emoji in different platforms.

### 6.2.2 Constructing the Mapping

In this section, we detail the emoji mapping construction process. To construct the mapping between emojis across different platforms, we consider it in a pair-wise scenario. We treat one platform as the source platform and the other as the target platform. Without loss of generality, let  $\mathcal{E} = \{e_i, i \in \{1, \dots, m\}\}$  be the set of emojis

that occur in both platforms. By learning the emoji embedding representations in each platform, we are able to capture the platform-dependent semantic features for the emojis. Thus, given an emoji in the source platform, we can leverage the emoji embedding representation to connect the semantic space between the source and target platforms, and then find the most similar emoji in the target platform.

Specifically, based on all the emoji embeddings from the source platform  $\{\mathbf{e}_i^s, i \in \{1, \dots, m\}\}$  and target platform  $\{\mathbf{e}_j^t, j \in \{1, \dots, m\}\}$ , we want to map the most similar emoji in target platform for each emoji in the source platform. To compute the similarity of two emoji embeddings, we adopt the cosine similarity measure as follows,

$$\text{sim}(\mathbf{e}_i^s, \mathbf{e}_j^t) = \frac{\mathbf{e}_i^s \cdot \mathbf{e}_j^t}{\|\mathbf{e}_i^s\| \cdot \|\mathbf{e}_j^t\|} \quad (6.2)$$

Given an emoji  $e_i$  in the source platform, we first compute the similarity between  $e_i$  with all emojis in target platform. Then we select the emoji which gives the maximum similarity score. We solve the following objective function to obtain the mapping emoji  $\hat{e}_j$  in target platform for  $e_i$ ,

$$\hat{e}_j = \arg \max_{e_j \in \mathcal{E}} \text{sim}(\mathbf{e}_i^s, \mathbf{e}_j^t) \quad (6.3)$$

we then get a mapping pair  $(e_i, \hat{e}_j)$ , where the first emoji is from the source platform and the second one is from the target platform. Note that the emoji mappings are directional, which means if  $(e_i, \hat{e}_j)$  is a mapping pair,  $(\hat{e}_j, e_i)$  is not necessarily a mapping pair.

### 6.3 Evaluating the Mapping

Now that we have presented the methodology for constructing the emoji mapping, we will validate its utility, answering **RQ4**. We measure the utility of our mapping by

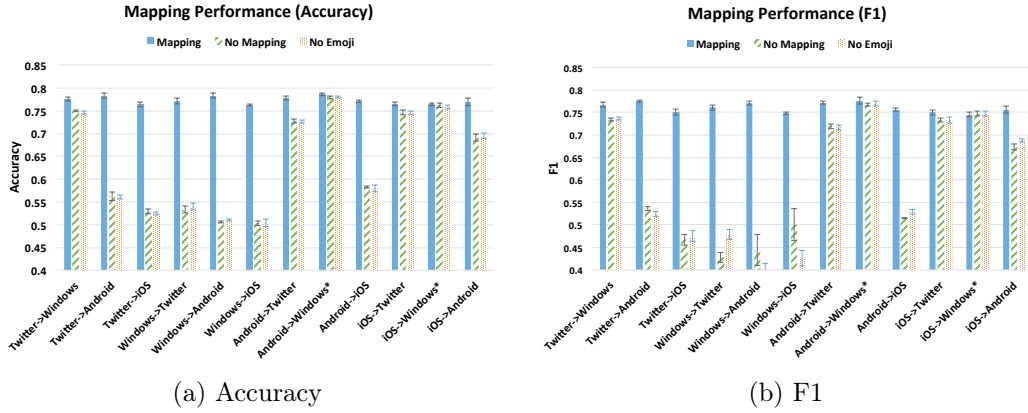


Figure 30: Performance Results across All “Source” X “Target” Pairs. Asterisks Indicate (Source, Target) Pairs Where the Mapping Is Not Significantly Better Than the Two Baselines.  $F_1$  Is Computed with Respect to the “Positive” Class. Tweets with Sentiment Scores in the Range of  $(-0.2, 0.2)$  were Deleted.

seeing how well it can help in a common text analysis task: sentiment analysis. We design experiments to show that the sentiment analysis task is improved by applying our emoji mapping to the data to bolster the consistency of analogy meaning in the dataset. This simultaneously shows the efficacy of our approach as well as the extent to which emoji ambiguity plays in standard text analysis tasks.

### 6.3.1 Predictive Evaluation

We have shown that emoji usage is different across platforms. This means that the same emoji can have different sentiment meanings across platforms. From a machine learning perspective, this means that platform-specific emoji renderings introduce *noise* into the dataset, ultimately lowering the classification performance. We hypothesize that by applying our mapping we are in effect converting the dataset into a single language emoji dataset, and the consistent emoji language will help in assigning a sentiment label.



### 6.3.2 Experimental Setup

In each experiment, we evaluate our mapping using a pair of platforms. One of the platforms is the “source”, and another is the “target.” The experiment consists of two phases. In the first phase, we mix the data from both platforms together, perform sentiment analysis, and measure the accuracy using 5-fold cross validation. We call the average accuracy across all 5 folds “A1.” In phase two, we apply the mapping to the emojis on the “target” platform, and then repeat the process in phase one. We call the average accuracy from this experiment “A2.” We measure the effectiveness of the mapping as  $A2 - A1$ .

In preprocessing, we remove stopwords, and strip the case from all words. We tokenize the dataset using the “TweetTokenizer” module in Python’s NLTK (Bird, Klein, and Loper 2009). Following the labeling approach outlined previously, we use the VADER library to assign a sentiment score to each tweet as ground truth. The sentiment score provided by this library is a continuous value from  $[-1.0, 1.0]$ . We convert this problem to a binary classification task. We delete all tweets in the range  $(-0.2, 0.2)$  to ignore ambiguous cases. We then assign the sign of the label from the Pattern library as the label of the tweet. When training the word embeddings as well as the emoji embeddings, we use  $K = 20$  as the number of dimensions. To prevent information leakage between the training and test sets, the emoji mapping used in these experiments is built using data collected from September 23rd, 2016 - October 4th, 2016. The training and test instances used in our cross validation experiments are taken from October 5th, 2016 - October 20th, 2016.

Each tweet is represented by the average of the word vectors for all of the words  $\mathcal{W}'$  and emojis  $\mathcal{E}'$  in the tweet. We always use the target embedding for emojis. Thus,

a tweet is represented as follows:

$$repr = \frac{1}{|\mathcal{W}'|} \sum_{w_k \in \mathcal{W}'} \mathbf{w}_k + \frac{1}{|\mathcal{E}'|} \sum_{e_i \in \mathcal{E}'} \mathbf{e}_i^t, \quad (6.4)$$

where  $\mathbf{w}$  and  $\mathbf{e}^t$  are word embedding and target platform emoji embedding. When the mapping is applied, the representation is as follows:

$$repr\_map = \frac{1}{|\mathcal{W}'|} \sum_{w_k \in \mathcal{W}'} \mathbf{w}_k + \frac{1}{|\mathcal{E}'|} \sum_{e_j \in \mathcal{E}'} \hat{\mathbf{e}}_j^t, \quad (6.5)$$

where  $\hat{e}_j$  is the mapping of the target emoji on the source platform.

