Spam analysis and detection on microblog

Sedhai Surendra

2017

Sedhai Surendra. (2017). Spam analysis and detection on microblog. Doctoral thesis,
Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/69594>

<https://doi.org/10.32657/10356/69594>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**SPAM ANALYSIS AND DETECTION ON
MICROBLOG**

SURENDRA SEDHAI

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
2017**

SURENDRA SEDHAI

SPAM ANALYSIS AND DETECTION ON MICROBLOG

SURENDRA SEDHAI

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2017

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Associate Professor **Aixin Sun**, for his guidance and support along these years. His passion and enthusiasm in his work have always inspired me. His detailed-oriented approach and supportive nature have greatly helped me to grow as a researcher. It is my great pleasure to be his student.

I would like to thank Associate Professor **Cong Gao**, for his guidance and suggestions on my research as a TAC member. I would also like to thank Associate Professor **Chen Lihui** for her support and encouragement as a TAC member.

I would like to express my gratitude to my friends at NTU for a friendly environment and stimulating discussions. My special thanks to Yao Yangjie, Hu Longke, Ma Zongyang, Prabhash Kumarasinghe, Ankit Das, Han Jialong, Han Jianglei, Li Chenliang, Phan Cong Minh, Shanshan Feng, Tu Hongkui, Xutao Li, Zheng Xin, Zhong Xiaosh and all my supportive friends at NTU.

I would like to thank my sisters Bhagawati Sedhai, Suma Sedhai and Shova Sedhai and brother Suvash Sedhai for their unconditional love and support. I am grateful to my in-laws Ram Prasad Silwal, Laxmi Mainali, Manoj Silwal for their encouragement. I thank my niece Aatmiya Silwal and Samipya Mainali for cheering me up.

Finally, I would like to thank my wife Kripa Upreti Sedhai for her support all these years. Her encouragement and endless support have always inspired me to work hard every day. I would like to thank my parents, Keshav Raj Sedhai and Chandra Kumari Sedhai, who have always believed in me and been unconditionally supportive throughout my life.

Contents

Acknowledgments	i
Abstract	v
List of Figures	vi
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Motivation	3
1.3 Contribution	5
1.4 Dissertation Organization	7
2 Literature Review	9
2.1 Spam on Twitter	9
2.2 Twitter Dataset Analysis	18
2.3 Spam on Other Communication Media	23
2.4 Hashtag Recommendation	24
2.5 Summary	27
3 HSpam14 Dataset Creation	28
3.1 Data Collection and Statistics	29
3.1.1 Data Crawling	29
3.1.2 Data Statistics	30
3.2 Tweet Annotation	32

3.2.1	Heuristics-based Tweet Selection	35
3.2.2	Cluster-based Manual Annotation	36
3.2.3	Reliable Ham Tweet Detection	47
3.2.4	Annotation by EM-Algorithm	48
3.2.5	Summary	49
3.3	Conclusion and Limitation	49
4	HSpam14 Dataset Analysis	51
4.1	Hashtag Usage in Spam and Ham	52
4.2	Content Analysis	61
4.2.1	Words and Content-based Metadata	61
4.2.2	Near-duplicate Tweets	63
4.3	User-based Analysis	66
4.3.1	Followers and Followees	67
4.3.2	User Profile-based Analysis	69
4.4	Case Study: Spam Detection	72
4.5	Discussion	75
4.6	Conclusion	76
5	Real Time Spam Detection	78
5.1	Semi-supervised Spam Detection	79
5.1.1	Spam Tweet Detection in Real Time	80
5.1.2	Model Update in Batch Mode	82
5.1.3	Computational Efficiency	85
5.2	Experiment and Discussion	86
5.2.1	Result and Discussion	87
5.2.2	Feature Analysis	91
5.3	Conclusion	93

6	Effect of Spam on Hashtag Recommendation	94
6.1	Hashtag Recommendation	96
6.1.1	Hashtags from Similar Tweets	97
6.1.2	Hashtags from Similar Web Pages	97
6.1.3	Hashtags based on the Domain of the Link	98
6.1.4	Random Walk with Restart (RWR) model	98
6.1.5	Language Translation (LT) model	99
6.1.6	Pairwise Learning to Rank	99
6.2	Dataset	101
6.3	Experiment	101
6.4	Method Comparison	103
6.5	A Case Study	106
6.5.1	Result and Discussion	107
6.6	Conclusion	110
7	Conclusion and Future Work	112
7.1	Conclusion	112
7.2	Future Work	115
	References	118

Abstract

Micro-blogging platforms such as Twitter and Weibo are popular platforms for information collection and dissemination. Due to ease of posting content and the huge social graph, micro-blogging platforms have attracted the attention of not only legitimate users but also spammers. Increasing activities of spammers on micro-blogging platforms make spam a serious problem which also affect user experience. Due to quick propagation of content in microblogging services, it is highly desirable to detect spam in real-time to minimize its impact. Hence, real-time spam detection technique that leverage fine-grained information is essential. As Twitter is one of the most popular micro-blogging platforms, in this dissertation we used Twitter dataset to study spam on microblog.