Having extracted the data, the labels, and formalized the representation, we use SVM to build a classifier using 5-fold cross validation. The only difference between each set up is how each tweet is represented. We compare three tweet representations:

1. **Mapping.** This is the representation where the emojis in the target platform are mapped to the emojis in the source platform. This corresponds to the representation obtained by Equation 6.5.
2. **No Mapping.** This is the representation when no mapping is applied. This is obtained by the representation in Equation 6.4.
3. **No Emojis.** Our hypothesis is that the incorrect emojis are adding noise to our dataset, which hinders classification. Aside from our proposed solution, another way to de-noise the dataset is to simply remove the emojis. We do this by using the following representation formula:

$$repr\_noemoji = \frac{1}{|\mathcal{W}'|} \sum_{w_k \in \mathcal{W}'} \mathbf{w}_k. \quad (6.6)$$

Emojis are simply not considered in the resulting feature representation.

### 6.3.3 Experimental Results

We report the average across the 5 folds in Figure 30. The results overall are encouraging. In most of the source/target pairs we obtain significantly better results than both doing nothing, and removing the emojis. Further, we note that removing the emojis beats doing nothing in almost all of the cases, further validating our noise assumption. The only two pairs where the mapping does not obtain significantly better results are iOS  $\rightarrow$  Windows (“source”  $\rightarrow$  “target”), and Android  $\rightarrow$  Windows. The lower results across these two cases could be a side effect of the nature of the Windows emojis. The results of our previous analysis indicate that the Windows emoji set is significantly different from the rest in many cases, and that could prevent us from learning a quality mapping.

The results in Figure 30 indicate that we can obtain a significant (sometimes as much as 25%) gain in sentiment classification accuracy for some pairs. It is important to note that these experiments were carried out using *only* tweets that contained emojis, which is due to the way the dataset was collected. Therefore, an improvement of this magnitude can only be achieved on a similar dataset.

Using the random 1% sample of Twitter data introduced in Section 6.1.4, we discover that 15.0% of all tweets contain at least one of any emoji. Our analysis from the same section indicates that 8.627% of all tweets, and thus 57.5% of tweets containing an emoji are affected by this phenomenon. This justifies the huge improvements seen in Figure 30. We speculate that another factor that contributes to the superior performance is the large amount of data upon which the embeddings were trained.

### 6.3.4 Results by Sentiment Threshold

In the previous experiment we removed tweets that had a sentiment score between  $(-0.2, 0.2)$ . The motivation behind this step is that tweets within this threshold were so ambiguous that they could not be meaningfully assessed for sentiment. In this section, we vary this parameter to see how robust our method is to ambiguous tweets. We vary the sentiment threshold from 0.1 (leaving many ambiguous tweets) to 0.9 (leaving only the most sentiment-expressive tweets).

Instead of showing the results of every possible combination, which is impossible due to space limitations, we instead test whether the two sets of results are significantly different from each other. The T-test tests the null hypothesis that the means of the two distributions are equal. The results can be seen in Table 22 for the comparison between Mapping and No Mapping, and in Table 23 for the comparison between Mapping and No Emoji. In the first case we can easily reject this null hypothesis at the  $\alpha = 0.05$  significance level in all cases except for  $\text{iOS} \rightarrow \text{Windows}$ . Despite passing the significance bar, this pair still yields the least significant result in Table 23. This is consistent with our previous result, where this particular mapping did not fare significantly better, and further supports our suspicion that Windows is an outlier due to the way it renders emojis.

## 6.4 Discussion

In this work, we set out to answer a series of questions regarding the nature and extent of cross-platform emoji misinterpretation, and to provide a solution that can

Table 22: Sentiment Threshold Significance T-Test between “Mapping” and “No Mapping” Experimental Designs. The Only Insignificant Mapping Is “iOS → Windows.”

	Accuracy	F1
Twitter→Windows	3.538e-09	1.200e-05
Twitter→Android	2.493e-08	8.754e-07
Twitter→iOS	8.071e-08	4.887e-07
Windows→Twitter	4.418e-07	5.687e-05
Windows→Android	3.538e-09	8.440e-07
Windows→iOS	2.115e-08	1.960e-05
Android→Twitter	1.234e-07	1.193e-05
Android→Windows	2.586e-07	3.037e-03
Android→iOS	2.081e-04	6.212e-06
iOS→Twitter	2.726e-06	1.313e-05
iOS→Windows	0.294	0.094
iOS→Android	8.877e-08	3.407e-06

help researchers and practitioners to overcome this platform-specific inconsistency in their analysis. In the introduction, we outlined four research questions based around these issues. Here, we summarize our findings as they pertain to each question and end with a discussion of areas of future work.

***RQ1*** *Does misinterpretation based upon emoji rendering occur in real world data?*

To perform this study we use Twitter, a large, open platform where users post hundreds of millions of tweets per day, and where the data made available by Twitter contains the source from which the tweet was posted, which can be used to identify the underlying platform. We find that the sentiment of the tweets in which emojis occur differs widely and significantly across platforms. In multiple cases, the same emoji exhibited opposing sentiment polarities on different platforms. This finding goes beyond surveys done in Miller et al. 2016 to find these patterns in real-world data.

***RQ2*** *What is the scale of this misinterpretation?* By analyzing a random sample of tweets, we obtain that 8.627%, or roughly 1 in every 11 of all of the tweets contain

Table 23: Sentiment Threshold Significance T-Test between “Mapping” and “No Emoji” Experimental Designs. Significance T-Test Results for Sentiment Classification with Respect to Sentiment Threshold. All the Results Are Significant with  $\alpha = 0.05$ .

	Accuracy	F1
Twitter→Windows	2.466e-06	5.142e-08
Twitter→Android	8.818e-09	1.923e-09
Twitter→iOS	6.121e-08	2.400e-08
Windows→Twitter	6.047e-08	1.665e-07
Windows→Android	3.293e-08	3.141e-07
Windows→iOS	9.665e-08	1.417e-06
Android→Twitter	3.525e-06	2.021e-06
Android→Windows	1.744e-03	1.124e-04
Android→iOS	7.423e-08	8.920e-09
iOS→Twitter	2.078e-05	2.954e-06
iOS→Windows	3.701e-04	3.399e-02
iOS→Android	3.303e-08	4.338e-08

an emoji that is used in a statistically significantly different fashion on a different platform.