Spam detection is an active area of research on Twitter. Most of the studies are focused on identifying spammers. Spam issues on Twitter are tackled by blocking spammers detected by the system. A user account that mistakenly grant permission to a malicious third party application may get blocked due to the posts by the application on behalf of the user. Similarly, compromised microblog accounts may get blocked because of the tweets posted by hijackers. Further, spammers are generally detected only after posting many spam tweets. To address the issue of spam on Twitter, we have used tweet-level spam detection is necessary.

Tweet-level spam detection is inherently a challenging task as tweet is short and noisy text. Spam tweets heavily exploit hashtags to promote the tweets to the wider audience. Hence, we focused on hashtag oriented spam detection by collecting tweets using trending hashtags as a query. Further, we propose an effective way of labeling

tweets for generating a dataset for such task. To the best of our knowledge, there was no any benchmark dataset hence we present HSpam14, a public dataset consisting of 14 million tweets labeled with the proposed labeling technique. We made a detailed tweet-level analysis based on hashtags and tweet content, and user-level analysis based on user profiles. Detailed understanding about spam tweets and legitimate tweets, which are also known as ham tweets, are utilized to design spam tweet detection system. Unlabeled tweets are easy to obtain and they can be utilized to improve system performance and to deal with evolving spamming activities. Hence, we proposed semi-supervised real-time spam detection system to effectively identify spam tweets.

Spams in a microblog introduce problems in several functionalities such as search, recommendation, and text analysis. As a case study, we analyze the effect of spam tweets on hashtag recommendation using the HSpam14 dataset. We observed that features and methods that are effective for spam tweet collection may not be effective for legitimate tweets. Our study shows that experiment conducted on spammy dataset gives misleading results. Hence, it is crucial to perform spam filtering before conducting any analysis on Twitter.

In a nutshell, this dissertation elucidates the effectiveness of tweet-level spam detection and different aspects of spam and ham tweets and also paves the way for further research on fine-grained spam detection on microblog.

List of Figures

2.1	Overview of social honeypot-based approach [LCW10]	11
2.2	System overview of real-time URL spam detection [TGM ⁺ 11]	12
2.3	Crowdturfing process [WWZZ14]	15
2.4	System overview of unsupervised social network spam detection system [TGC ⁺ 13]	16
3.1	User frequency distribution	30
3.2	Hashtag/mention frequency distribution	31
3.3	No. of hashtags/mentions per tweet	32
3.4	Overview of the four major steps in labeling spam/ham tweets	34
3.5	Size distribution of the near-duplicate clusters	37
4.1	Cumulative fraction of spam/ham tweets with different number of hashtags per tweet	52
4.2	Orthography and position of hashtags in spam and ham tweets	54
4.3	Hashtag co-occurrence graph. Green node is the least spammy hashtag, blue is moderately spammy and red is the most spammy hashtag.	58
4.4	Semantic similarity of co-occurring hashtags and the fraction of times the pair of hashtags appear in spam tweets. The number of hashtag pairs in each hexagon is indicated by its color.	59
4.5	Differences in content metadata in spam and ham tweets	62

4.6	The distribution of the number of clusters against the number of users posting these tweets in each cluster. Tweets in each cluster contain links to one particular domain.	65
4.7	Follower distribution of ham users	66
4.8	Follower distribution of spam users	67
4.9	Followee distribution of ham users	68
4.10	Followee distribution of spam user	69
4.11	Distributions of number of tweets of spam user. The number of tweet-user pair in a bin is indicated by its color.	70
4.12	Distributions of number of tweets of spam and ham users. The number of tweet-user pairs in a bin is indicated by its color.	71
4.13	Fraction of spam/ham users providing information in user profile page	72
5.1	System Overview	80
5.2	Precision of supervised and semi-supervised approach	88
5.3	Recall of supervised and semi-supervised approach	89
5.4	F1-score of supervised and semi-supervised approach	90
5.5	Accumulated numbers of blacklisted domains, confident clusters and trusted users	91
5.6	Precision of confidently labeled clusters/tweets	92
6.1	Hashtag, hyperlinked tweet, and linked document	96
6.2	Entity-hashtag graph	97
6.3	Hit rate of the 7 methods	103
6.4	Precision of the 7 methods	104
6.5	Recall of the 7 methods	105
6.6	Hit rate of the 7 methods on spam, ham and merged datasets	107
6.7	Precision of the 7 methods on spam, ham and merged datasets	108
6.8	Recall of the 7 methods on spam, ham and merged datasets	109

List of Tables

3.1	The top-20 most popular hashtags. All these 20 hashtags appeared as trending up/down hashtags which were used for sampling the tweets.	33
3.2	Example tweets from two near-duplicate clusters	38
3.3	Measures of intra-/inter-cluster samples of tweets	38
3.4	Initial manual annotation of near-duplicate clusters	41
3.5	Statistics of the labeled tweets in HSpam14 dataset	48
4.1	Top 20 most used hashtags in ham/spam tweets. Hashtags are in descending order of their frequencies. The most frequent ham hashtag #FF appears in 43K tweets, while the 20 th ham hashtag #SS5INADay2 appears in 9K tweets. The most frequent spam hashtag #TEAM-FOLLOWBACK appears in 661K tweets, and the 20 th spam hashtag #HITFOLLOWSTEAM appears in 127K tweets	53
4.2	Top-20 hashtags with the highest spammy-index. All hashtags listed in these two tables have frequency larger than 5000, and the hashtags are case-sensitive.	55
4.3	Top-20 hashtags with the lowest spammy-index. All hashtags listed in these two tables have frequency larger than 5000, and the hashtags are case-sensitive.	56
4.4	Top-20 most spammy words by Odds Ratio	57
4.5	Ten co-occurring hashtag pairs in spam and ham tweets having low and high Jensen-Shannon divergence	60