**RQ3** *How can the problem of cross-platform emoji interpretation be addressed?*

With these findings in mind, we endeavor to construct an emoji mapping to help researchers, practitioners, and even readers of social media data to better understand the intended message of the sender given their platform-specific emoji mapping. By leveraging word embeddings, we are able to exploit the property that “similar words are embedded closer together” in order to find the corresponding emoji across platforms and build an (emoji, platform)  $\rightarrow$  (emoji, platform) mapping.

**RQ4** *Does correcting for emoji misinterpretation have a meaningful effect on analysis?* We evaluate the effectiveness of our embedding by applying it to sentiment analysis. We chose sentiment analysis as it is a prediction problem very common to social media data. We show that by mapping all tweets to a consistent emoji

vocabulary, we can significantly increase the performance of sentiment analysis by diminishing the amount of emoji noise in the dataset.

This work opens up the doors for many areas of research. While we have largely explored this platform disagreement in emoji meanings from the perspective of computational and predictive tools, it certainly is not limited to this. In fact, it may be useful to include our mapping in user interfaces in order to increase understanding between individuals communicating on different platforms. Alternatively, social media platforms can take the approach of a “closed” platform, where all platforms conform to a single emoji set. WhatsApp is an exemplar of this closed approach.

There are many possible extensions to our mapping approach. For example, our mapping assumes that each emoji will have an analogue on the target platform. That is not always the case. For instance, the clown emoji (“clown face,” 🤡, character code U+1F921) exists on Android, Twitter, and Windows Phone, but it is not available on iOS at the time of this writing. Furthermore, our mapping only consists of four platforms, a limitation. With sufficient data the mapping can be extended to represent many more. Resolving these issues are key to improving the utility of our mapping.

The mapping, code, and data used in this work is available in accordance with Twitter’s data sharing policies online.<sup>49</sup> We provide all of the performance scores used to obtain the significance results from the sentiment threshold experiment, as well as an expanded version of Figure 28, containing all of the emojis considered in the analysis.

---

<sup>49</sup><http://www.public.asu.edu/~fmorestat/emojimapping/>

## CONCLUSIONS

This chapter concludes the dissertation with a summary of my contributions and a look towards future directions of this thesis.

### 7.1 Methodological Contributions

Here we discuss the methodological contributions made in this thesis including the technical and algorithmic contributions we made throughout.

1. **Discovering Bias in Social Media APIs.** I introduce a methodology for identifying bias in social data APIs. Introduced in Chapter 3, this methodology uses statistical techniques to identify the extent to which a dataset collected from social media differs from the true population. An important part of this methodology is the way in which I create multiple samples from the population for more robust evaluation of these statistical difference. However, this methodology relies on the full population, which is not a valid assumption for researchers to have given the limiting cost. Due to this limitation, this dissertation provides other ways of validating these differences.
2. **Statistical Techniques for Assessing Bias without the Firehose.** In Chapter 4, I introduce a new methodology for identifying sample bias in the API without the need for the Firehose. This works by comparing the data with other sources of data which are readily available for researchers. This methodology



works by bootstrapping these sources and creating confidence intervals for how biased these data sources are.

3. **Algorithms for Collecting Data with less Sampling Bias.** While identifying data collection bias is important for researchers to understand the limitations of their data, it is ideal to collect a dataset that is not biased to begin with. Towards this end, I have developed several methodologies that collect data from these APIs in such a way that they contain less sampling bias. These methods rely on clustering techniques to reparameterize the queries to the API in such a way that a larger sample is gleaned from these resources. I verify that these samples are indeed a more accurate representation of the data from the Firehose.
4. **Algorithms for Detecting Malicious Users.** In this dissertation, I view these from two angles. First, from the perspective of users who intentionally manipulate the data from the APIs in order to make their messaging seem more prominent. Secondly, from the perspective of bot detection where I develop a new approach to identify more bots in a social media dataset.
  - a) *Bots manipulating the APIs.* In Chapter 6 I develop an algorithm to prove that it is possible to build bots in such a way that they will be able to disproportionately skew the content of the Sample API. We then develop a methodology that uses knowledge about the limitations that Twitter imposes on users to expose the bots that are carrying out these attacks.
  - b) *Improving bot detection approaches.* Most bot detection approaches focus on precision, or ensuring that the users that they identify as bots are correct. While precision is important from the perspective of companies to lower the chance of enraging real users, it is less important from a research perspective. When using algorithms that focus on precision, we find that

these algorithms still leave a number of bots in the dataset. Thus, it is crucial that these algorithms also focus on *recall*. In Chapter 7 we developed a new algorithm which is a tweak on AdaBoost to improve the recall of bot detection algorithms. We showed that this methodology can achieve superior performance to state-of-the-art bot detection algorithms.

5. **Identifying and Correcting Perceptual Bias in Social Media Data.** In the penultimate chapter of this thesis, I shed light on the problem of perceptual bias in social data. That is, even with a dataset collected using a representative sample and after removing the bots, there are ways in which we perceive the data that do not match reality. One way is the way in which we interpret the way users choose emojis for their tweets. Prior work has discovered that emoji use varies across platforms. I show how it can hinder a common task done on social media data: sentiment analysis. I also develop a methodology which can construct a mapping from emojis on one platform to those on another.

## 7.2 Future Directions

These findings pave the way for many areas of future work, both from the angle of theoretical issues and applications. These include:

1. **Considering other possible sources of bias.** This thesis presents several avenues for bias in social media data. However, there are countless more. For instance, there are hundreds of different cognitive biases that can manifest in social media data, as discussed in the related work chapter. These biases all have the potential to wreak havoc on analyses performed on social data. Future work

is to assess the extent to which these biases skew social data. If they do, then other areas of future work will be to develop techniques to reduce their effect.

2. **Extending the understanding shill behavior.** Our study of shills shined light on how they behave. We were able to learn important patterns of how they carry out their attacks online. Unfortunately, our study was limited in two critical ways. First, we did not have a dataset of adequate size to make meaningful claims about specific patterns, including temporal or language patterns. A larger dataset would be needed to make credible claims about their behavior pertaining to language. Second, we only studied one specific instance of shill behavior online: the use of shills by Hillary Clinton’s campaign in the 2016 election. More work will be needed to verify these patterns in other shill campaigns. Finally, we used off-the-shelf classification techniques

These areas address the sources of social data bias that we discuss in this work. To conclude the future work discussion, we echo a sentiment that was introduced in the introduction. There are many areas of social data bias that are ripe for future work. For example, there are ethical biases and behavioral biases such as those discussed in Olteanu et al. 2016. In addition to these, we point out that we have provided a preliminary framework to deal with the biases that we solve in this dissertation. This framework is intended to be employed by future work. This dissertation serves as a platform to illuminate the way for researchers to obtain more credible, reproducible results using this novel source of data, social media. This framework can both be extended to further explore the biases presented here, and it can also be applied to new types of bias.

## REFERENCES

- Achrekar, Harshavardhan, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. 2011. “Predicting Flu Trends using Twitter Data.” In *INFOCOM*, 702–707. IEEE.
- Agresti, A. 2010. *Analysis of Ordinal Categorical Data*. Vol. 656. Hoboken, New Jersey: Wiley.
- Ahmed, Nesreen K., Jennifer Neville, and Ramana Kompella. 2013. “Network Sampling: From Static to Streaming Graphs.” *ACM Trans. Knowl. Discov. Data* (New York, NY, USA) 8, no. 2 (June): 7:1–7:56. doi:10.1145/2601438.
- Allcott, Hunt, Matthew Gentzkow, et al. 2017. “Social Media and Fake News in the 2016 Election.” *Journal of Economic Perspectives* 31 (2): 211–236.
- Angwin, Julia, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. “Machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks.” *ProPublica* (May).
- Asur, Sitaram, and Bernardo A Huberman. 2010. “Predicting the Future with Social Media.” In *WI-IAT*, 1:492–499. IEEE.
- Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. 2010. “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.” In *LREC*, 10:2200–2204.
- Barbieri, Francesco, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016. “How Cosmopolitan Are Emojis?: Exploring Emojis Usage and Meaning over Different Languages with Distributional Semantics.” In *Proceedings of the 2016 ACM on Multimedia Conference*, 531–535. ACM.
- Barbieri, Francesco, Francesco Ronzano, and Horacio Saggion. 2016. “What does this Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis.” In *Language Resources and Evaluation conference, LREC, Portoroz, Slovenia*.
- Baron, Jonathan. 2000. *Thinking and Deciding*. Cambridge University Press.
- Bengio, Yoshua, Holger Schwenk, Jean-Sebastien Senecal, Frederic Morin, and Jean-Luc Gauvain. 2006. “Neural Probabilistic Language Models.” In *Innovations in Machine Learning*, 137–186. Springer.

- Berk, Richard. 2011a. “Asymmetric loss functions for forecasting in criminal justice settings.” *Journal of Quantitative Criminology* 27 (1): 107–123.
- . 2011b. “Balancing the costs of forecasting errors in parole decisions.” *Albany Law Review* 74 (3): 1071–1086.
- Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O’Reilly Media, Inc.
- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Springer.
- Blei, D M, A Y Ng, and M I Jordan. 2003. “Latent dirichlet allocation.” *The Journal of Machine Learning Research* 3:993–1022.
- Boldi, Paolo, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. 2004. “Ubi-crawler: A scalable fully distributed web crawler.” *Software: Practice and Experience* 34 (8): 711–726.
- Bolukbasi, Tolga, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. “Man is to computer programmer as woman is to homemaker? debiasing word embeddings.” In *Neural Information Processing Systems*, 4349–4357.
- Borgatti, Stephen P, Kathleen M Carley, and David Krackhardt. 2006. “On the robustness of centrality measures under conditions of imperfect data.” *Social Networks* 28 (2): 124–136. doi:10.1016/j.socnet.2005.05.001.
- Boshmaf, Yazan, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. 2011. “The socialbot network: when bots socialize for fame and money.” In *Annual Computer Security Applications Conference*, 93–102. ACM.
- . 2013. “Design and analysis of a social botnet.” *Computer Networks* 57 (2): 556–578.
- Byron, Kristin, and David C Baldridge. 2005. “Toward a Model of Nonverbal Cues and Emotion in Email.” In *Academy of Management Proceedings*, 2005:1–26. 1. Academy of Management.
- Campbell, Denis G. 2011. *Egypt Unshackled: Using Social Media to @#:) the System*. Amherst, NY: Cambria Books.
- Cappallo, Spencer, Thomas Mensink, and Cees GM Snoek. 2015. “Image2emoji: Zero-shot emoji prediction for visual media.” In *ACM International Conference on Multimedia*, 1311–1314.

- Carley, K M, J Pfeffer, J Reminga, J Storrick, and D Columbus. 2012. *ORA User's Guide 2012*. Technical Report CMU-ISR-12-105. Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, Institute for Software Research.
- Casella, George, and Roger L Berger. 2001. *Statistical Inference*. Belmont, CA: Duxbury Press.
- Castillo, Carlos, Mohammed El-Haddad, Jurgen Pfeffer, and Matt Stempeck. 2014. "Characterizing the life cycle of online news stories using social media reactions." In *Computer Supported Cooperative Work*, 211–223.
- Ceron, Andrea, Luigi Curini, Stefano M Iacus, and Giuseppe Porro. 2014. "Every Tweet Counts? How Sentiment Analysis of Social Media can Improve our Knowledge of Citizens' Political Preferences with an Application to Italy and France." *New Media & Society* 16 (2): 340–358.
- Chae, Junghoon, Dennis Thom, Herald Bosch, Yun Jang, Ross Maciejewski, David S Ebert, and Thomas Ertl. 2012. "Spatiotemporal Social Media Analytics for Abnormal Event Detection and Examination using Seasonal-Trend Decomposition." In *IEEE Visual Analytics Science and Technology*. Brighton, UK.
- Cheng, Zhiyuan, James Caverlee, and Kyumin Lee. 2010. "You Are Where You Tweet: A Content-Based Approach to Geo-locating Twitter Users." In *ACM International Conference on Information and Knowledge Management*, 759–768. Toronto, Ontario, Canada. doi:10.1145/1871437.1871535.
- Cho, Charles H, Martin L Martens, Hakkyun Kim, and Michelle Rodrigue. 2011. "Astroturfing global warming: It isn't always greener on the other side of the fence." *Journal of Business Ethics* 104 (4): 571–587.
- Cho, Junghoo, and Hector Garcia-Molina. 1999. *The evolution of the web and implications for an incremental crawler*. Technical report. Stanford.
- Chu, Zi, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2010. "Who is tweeting on Twitter: human, bot, or cyborg?" In *Annual Computer Security Applications Conference*, 21–30.
- Conover, Michael D, Bruno Goncalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. 2011. "Predicting the political alignment of twitter users." In *Privacy, Security, Risk and Trust*, 192–199. IEEE.