4.6	The top 20 most frequently used words in ham and spam tweets . . .	62
4.7	Top-20 most commonly used words in ham and spam users' profile descriptions, ordered by their frequencies.	72
4.8	The 15 features derived from this study and ranked by Gini coefficient	73
4.9	The top-15 terms (words and bi-/tri-grams) ranked by Gini coefficients and the top-15 hashtags ranked by spammy index. The matched terms and hashtags in both lists are highlighted in boldface.	74
5.1	Features used to represent tweets and clusters for classification; FoT means <i>Fraction of Tweets</i> and FoU means <i>Fraction of Uers</i> . The top- 15 most effective features for tweet classification and cluster classification based on the Gini impurity score are indicated by the numbers (x) following the features ($1 \leq x \leq 15$).	83
5.2	Fraction of tweets detected by each detector	87
5.3	Top-15 most effective vocabulary features	93
6.1	The 7 methods in evaluation	101

Chapter 1

Introduction

1.1 Background

Over the years, social media has become an integral part of our day to day life. Online social sites provide platforms to create, share, or exchange ideas, videos, pictures and other information. Online social network, such as Twitter, Facebook, LinkedIn, Youtube, and Flickr are not only used as a platform for communication but also as source and repository of information. Among the different social media, micro-blogging platforms such as Twitter, Weibo have become widely popular as information dissemination and collection is highly effective in these platforms. Due to the effective dissemination of content and active participation of users, there is an exponential growth of user-generated content on micro-blogging sites. Microblogging platforms have not only attracted a lot of legitimate users but also spammers which makes spam as a serious issue on micro-blog. In this dissertation, we focus on detecting spam tweets and analyzing spam and ham tweets from different aspects.¹

Twitter is by far the most popular and active micro-blogging service and has become one of the most efficient platforms for online communication and information sharing. It has been reported that Twitter streams contain significant amounts of meaningless messages [HW11], rumors [QRRM11] and polluted content [SS15], which

¹Without loss of generality, the proposed methods work on any micro-blogging platforms

pollutes the information stream and can forward traffic to malicious sites. Spam is not a recent problem and has been studied since the inception of the Internet. Various techniques have been developed to tackle the spam problem on traditional communication media such as email, web page, and blog. However, existing methods do not work well for detection of spam on Twitter due to the characteristics of tweets, such as short, informal and noisy.

Spammers utilized social graphs, popular hashtags and compromised legitimate accounts to reach a larger audience. They use techniques such as URL shortening, posting covering legitimate tweets, mimicking legitimate user's behaviors to avoid detection. Further, creating an account on Twitter is very easy which makes it convenient for spammers to construct a large network of spammers. Research studies have been done to detect such spammers on Twitter [HTZL13, LCW10]. However, before a spam account is identified tweets from the account may pollute timeline of a large number of users. Studies show that there are underground markets that create spam network to post tweets on behalf of its clients [GVK⁺12]. Recently, sophisticated techniques such as social bot, crowdturfing have also been heavily exploited to spam on Twitter [WWZ⁺12]. Hence, we need a better approach that leverages fine-grained signals to tackle spam problem in micro-blogging services. To this end, we work on tweet-level spam detection which complements spammer detection on Twitter to effectively address spam problem on Twitter.

Twitter has been widely used to share information during natural disasters such as earthquakes [SOM10], hurricanes [HP09] and wildfires [DLSL09]. Further, Twitter has also been exploited to analyze the opinion of users about a particular product [KWM11]. Companies are using Twitter for expanding user base, reachability and doing business analytics to collect information such as public view. Similarly, event detection, sentiment analysis, election prediction, hashtag recommendations are some of the important applications on Twitter [AK15, AXV⁺11, TSSW10, SS14]. A broad range of applications based on tweets also makes Twitter as an attractive platform for spammers. Due to the increasing number of spam on Twitter, timelines of users are

highly polluted which has adverse effects on user experience as well as applications based on Twitter dataset.

In this dissertation, we focus on a fine-grained hashtag-oriented spam detection on Twitter, analysis of different aspects of spam and ham tweets and understanding the effect of spam on Twitter. To the best of our knowledge, there is no other publicly available annotated Twitter dataset for spam tweet detection task. Most of the existing work on spam have been focused on detecting spammers rather detecting spam tweet. This is mainly because of lack of publicly available annotated Twitter dataset.

Furthermore, the hashtag is an important component of a tweet which has been heavily exploited by spammers to promote their tweets. As hashtags are heavily exploited by spammers, we focus on hashtag-oriented spam detection by collecting tweets using trending hashtags as a query. We propose an effective way of labeling tweets to generate a dataset for spam detection task. We created HSpam14, a public dataset consisting of 14 million tweets labeled with the proposed labeling technique. Further, HSpam14 dataset is utilized to analyze similarities and differences between spam and ham tweets/users in various aspects based on hashtag, tweet content, and user profile. Understanding different aspects of spam and ham tweets, we proposed a fine-grained real-time spam detection framework. We also perform a case study to understand the effect of spam tweets on hashtag recommendation. In this dissertation, we formulate effective tweet-level spam detection techniques and present detailed analysis and comparison of spam and ham tweets based on different aspects.