- Cook, David M, Benjamin Waugh, Maldini Abdipanah, Omid Hashemi, and Shaquille Abdul Rahman. 2014. "Twitter Deception and Influence: Issues of Identity, Slacktivism, and Puppetry." *Journal of Information Warfare* 13 (1): 58–71.
- Costenbader, Elizabeth, and Thomas W Valente. 2003. "The stability of centrality measures when networks are sampled." *Social networks* 25 (4): 283–307.
- Cover, T M, and J A Thomas. 2006. *Elements of Information Theory*. Hoboken, New Jersey: Wiley InterScience.
- Crump, Matthew JC, John V McDonnell, and Todd M Gureckis. 2013. "Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research." *PloS ONE* 8 (3): e57410.
- Daft, Richard L, and Robert H Lengel. 1986. "Organizational information requirements, media richness and structural design." *Management Science* 32 (5): 554–571.
- De Choudhury, Munmun, Yu-Ru Lin, Hari Sundaram, K. Selcuk Candan, Lexing Xie, and Aisling Kelliher. 2010. "How Does the Data Sampling Strategy Impact the Discovery of Information Diffusion in Social Media." In *AAAI Conference on Weblogs and Social Media*, 34–41. Washington, DC, USA: AAAI.
- De Longueville, Bertrand, Robin S. Smith, and Gianluca Luraschi. 2009. "'OMG, from here, I can see the flames!': a use case of mining location based social networks to acquire spatio-temporal data on forest fires." In *International Workshop on Location Based Social Networks*, 73–80. Seattle, Washington: ACM. doi:10.1145/1629890.1629907.
- Dhillon, Paramveer, Dean P Foster, and Lyle H Ungar. 2011. "Multi-view learning of word embeddings via CCA." In *Neural Information Processing Systems*, 199–207.
- DiGrazia, Joseph, Karissa McKelvey, Johan Bollen, and Fabio Rojas. 2013. "More tweets, more votes: Social media as a quantitative indicator of political behavior." *PloS ONE* 8 (11): e79449.
- Duggan, Maeve, and Joanna Brenner. 2013. *The Demographics of Social Media Users, 2012*. Pew Research Center's Internet & American Life Project.
- Dumais, S. T., G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. 1988. "Using latent semantic analysis to improve access to textual information." In *SIGCHI Conference on Human Factors in Computing Systems*, 281–285. Washington, D.C., USA: ACM. doi:10.1145/57167.57214.

- Efron, Bradley. 1987. *The Jackknife, The Bootstrap and Other Resampling Plans*. Society for Industrial / Applied Mathematics.
- Efron, Bradley, and Robert J Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC press.
- Efron, Miles. 2010. “Hashtag retrieval in a microblogging environment.” In *Proc. of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 787–788. SIGIR ’10. Geneva, Switzerland: ACM. doi:10.1145/1835449.1835616.
- Eisner, Ben, Tim Rocktaschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016. “emoji2vec: Learning Emoji Representations from their Description.” In *Empirical Methods in Natural Language Processing*, 48–54.
- Elder, Jeff. 2013. “Inside a Twitter Robot Factory.” November 24. Accessed November 24, 2013. <http://on.wsj.com/1Qo215n>.
- Ferrara, Emilio, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. “The Rise of Social Bots.” *Communications of the ACM* 59 (7): 96–104.
- Ferrara, Emilio, Onur Varol, Filippo Menczer, and Alessandro Flammini. 2016. “Detection of Promoted Social Media Campaigns.” In *International AAAI Conference on Web and Social Media*, 563–566.
- Freeman, L C. 1979. “Centrality in Social Networks: Conceptual Clarification.” *Social Networks* 1 (3): 215–239.
- Galil, Zvi. 1986. “Efficient Algorithms for Finding Maximum Matching in Graphs.” *ACM Computing Surveys* (New York, NY, USA) 18, no. 1 (March): 23–38. doi:10.1145/6462.6502.
- Garcia-Herranz, Manuel, Esteban Moro, Manuel Cebrian, Nicholas A. Christakis, and James H. Fowler. 2014. “Using Friends as Sensors to Detect Global-Scale Contagious Outbreaks.” *PLoS ONE* 9, no. 4 (April): e92413. doi:10.1371/journal.pone.0092413.
- Gayo-Avello, D, P T Metaxas, and E Mustafaraj. 2011. “Limits of Electoral Predictions using Twitter.” In *International Conference on Weblogs and Social Media*, 21:490–493. Barcelona, Spain: AAAI.
- Gerber, Alan, and Donald Green. 1999. “Misperceptions about Perceptual Bias.” *Annual Review of Political Science* 2 (1): 189–210.



- Ghosh, Saptarshi, Muhammad Bilal Zafar, Parantapa Bhattacharya, Naveen Sharma, Niloy Ganguly, and Krishna Gummadi. 2013. “On Sampling the Wisdom of Crowds: Random vs. Expert Sampling of the Twitter Stream.” In *Conference on Information and Knowledge Management*, 1739–1744. ACM.
- Gomide, Janaina, Adriano Veloso, Wagner Meira Jr, Virgilio Almeida, Fabricio Benvenuto, Fernanda Ferraz, and Mauro Teixeira. 2011. “Dengue Surveillance Based on a Computational Model of Spatio-temporal Locality of Twitter.” In *WebSci*, 1–8. ACM.
- Gonzalez-Bailon, Sandra, Ning Wang, Alejandro Rivero, Javier Borge-Holthoefer, and Yamir Moreno. 2014. “Assessing the Bias in Samples of Large Online Networks.” *Social Networks* 38:16–27.
- Granovetter, Mark. 1976. “Network Sampling: Some First Steps.” *American Journal of Sociology* 81 (6): 1287–1303.
- Gray, Robert M. 2011. *Entropy and information theory*. Springer Science & Business Media.
- Green, Jonathon. 2011. *Crooked Talk: Five Hundred Years of the Language of Crime*. Random House.
- Grier, Chris, Kurt Thomas, Vern Paxson, and Michael Zhang. 2010. “@spam: the underground on 140 characters or less.” In *Computer and Communications Security*, 27–37. ACM.
- Guthrie, William. 2010. “NIST/SEMATECH Engineering Statistics Handbook.”
- Hallsmar, Fredrik, and Jonas Palm. 2016. “Multi-class Sentiment Classification on Twitter using an Emoji Training Heuristic.” Master’s thesis, KTH Royal Institute of Technology School of Computer Science and Communication.
- Harfoush, Rahaf. 2009. *Yes We Did! An Inside Look at how Social Media Built the Obama Brand*. New Riders.
- Hong, Liangjie, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsoulouklis. 2012. “Discovering Geographical Topics in the Twitter Stream.” In *International Conference on World Wide Web*, 769–778. Lyon, France: ACM. doi:10.1145/2187836.2187940.

- Hong, Liangjie, and Brian D. Davison. 2010. "Empirical study of topic modeling in Twitter." In *Proc. First Workshop on Social Media Analytics*, 80–88. SOMA '10. Washington D.C., District of Columbia: ACM. doi:10.1145/1964858.1964870.
- Hu, Xia, Jiliang Tang, Huiji Gao, and Huan Liu. 2013. "Unsupervised Sentiment Analysis with Emotional Signals." In *International conference on World Wide Web*, 607–618.
- Humanitarianism in the Network Age*. 2012. United Nations Office for the Coordination of Humanitarian Affairs.
- Issacharoff, Samuel. 2002. "Gerrymandering and Political Cartels." *Harvard Law Review*:593–648.
- Jabbari, Shahin, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. 2016. "Fair Learning in Markovian Environments." In *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 1–25.
- Jerit, Jennifer, and Jason Barabas. 2012. "Partisan perceptual bias and the information environment." *The Journal of Politics* 74 (3): 672–684.
- John, John P, Alexander Moshchuk, Steven D Gribble, and Arvind Krishnamurthy. 2009. "Studying Spamming Botnets Using Botlab." In *Networked Systems Design and Implementation*, 9:291–306.
- Joseph, Kenneth, Peter M Landwehr, and Kathleen M Carley. 2014. "Two 1% Don't Make a Whole: Comparing Simultaneous Samples from Twitter's Streaming API." In *Social Computing, Behavioral-Cultural Modeling and Prediction*, 75–83. Springer.
- Joseph, Kenneth, Chun How Tan, and Kathleen M. Carley. 2012. "Beyond "local", "categories" and "friends": clustering foursquare users with latent "topics"." In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 919–926. UbiComp '12. Pittsburgh, Pennsylvania: ACM. doi:10.1145/2370216.2370422.
- Joseph, Matthew, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. 2016. "Fairness in learning: Classic and contextual bandits." In *Advances in Neural Information Processing Systems*, 325–333.
- Joseph, Matthew, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2016. "Rawlsian fairness for machine learning." *arXiv preprint arXiv:1610.09559*.

- Juris, Jeffrey S. 2012. "Reflections on # Occupy Everywhere: Social media, public space, and emerging logics of aggregation." *American Ethnologist* 39 (2): 259–279.
- Kanich, Chris, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M Voelker, Vern Paxson, and Stefan Savage. 2008. "Spamalytics: An empirical analysis of spam marketing conversion." In *Conference on Computer and Communications Security*, 3–14. ACM.
- Kelly, Ryan, and Leon Watts. 2015. "Characterising the inventive appropriation of emoji as relationally meaningful in mediated close personal relationships."
- Kergl, Dennis, Robert Roedler, and Sebastian Seeber. 2014. "On the Endogenesis of Twitter's Spritzer and Gardenhose Sample Streams." In *Advances in Social Networks Analysis and Mining*, 357–364. IEEE.
- King, Gary, Jennifer Pan, and Margaret E Roberts. 2013. "How censorship in China allows government criticism but silences collective expression." *American Political Science Review* 107 (02): 326–343.
- . 2014. "Reverse-engineering censorship in China: Randomized experimentation and participant observation." *Science* 345 (6199): 1251722.
- Kireyev, K, L Palen, and K Anderson. 2009. "Applications of Topics Models to Analysis of Disaster-Related Twitter Data." In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, vol. 1.
- Kirk, Paul DW, and Michael PH Stumpf. 2009. "Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data." *Bioinformatics* 25 (10): 1300–1306.
- Kossinets, Gueorgi. 2006. "Effects of missing data in social networks." *Social Networks* 28 (3): 247–268. doi:10.1016/j.socnet.2005.07.002.
- Kumar, S, G Barbier, M A Abbasi, and H Liu. 2011. "Tweettracker: An Analysis Tool for Humanitarian and Disaster Relief." In *Fifth International AAAI Conference on Weblogs and Social Media*. Barcelona Spain: AAAI.
- Kumar, Shamanth, Fred Morstatter, and Huan Liu. 2014. *Twitter Data Analytics*. Springer.
- Kumar, Shamanth, Fred Morstatter, Grant Marshall, Huan Liu, and Ullas Nambiar. 2012. "Navigating information facets on Twitter (NIF-T)." In *KDD*, 1548–1551. ACM.

- Kumar, Shamanth, Fred Morstatter, Reza Zafarani, and Huan Liu. 2013. “Whom Should I Follow?: Identifying Relevant Users During Crises.” In *Conference on Hypertext & Social Media (HT)*, 139–147.
- Le, Quoc V, and Tomas Mikolov. 2014. “Distributed representations of sentences and documents.” *arXiv preprint arXiv:1405.4053*.
- Lee, Kyumin, Brian David Eoff, and James Caverlee. 2011. “Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter.” In *ICWSM*, 185–192. AAAI.
- Lee, Kyumin, Prithivi Tamilarasan, and James Caverlee. 2013. “Crowdturfers, Campaigns, and Social Media: Tracking and Revealing Crowdsourced Manipulation of Social Media.” In *ICWSM*.
- Lee, Sangho, and Jong Kim. 2014. “Early filtering of ephemeral malicious accounts on Twitter.” *Computer Communications* 54:48–57.
- Leskovec, Jure, and Christos Faloutsos. 2006. “Sampling from Large Graphs.” In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 631–636.
- Li, Huayi, Arjun Mukherjee, Bing Liu, Rachel Kornfield, and Sherry Emery. 2014. “Detecting campaign promoters on twitter using markov random fields.” In *ICDM*, 290–299. IEEE.
- Li, Rui, Shengjie Wang, and Kevin Chen-Chuan Chang. 2013. “Towards Social Data Platform: Automatic Topic-focused Monitor for Twitter Stream.” *VLDB* 6 (14): 1966–1977.
- Lin, Jianhua. Jan. “Divergence measures based on the Shannon entropy.” *Information Theory, IEEE Transactions on* 37 (1): 145–151. doi:10.1109/18.61115.
- Liu, Huan, Fred Morstatter, Jiliang Tang, and Reza Zafarani. 2016. “The good, the bad, and the ugly: uncovering novel research opportunities in social media mining.” *International Journal of Data Science and Analytics* 1 (3-4): 137–143.
- Lu, Xuan, Wei Ai, Xuanzhe Liu, Qian Li, Ning Wang, Gang Huang, and Qiaozhu Mei. 2016. “Learning from the Ubiquitous Language: An Empirical Analysis of Emoji Usage of Smartphone Users.” In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp ’16. New York, NY, USA: ACM.

- Lu, Yafeng, Feng Wang, and R. Maciejewski. 2014. “Business Intelligence from Social Media: A Study from the VAST Box Office Challenge.” *Computer Graphics and Applications, IEEE* 34, no. 5 (September): 58–69. doi:10.1109/MCG.2014.61.
- Maas, Andrew L, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. “Learning word vectors for sentiment analysis.” In *Association for Computational Linguistics: Human Language Technologies*, 142–150.
- Madlberger, Lisa, and Amai Almansour. 2014. “Predictions Based on Twitter-A Critical View on the Research Process.” In *ICODSE*, 1–6. IEEE.
- Mann, Henry. B., and DONald. R. Whitney. 1947. “On A Test Of Whether One Of Two Random Variables is Stochastically Larger Than The Other.” *The Annals of Mathematics Statistics* 18 (1): 50–60.
- Mathioudakis, Michael, and Nick Koudas. 2010. “TwitterMonitor: trend detection over the twitter stream.” In *Proc. of the 2010 ACM SIGMOD Int’l Conference on Management of data*, 1155–1158. SIGMOD ’10. Indianapolis, Indiana, USA: ACM. doi:10.1145/1807167.1807306.
- Mejova, Yelena, Ingmar Weber, and Michael W Macy. 2015. *Twitter: A Digital Socioscope*. Cambridge University Press.
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. “Distributed representations of words and phrases and their compositionality.” In *Neural Information Processing Systems*, 3111–3119.
- Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. “Linguistic Regularities in Continuous Space Word Representations.” In *NAACL*, 746–751.
- Miller, George A. 1995. “WordNet: a lexical database for English.” *Communications of the ACM* 38 (11): 39–41.
- Miller, Hannah, Daniel Kluver, Jacob Thebault-Spieker, Loren Terveen, and Brent Hecht. 2017. “Understanding Emoji Ambiguity in Context: The Role of Text in Emoji-Related Miscommunication.” In *ICWSM*.
- Miller, Hannah, Jacob Thebault-Spieker, Shuo Chang, Isaac Johnson, Loren Terveen, and Brent Hecht. 2016. ““Blissfully happy” or “ready to fight”: Varying Interpretations of Emoji.” In *ICWSM*.