1.2 Motivation

Micro-blogging services have attracted the attention of not only legitimate users but also spammers. Grier *et al.* reported that 0.13% of messages advertised on Twitter are clicked, which is two orders of magnitude higher than that of email spam [GTPZ10]. High click-rate and effective message propagation make Twitter an attractive platform for spammers. Hence, there are increasing spamming activities on Twitter which have

adversely affected user experience as well as many tasks such as user behavior analysis, recommendation, and sentiment analysis.

Spam is prevalent in all types of online communication medium. Spam detection on traditional communication media such as email, web, and blog has already got a lot of attentions from research communities and a wide range of techniques have been proposed to address spam on these communication media [Cor08, SH12, ANC10]. However, techniques developed for these formal documents can not be directly utilized on Twitter. Most existing studies on Twitter spam focus on account blocking, which is to identify and block spam users, or spammers [HTZL13, LCW10]. This approach is less effective for spammers who may act as legitimate users by posting non-spam content regularly. Blocking spammers may even hurt a legitimate user who happens to grant permission to a third-party application that posts spammy tweets under her username. This legitimate account may be blocked because of such spam tweets. Furthermore, spammers mimic tweet content and strategies of legitimate users to make their tweets and activities look like legitimate [BMRA10]. While identifying and blocking spammer accounts remain a crucial and challenging task, tweet-level spam detection is essential to fight against spamming at a more fine-grained level.

Tweet-level spam detection is essential to fight against spamming in a more fine-grained level. However, spam tweet detection is intrinsically a difficult task because of limited information in an individual tweet. In addition, there was no benchmark dataset for spam tweet detection task which is essential to carry out the experiments and evaluate the performance. Hence, we took the first step to create a large-scale tweet collection for hashtag-oriented spam research named as HSpam14 and made the dataset publicly available [SS15].

Understanding different aspects of spam and ham tweets helps to identify spam on Twitter at content-level. Hence, HSpam14 dataset can be utilized to analyze different aspects of spam and ham tweets. Tweet-level spam analysis based on hashtags and tweet content and user-level spam analysis based on user profiles give a better insight of spam on Twitter. Information such as usage of sentiment word, link, exclamation

sign, question mark, capitalization, positive emoticon, negative emoticon and money sign can be easily extracted in the HSpam14 dataset. Comparing the information of spam and legitimate tweets give a fine-grained view of the attributes exploited by spammers.

Spammers keep changing their strategies over time and remain undetected for a long time. Hence, such spammers may post many tweets before they get detected by spam detection system. In order to deal with the issue, fine-grained real-time spam detection is essential to prevent a large number of users being affected by the spam tweets. Robust spam detection system needs to continuously include new spamming information in the model to effectively detect evolving spamming activities. To address the issue of continuously evolving spam on Twitter, confidently labeled new tweets can be included in the training dataset to capture new and evolving spamming activities.

Growing number of spam activities on Twitter may not only affect user experiences [JHTS11] but also the applications using Twitter data, from many perspectives including feature selection, algorithm formulation, and system evaluation. The presence of spam tweets may result in misleading results because features and methods that are effective in the presence of spam tweets may not necessarily be effective for ham tweets. Hence, it would be helpful to get the insights about the effect of spam on an application.

1.3 Contribution

In this research, we focus on spam detection on the micro-blogging platform. Focusing on the Twitter dataset, we have made the following four major contributions.

- **HSpam14 dataset creation.** We have created the HSpam14 dataset, a collection of 14 million annotated tweets in English for hashtag-oriented spam research. We collected trending hashtags on a daily basis from hashtags.org and used the collected hashtags as query keywords to collect tweets through Twitter streaming API, for two months time. We have proposed a framework to

annotate 14 million tweets which consist of four major steps. The first step is heuristic-based tweets selection where we search for the tweets that are likely to be spam. The selection was based on popular hashtags, and keywords related to adult content, money gain and others. In near-duplicate cluster based annotation step, similar tweets are grouped into clusters using the MinHash-based algorithm. The annotation process includes both manual annotation and confident k-nearest neighbor based annotation with human intervention in the process. Tweets posted by the same users and tweets containing links from the same domains are also clustered and annotated in this step. In the next step, reliable ham tweets are identified utilizing the labeled tweets of previous steps. In the last step, we labeled remaining unlabeled tweets using Expectation-Maximization algorithm.

- **Analysis of HSpam14 dataset.** We have made a detailed analysis of spam and ham tweets based on the usage of hashtags, tweet content, and user profiles. The study shows that spam tweets contain more hashtags compared to ham tweets, and position and orthography of hashtags used in spam and ham tweets are also significantly different. Our analysis shows that spammers exploit globally popular hashtags to promote content to a wider audience. Further, spam tweets are emphasized to grab the attention of users by using exclamation sign, capitalized words and money sign. We also observed that spammers post similar content multiple times and distribute the spamming activity among many accounts to promote their content. User profiles based analysis shows that there are significant differences in user profile information of spammers and legitimate users. Legitimate users are likely to provide more detailed information in their profile compared to spammers.
- **Real-time semi-supervised spam tweet detection.** We have proposed a framework for a real-time spam detection using semi-supervised approach. It consists of four light-weight detectors that are designed based on observations made from the HSpam14 dataset, and the detectors are computationally effec-

tive, suitable for real-time detection. More importantly, our detectors utilize classification techniques at two levels, tweet-level and cluster-level. With this flexible design, any features that may be effective in spam detection can be easily incorporated into the detection framework. The framework starts with a small set of labeled samples, and updates the detection models in a semi-supervised manner by utilizing the confidently labeled tweets from the previous time window. This semi-supervised approach helps to learn new spamming activities, making the framework more robust in identifying spam tweets.

- **Effect of spam on hashtag recommendation.** As a case study, the effect of spam on hashtag recommendation are analyzed to understand the effect of spam on Twitter. It is observed that simple approaches achieve high accuracy in recommending hashtags for spam tweets. This is probably due to the extensive use of popular hashtags in spam tweets. It is observed that features and methods which are effective for spam tweets dataset may not be effective to ham tweets dataset. It is observed that performance on spam dataset is substantially better than ham dataset across all evaluation measures. The same hashtag recommendation method may achieve much better accuracy on the dataset containing spam tweets than a dataset containing only ham tweets. Without removing spam in a data collection, the results obtained for a hashtag recommendation method may be (misleadingly) better than its actual performance.

1.4 Dissertation Organization

In the next Chapter, we provide literature review of related studies. In Chapter 3, the HSpam14 creation process is presented and in Chapter 4 a detailed analysis of spam and ham tweets are presented. In Chapter 5, we present description of a framework for a real-time spam detection using semi-supervised approach. In Chapter 6, the effect of spam on hashtag recommendation are analyzed to understand the effect of

spam on Twitter. Finally, Chapter 7 concludes this dissertation and suggests a few promising directions for future research.

Chapter 2

Literature Review

In this chapter, we review research studies on spam detection on Twitter. Spam related studies on different communication platforms provide motivation to design spam detection system on Twitter. Hence, we review spam detection on various communication media namely email, web, and product review. We also review the studies on hashtag recommendation as we have used hashtag recommendation as a case study to analyze the effect of spam on Twitter.

2.1 Spam on Twitter

Inherent characteristics of tweets such as short, informal and noisy make spam detection a challenging task on Twitter. Spammers use Twitter-specific techniques to evade traditional spam detection system. Recently, spammers are aggressively exploiting Twitter to promote their content. Hence, studies have been done to design effective spam detection system on Twitter [LCW10, HTL14a, FVD⁺14].

Spammers try to mimic the behaviors of legitimate users to remain undetected from spam detection system [TGSP11, VG13]. On Twitter, many users simply follow back when they are followed by someone for the sake of courtesy [WLJH10]. Using the simple trick spammers may gain many followers and look like a legitimate user. Likewise, URL shortening has become the norm among legitimate users for sharing

URLs on Twitter, mainly due to the 140-character limit per message. Exploiting the behavior, spammers camouflage malicious content and it also helps to improve the click-through rate of their spam URLs [WNL⁺13]. Besides gaining followers and using malicious short URLs, spammers also post legitimate tweets to dilute the proportion of their spam tweets [YHG13]. As mimicking the behavior of the legitimate user is not difficult, spammers mimic behaviors of legitimate users on Twitter to evade spam detection system. This makes spam detection task on Twitter even more challenging.

Even though spammers try to mimic normal users' behaviors, they show some kind of anomalous activities which are less likely to be done by normal users. Hence, an anomaly-based approach has also been utilized to deal with spam on Twitter. Miller *et al.* proposed cluster-based anomaly detection approach to detect spammers. They used historical tweets and network information to detect spammer on Twitter. A co-clustering algorithm based on non-negative matrix factorization is also utilized to identify anomalous users and messages [SYW⁺14]. It is observed that tweets containing link and pictures are more suspicious compared to other tweets. Similarly, topical anomalies of tweets are investigated comparing topical information from a tweet and content of the document pointed by the URLs in the tweet. The study determined the divergence of a topic of tweets and referenced document which is found to be an important feature to identify malicious tweets [ATS12].

Machine learning techniques have been extensively used to address the issues of spam on Twitter [LEC11, SKV10, MDD⁺14, ZZL12]. Social honeypot-based approach exploits behaviors of spammers and machine learning techniques to effectively detect spammer on Twitter. Figure 2.1 shows the overview of a social honeypot-based system where social honeypots are deployed to trap users who follow any users [LCW10]. The users who are detected by the honeypot have a high likelihood of being a spammer. Such detected accounts are classified using standard classifiers. The classifiers utilized features derived from user demographics, social graph, tweet content and temporal information to detect content polluters. Those accounts which are classified as spam accounts are manually inspected and added in the training set which helps to improve

Figure 2.1: Overview of social honeypot-based approach [LCW10]

the coverage of spam pattern over time. Social honeypot-based spammers detection system is reported to be effective in identifying spam accounts [LCW10].

Likewise, proven machine learning techniques in other applications have also been explored to detect spam on Twitter. Hu *et al.* formulated spam detection task as an optimization problem and utilized both the social network and content information in the framework [HTZL13]. They have used network information as one of the regularizers in their optimization formulation. They have also proposed an algorithm to solve a non-smooth convex optimization problem to design robust spam detection system. Experiment results show that their optimization framework can effectively incorporate both content and network information for spam detection task.