- Mislove, Alan, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J. Niels Rosenquist. 2011. “Understanding the Demographics of Twitter Users.” In *ICWSM*.
- Mohammad, Saif M, and Peter D Turney. 2013. “Crowdsourcing a Word–Emotion Association Lexicon.” *Computational Intelligence* 29 (3): 436–465.
- Morales, A J, Javier Borondo, Juan Carlos Losada, and Rosa M Benito. 2015. “Measuring political polarization: Twitter shows the two sides of Venezuela.” *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25 (3): 33114.
- Morstatter, Fred, Harsh Dani, Justin Sampson, and Huan Liu. 2016. “Can One Tamper with the Sample API?: Toward Neutralizing Bias from Spam and Bot Content.” In *World Wide Web (WWW) Conference*, 81–82.
- Morstatter, Fred, Jurgen Pfeffer, and Huan Liu. 2014. “When is it Biased?: Assessing the Representativeness of Twitter’s Streaming API.” In *WWW*, 555–556.
- Morstatter, Fred, Jurgen Pfeffer, Huan Liu, and Kathleen M Carley. 2013. “Is the Sample Good Enough? Comparing Data from Twitter’s Streaming API with Twitter’s Firehose.” In *ICWSM*, 400–408.
- Morstatter, Fred, Liang Wu, Tahora H Nazer, Kathleen M Carley, and Huan Liu. 2016. “A New Approach to Bot Detection: Striking the Balance Between Precision and Recall.” In *Advances in Social Networks Analysis and Mining (ASONAM)*, 533–540.
- Mosteller, Frederick. 1949. *The Pre-election Polls of 1948: The Report to the Committee on Analysis of Pre-election Polls and Forecasts*. Vol. 60. Social Science Research Council.
- Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. 2001. “On Spectral Clustering: Analysis and an Algorithm.” In *NIPS*, 849–856. MIT Press.
- Novak, Petra Kralj, Jasmina Smailovic, Borut Sluban, and Igor Mozetic. 2015. “Sentiment of emojis.” *PLoS ONE* 10 (12): e0144296.
- Olteanu, Alexandra, Carlos Castillo, Fernando Diaz, and Emre Kiciman. 2016. “Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries.” *SSRN* (December).
- O’Neil, Cathy. 2016. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown Publishing Group (NY).

- Ono, Masataka, Makoto Miwa, and Yutaka Sasaki. 2015. “Word Embedding-Based Antonym Detection using Thesauri and Distributional Information.” In *NAACL*, 984–989.
- Ott, Myle, Claire Cardie, and Jeff Hancock. 2012. “Estimating the prevalence of deception in online review communities.” In *WWW*, 201–210. ACM.
- Ott, Myle, Claire Cardie, and Jeffrey T Hancock. 2013. “Negative Deceptive Opinion Spam.” In *HLT-NAACL*, 497–501.
- Pavalanathan, Umashanthi, and Jacob Eisenstein. 2015. “Emoticons vs. emojis on Twitter: A causal inference approach.” *arXiv preprint arXiv:1510.08480*.
- Pennington, Jeffrey, Richard Socher, and Christopher D Manning. 2014. “Glove: Global Vectors for Word Representation.” In *EMNLP*, 14:1532–43.
- Pozdnoukhov, Alexei, and Christian Kaiser. 2011. “Space-time dynamics of topics in streaming text.” In *Proc. of the 3rd ACM SIGSPATIAL Int’l Workshop on Location-Based Social Networks*, 1–8. LBSN ’11. Chicago, Illinois: ACM. doi:10.1145/2063212.2063223.
- Ranganath, Suhas, Xia Hu, Jiliang Tang, and Huan Liu. 2016. “Understanding and identifying advocates for political campaigns on social media.” In *WSDM*, 43–52. ACM.
- Ratkiewicz, Jacob, Michael Conover, Mark Meiss, Bruno Goncalves, Alessandro Flammini, and and Filippo Menczer. 2011. “Detecting and Tracking Political Abuse in Social Media.” In *ICWSM*, 297–304. AAAI.
- Ratkiewicz, Jacob, Michael Conover, Mark Meiss, Bruno Goncalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. 2011. “Truthy: mapping the spread of astroturf in microblog streams.” In *World Wide Web Companion*, 249–252. ACM.
- Recuero, Raquel, and Ricardo Araujo. 2012. “On the rise of artificial trending topics in twitter.” In *Proceedings of the 23rd ACM conference on Hypertext and social media*, 305–306. HT ’12. Milwaukee, Wisconsin, USA: ACM. doi:10.1145/2309996.2310046.
- Rehurek, Radim, and Petr Sojka. 2010. “Software Framework for Topic Modelling with Large Corpora.” In *Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA.

- Ryan, Thomas P. 2011. *Statistical Methods for Quality Improvement*. Vol. 840. Wiley.com.
- Sabidussi, G. 1966. "The centrality index of a graph." *Psychometrika* 69:581–603.
- Sampson, Justin, Fred Morstatter, Ross Maciejewski, and Huan Liu. 2015. "Surpassing the Limit: Keyword Clustering to Improve Twitter Sample Coverage." In *Conference on Hypertext & Social Media (HT)*, 237–245.
- Shirky, Clay. 2008. *Here comes everybody: The power of organizing without organizations*. Penguin.
- Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. "Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks." In *EMNLP*, 254–263.
- Socher, Richard, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. "Parsing with compositional vector grammars." In *In Proceedings of the ACL conference*. Citeseer.
- Sofean, Mustafa, and Matthew Smith. 2012. "A real-time architecture for detection of diseases using social networks: design, implementation and evaluation." In *Proceedings of the 23rd ACM conference on Hypertext and social media*, 309–310. HT '12. Milwaukee, Wisconsin, USA: ACM. doi:10.1145/2309996.2310048.
- Steiglitz, Kenneth. 2007. *Snipers, skills, & sharks: eBay and human behavior*. Princeton University Press.
- Stone-Gross, Brett, Thorsten Holz, Gianluca Stringhini, and Giovanni Vigna. 2011. "The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns." In *USENIX Workshop on Large-Scale Exploits and Emergent Threats*.
- Tang, Duyu, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014. "Coooooll: A Deep Learning System for Twitter Sentiment Classification." In *International Workshop on Semantic Evaluation*, 208–212.
- Tang, Duyu, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. "Learning sentiment-specific word embedding for twitter sentiment classification." In *Association for Computational Linguistics*, 1:1555–1565.