Inspired by psychological findings in the real world, the importance of sentiment analysis for spammer detection task on Twitter is studied in [HTGL14]. Hu *et al.* incorporate the sentiment differences between spammers and normal users in their optimization based spam detection framework. The study shows that sentiment information is crucial to detect spammers effectively on Twitter [HTGL14]. In other studies also sentiment based feature are found to be discriminative to detect social

Figure 2.2: System overview of real-time URL spam detection [TGM⁺11]

spam [RCM⁺11]. The presence of emoticons and sentimental words are found to be discriminative features for spam detection task.

The optimization-based approach is further extended utilizing information from external spam resources to detect spam on Twitter [HTL14a]. They formulate spam detection task as matrix factorization problem which effectively integrating information from other media and content information from microblog. They have utilized two constraints in their matrix factorization framework. The first constraint is based on lexical information from the other media source and the second constraint is regularization learned from microblogging content information. It is shown that proposed method is effective to combine external information from other media with content information from microblogging for spam detection task. However, for the continuous stream of data, online learning approach is necessary to update the model incrementally. Hence, the authors also proposed online learning algorithms to cope with the fast evolving social spamming activities [HTL14b].

There is a prevalence of malicious tweets containing suspicious URLs for spamming, phishing, and malware distribution on Twitter [CABK11, TN10, TGM⁺11]. In order to address the issue, Thomas *et al.* proposed an efficient system to detect spam URLs. Figure 2.2 illustrates the overview of their system [TGM⁺11] which consists of four steps namely, URL aggregation, feature collection, feature extraction, and classification. In the system, they collected URLs from email and tweets and for these URLs they extract the features. Such extracted features are transformed into the

feature vector. Finally, the feature vector is fed to the classifier which classifies URLs as spam or non-spam. As a classifier, they have used distributed logistic regression. However, their training dataset is based on 5 popular blacklisting services: Google Safebrowsing, SURBL, URIBL, Anti-Phishing Work Group (APWG), and Phishtank. As blacklisting is a slow process, they have acknowledged the limitation of their dataset.

Antoniades *et al.* reported that short URLs are exploited to promote spam link [APK⁺11]. Tweets usually contain short URLs hence it is difficult for users to identify the URLs' contents without visiting the page. Further, spammers have exploited words related to trending topics along with short URLs in their tweets to generate more traffic to their websites. These spamming activities have adversely affected the user experience on Twitter. To address the issue, Benevenuto *et al.* utilized tweet content and user social behavior in machine learning framework to detect such spamming activities [BMRA10]. Similarly, to identify suspicious URLs redirect chain has also been exploited. As spammers have resource constraint they need to use the same redirect chain hence a significant portion of their redirect chain is found to be similar. Understanding the constraint of spammers, Sangho *et al.* utilized correlated redirected chains of URLs in tweets to detect malicious tweets. They used L2-regularized logistic regression to identify these malicious URLs shared on Twitter. Further, Wang *et al.* exploited click traffic-based information to identify short spam URLs [WNL⁺13].

Besides creating malicious accounts and using URL shortening services, recently social bots have been exploited to post spam tweets. Twitter is increasingly threatened by social bots which are software-controlled accounts that mimic human users with malicious intentions. Social bots automatically produce content and interact with users on social media. Zhang *et al.* demonstrate that social bots are utilized for spam distribution and digital influence manipulation on Twitter [ZZZY16]. An increasing number of such social bots in Twitter has adversely affected the user experience of legitimate users.

Social bots post tweets about popular and focused topics and follow back the users who follow the bots [MSOB13]. Using such a simple strategy, bots could get high influence on Twitter and may pollute timelines of users. These spam bots can be detected by social honeypot traps [LCW10] and also based on features derived from temporal behavior, tweet content, and user profile [CGWJ10, CGWJ12, FVD⁺14]. Malicious accounts associated with bots are detected by identifying synchronized group activities of suspicious accounts [BXG⁺13, CYYP14]. However, all Twitter users are not equally vulnerable to social bots, and the behaviors that make users more susceptible to social bots have been studied in [TGSP11].

Bots accounts and fake accounts typically exhibit highly anomalous behavior which is relatively easy to detect. Hence, attackers compromise and make use of legitimate accounts to promote their content. Compromising legitimate accounts is very effective, as attackers can leverage the trust relationships that the account owners have established in the past. It is reported that compromised accounts are heavily exploited on Twitter to promote malicious content [GTPZ10, GHW⁺10]. Such compromised account shows a drastic change in behavior in a short interval of time. Such compromised accounts are detected using statistical model and anomaly detection technique [ESKV13].

Furthermore, crowdsourcing platforms are also exploited by spammers to generate and propagate malicious campaigns and rumors. Crowdsourcing platforms such as Mechanical Turk, Freelancer, and Innocentive have created open platforms to assign work to the people who are willing to perform them for various levels of compensation. Study of the crowdsourcing sites reveals that about 90% of the tasks are for crowdturfing [WWZ⁺12]. Such accounts are involved in manipulating URL, search keyword, and post to skew the popularity of the content.

Figure 2.3 depicts the detailed of crowdturfing campaign process. A customer pays the crowdturfing site to carry out a campaign. A campaign consists of a collection of small tasks which are accepted by workers for financial benefit. Workers use their fake accounts to perform the malicious tasks. Increasing the number of followers of a

Figure 2.3: Crowdturfing process [WWZZ14]

customer's account by following the account and retweeting the spam message or URLs are two major crowdturfing campaigns on a microblog. It is crucial to identify microblog account of customers as well as workers. In order to detect such account, machine learning techniques have been used [WWZZ14]. Song *et al.* proposed a technique to identify whether a tweet got a retweet from a crowdturfing account or normal account [SLK15]. Retweet time distribution, number of unchangeable retweeters, and number of received clicks are found to be crucial attributes to determine tweets mostly retweet by crowdturfing accounts.

Besides crowdsourcing platform, there is also underground markets on Twitter which provides spam-as-service [TGSP11, TGP12]. In the study, the authors reported that 77% of spam accounts are identified by Twitter within the first day of creation, and 92% of spam accounts within the first three days of creation. It is reported that 89% of spam accounts have fewer than ten followers and 17% of spam users exploit

Figure 2.4: System overview of unsupervised social network spam detection system [TGC⁺13]

hashtags to make their tweets visible in search and trending topics [TGSP11]. This is a similar phenomenon as in web spam and blog spam, where trending topics and popular search terms are hijacked for spamming purpose [HKG07, ZSC11]. Further as on the web, spammers have also used link farming technique to promote content on Twitter [GVK⁺12, TGC⁺13].

The Twitter social graph provides information such as followers, followees, and influence of a user which is vital information for spam detection on Twitter. Yang *et al.* generate features utilizing the relationship between nodes and their neighbors [YHG11]. Relationships such as bidirectional link ratio and betweenness centrality are used, as spam nodes usually cannot establish strong relationships with their neighboring nodes [YHG11]. Similarly, features derived from graph-based relations such as shortest paths and minimum cut between spam senders and receiver are also utilized to detect spammers [SLK11].

Tan *et al.* reported that spammers and non-spammers belong to different communities in user graph [TGC⁺13]. They used social graph and user-link graph to identify legitimate users and spammers. Figure 2.4 shows that overview of unsupervised social network spam detection system named as UNIK [TGC⁺13]. UNIK system first identifies legitimate users using a sybil defense based spam detection scheme SD²

algorithm on the social graph. Utilizing the identified list of legitimate users, UNIK generates a whitelist URLs. There might be the cases where spam URL is incorrectly included in the whitelist which can be effectively identified using user-link graph. As shown in the figure, UNIK analyzes the connections of user-link graph based on the whitelist and edge of the user-link graph are trimmed if the edge is due to the whitelisted links. This trimming process isolates legitimate users from the social graph. Once the isolated users are removed the user-link graph mostly consists of spammers. This approach is effective in capturing spammers promoting malicious URLs. UNIK system is also utilized to identify clusters of spammers and their spamming patterns.

Gosh *et al.* analyzed rank of users based on PageRank and followers count [GVK⁺12]. Their study showed that spammers appear in the list of top users based on PageRank and followers count which is mainly because of social capitalists who follow back any random account. To penalize such spammers and users who follow any random users, they have proposed a ranking technique called Collusionrank [GVK⁺12]. They showed that Collusionrank based ranking effectively remove social capitalists and spammers from the list of top users. A few studies on the Twitter social graph show that leveraging graph-based information robust spam detection system can be designed, however, there are limited studies on spam detection on Twitter using social graph because of difficulties in obtaining Twitter social graph data.

Furthermore, existing social spam detection studies are focused mainly on the identification of spam accounts. However, removing these spam users cannot filter every spam message as spammer may create another account and restart spamming activity. In order to address the issue, tweet-level spam detection has been studied [SML⁺13, MRA13, WZLP15]. Inspired from e-mail content-based technique, [SML⁺13] utilized standard classifiers to detect spam tweets. Language modeling approach has been used to compute the divergence of the trending topic, suspicious message and title of the page linked in the tweet. Language model based features are reported to be effective for spam tweet detection [MRA13]. Similarly, the credibility of tweets on trending topics is estimated based on tweet content, user profile, topic, and propagation

based features [CMP11]. However, there are a limited number of studies on tweet-level spam detection.

2.2 Twitter Dataset Analysis

Twitter is one of the well-studied social media platforms due to its popularity and publicly available data collection APIs. Analyses of the datasets have been conducted to understand different aspects of Twitter which are useful for applications such as spam detection, hashtag recommendation, retweet prediction, event detection to name a few. In this section, we focus on the analyses that are relevant in the context of spam detection such as analysis of hashtag, tweet content, user behavior, and social connection. Studies on analysis of crowdturfing and social bots are also reviewed.

Study of hashtag usage pattern of spammer and legitimate users shows that spammers use more hashtags compare to legitimate users probably to make the tweets visible in search result [YRSB10]. It is reported that spammers heavily exploit hashtags and trending topics hence a significant portion of trending hashtags could be due to the spammers [GTPZ10]. Twitter considered an account as spam if the account frequently uses unrelated trending hashtags. However, spammer posts multiple unrelated hashtags to lure legitimate users to read their tweets. Hence, detailed analysis of hashtags used in the recent posts is important information to identify malicious users on Twitter [Wan10].