- Thomas, Kurt, Chris Grier, and Vern Paxson. 2012. "Adapting social spam infrastructure for political censorship." In *Conference on Large-Scale Exploits and Emergent Threats*. USENIX.
- Thomas, Kurt, Chris Grier, Dawn Song, and Vern Paxson. 2011. "Suspended accounts in retrospect: an analysis of twitter spam." In *Internet Measurement Conference*, 243–258. ACM.
- Thonnard, Olivier, and Marc Dacier. 2011. "A strategic analysis of spam botnets operations." In *Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, 162–171. ACM.
- Tigwell, Garreth W, and David R Flatla. 2016. "Oh that's what you meant!: reducing emoji misunderstanding." In *MobileHCI*, 859–866. ACM.
- Tsur, Oren, and Ari Rappoport. 2012. "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities." In *Proceedings of the fifth ACM international conference on Web search and data mining*, 643–652. WSDM '12. Seattle, Washington, USA: ACM. doi:10.1145/2124295.2124320.
- Tufekci, Zeynep. 2014. "Big Questions for Social Media Big Data: Representativeness, Validity and Other Methodological Pitfalls." In *ICWSM*, 505–514.
- Tufekci, Zeynep, and Christopher Wilson. 2012. "Social media and the decision to participate in political protest: Observations from Tahrir Square." *Journal of Communication* 62 (2): 363–379.
- Tumasjan, Andranik, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welp. 2010. "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment." *ICWSM* 10:178–185.
- Vieweg, Sarah, Amanda L Hughes, Kate Starbird, and Leysia Palen. 2010. "Microblogging during two natural hazards events: what twitter may contribute to situational awareness." In *Proceedings of the SIGCHI conference on human factors in computing systems*, 1079–1088. ACM.
- Wakamiya, Shoko, Ryong Lee, and Kazutoshi Sumiya. 2011. "Crowd-based urban characterization: extracting crowd behavioral patterns in urban areas from Twitter." In *Proc. of the 3rd ACM SIGSPATIAL Int'l Workshop on Location-Based Social Networks*, 77–84. LBSN '11. Chicago, Illinois: ACM. doi:10.1145/2063212.2063225.

- Walther, Joseph B, and Kyle P D’Addario. 2001. “The impacts of emoticons on message interpretation in computer-mediated communication.” *Social science computer review* 19 (3): 324–347.
- Wang, Alex Hai. 2010. “Detecting spam bots in online social networking sites: a machine learning approach.” In *Data and Applications Security and Privacy XXIV*, 335–342. Springer.
- Wang, Gang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y Zhao. 2012. “Serf and turf: crowdturfing for fun and profit.” In *WWW*, 679–688. ACM.
- Wasserman, S, and K Faust. 1994. *Social Network Analysis: Methods and Applications*. Cambridge, MA: Cambridge University Press.
- Watts, Duncan J, and S H Strogatz. 1998. “Collective dynamics of ’small-world’ networks.” *Nature* 393:440–442.
- Wei, Wei, Kenneth Joseph, Huan Liu, and Kathleen M Carley. 2015. “The fragility of Twitter social networks against suspended users.” In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, 9–16. ACM.
- Wijeratne, Sanjaya, Lakshika Balasuriya, Amit Sheth, and Derek Doran. 2016. “Emojinet: Building a machine readable sense inventory for emoji.” In *International Conference on Social Informatics*, 527–541. Springer.
- Wilson, Christo, Bryce Boe, Alessandra Sala, Krishna PN Puttaswamy, and Ben Y Zhao. 2009. “User interactions in social networks and their implications.” In *Proceedings of the 4th ACM European Conference on Computer Systems*, 205–218. ACM.
- Wong, Felix Ming Fai, Chee Wei Tan, Soumya Sen, and Mung Chiang. 2013. “Quantifying Political Leaning from Tweets and Retweets.” *ICWSM* 13:640–649.
- Wu, Liang, Fred Morstatter, and Huan Liu. 2016. “SlangSD: Building and Using a Sentiment Dictionary of Slang Words for Short-Text Sentiment Classification.” *arXiv preprint arXiv:1608.05129*.
- Wu, Yongkai, and Xintao Wu. 2016. “Using loglinear model for discrimination discovery and prevention.” In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, 110–119. IEEE.

- Xie, Yinglian, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten, and Ivan Osipkov. 2008. "Spamming botnets: signatures and characteristics." *ACM SIGCOMM Computer Communication Review* 38 (4): 171–182.
- Yang, Lei, Tao Sun, Ming Zhang, and Qiaozhu Mei. 2012. "We know what @you #tag: does the dual role affect hashtag adoption?" In *Proc. of the 21st int'l conference on World Wide Web*, 261–270. WWW '12. Lyon, France: ACM. doi:10.1145/2187836.2187872.
- Ye, Shaozhi, Juan Lang, and Felix Wu. 2010. "Crawling online social graphs." In *Web Conference (APWEB), 2010 12th International Asia-Pacific*, 236–242. IEEE.
- Yin, Zhijun, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. 2011. "Geographical topic discovery and comparison." In *Proceedings of the 20th international conference on World wide web*, 247–256. WWW '11. Hyderabad, India: ACM. doi:10.1145/1963405.1963443.
- Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social media mining: an introduction*. Cambridge University Press.
- Zafarani, Reza, and Huan Liu. 2013. "Connecting Users across Social Media Sites: A Behavioral-Modeling Approach." In *KDD*, 41–49.
- . 2015. "10 Bits of Surprise: Detecting Malicious Users with Minimum Information." In *Conference on Information and Knowledge Management*, 423–431. ACM.
- Zeng, Jiaming, Berk Ustun, and Cynthia Rudin. 2016. "Interpretable classification models for recidivism prediction." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*.
- Zhang, Lu, Yongkai Wu, and Xintao Wu. 2016. "Situation testing-based discrimination discovery: A causal inference approach." In *Proceedings of IJCAI*, vol. 2016.
- Zimmer, Michael. 2010. "'But the data is already public': on the ethics of research in Facebook." *Ethics and Information Technology* 12 (4): 313–325.
- Zou, Will Y, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. "Bilingual Word Embeddings for Phrase-Based Machine Translation." In *EMNLP*, 1393–1398.