Naveed *et al.* analyzed the factors that influence the number of retweets of the tweet [NGKA11]. Their analysis shows that presence of URL, mention, question mark, exclamation mark and positive emoticon are discriminative features to determine the number of retweets of a particular tweet. Further, hashtag, topic, and mentions in user's previous retweet is important features to predict the probability of user retweeting a tweet [PZP⁺11]. However, their analysis shows that time factor is not among the discriminative features. The features that are more likely to produce more retweet are likely to be exploited by spammers to promote their tweets.

Spammers not only post tweets containing unrelated hashtags and manipulated content but also post such tweet multiple times. Hence, near-duplicate detection approach is suitable to identify such users. In order to identify near-duplicate tweets syntactic characteristics, semantic similarity, and contextual information are utilized [TAH⁺13]. The study shows that a small fraction of tweets is exact-duplicate; however, there is a significant portion of the tweets that are near-duplicate tweets with different degrees of overlap. Such near-duplicate tweets are likely to be posted around same time hence there is a less temporal distance among near-duplicate tweets compared to other tweets. Lumezanu *et al.* show that propagandists post significantly more duplicate and near-duplicate [LFK12]. As such spammers want to promote their contents, they are likely to include URLs in their tweets. Study shows that duplicate URL posting behavior is one of the characteristics of spammers [ARY⁺13].

Due to the effective propagation of tweets, Twitter is considered as one of the effective platforms to share links. Java *et al.* reported that 13% of the tweets contains URL [JSFT07]. As Twitter is effective for message propagation spammers aggressively share URLs on Twitter. Hence, Grier *et al.* study spamming activities on Twitter with the primary focus on URL usage [GTPZ10]. Their analysis shows that 8% of the URLs posted on Twitter point to phishing, malware, and scams. Analysis of clickthrough rate shows that click on spam tweet is orders of magnitude more than on spam email. The study also shows that blacklists are too slow and spam detection solely based on blacklist-based approach is not effective as 90% of visitors view a malicious page before the page is blacklisted [GTPZ10].

The 140-character limit of Twitter has encouraged users to share the link on Twitter using short URLs. However, this behavior of legitimate users is heavily exploited by spammers to camouflage their spam URLs. To understand the characteristics of spam and non-spam short URLs on Twitter, click traffic data is analyzed [WNL⁺13]. Behavioral factors of a user posting the URL and user clicking on the URL are analyzed to identify spam links on Twitter [CC15]. The analysis shows that links shared by legitimate users and spammers are significantly different. It is observed that relation of

the post and click timing is a discriminative feature to detect spam URLs. Comarela *et al.* analyzed the factor that affects the response time of the users for content on Twitter and they show that temporal behavior of users is crucial to effectively identify spammers [CCAB12].

Spammers may exploit information about production and consumption behavior of users. Wu *et al.* analyze production and consumption of information on Twitter [WHMW11] and introduced four categories of users namely celebrities, media, organization, and blogs. Their study shows that half of the URLs consumed are generated by a small number of top elite users. Among the four types of the users, media produces the most of the information and they are aggressive in posting content. Hence, spammers may pretend as a media and post content aggressively. They have also observed the homophily within categories where celebrities listen to celebrities, while bloggers listen to bloggers. This shows that a spammer may pretend as a gamer if he wants to promote gaming apps. In order to act as a gamer, the spammers can post content related to gaming and manipulate their social connection as well.

Spammers try to manipulate their social connections to promote their content and evade the spam detection system. To understand social connection based behavior of spammers comparative studies have been done [YRSB10, TLGP14]. Yardi *et al.* show that spammers have more followers and followees than legitimate users [YRSB10]. Followers and followees of the compromised account, fraudulent account, and randomly selected account are also compared and found to have significant difference [TLGP14]. Randomly selected account is likely to have more followers and followees whereas a fraudulent account is likely to have fewer followers and following and this finding is different from finding in [YRSB10]. It is interesting to observe that 50% of the spammers has less than ten followers and following [TLGP14].

Social connection is an important aspect of online social network and usually, influential users have more followers. Hence, spammers try to get more followers to increase the readership and improve the perceived influence. In order to improve the perceived influence, spammers have also utilized link farming technique on Twitter [GVK⁺12].

The analysis shows that majority of spammers' links are farmed from a small fraction of Twitter users who follow back any random user. Their study shows that indegree and outdegree of spammers and randomly selected users are significantly different. Further, profile information and words used in profile description of accounts involved in link farming are found to be significantly different [GVK⁺12]. However, analysis shows that there is no significant difference in the distribution of age of accounts of spammers and legitimate users which reveals that spammers are able to remain undetected in Twitter for a long time [YRSB10].

Haewoon *et al.* analyzed Twitter dataset to understand topological characteristics of Twitter [KLPM10]. They ranked users based on followers count, PageRank and retweet count. It is reported that top list of the users based on followers and PageRank are similar whereas ranked list based on retweet count is different from the two previous rankings. The result shows that influence based on follower count is different from influence based on tweet popularity. Further, they reported that irrespective of the number of followers of initial user any retweeted post reached in an average 1,000 users. In their dataset, they have observed only 22.1% of users have reciprocal relationships [KLPM10]. Java *et al.* reported that information seekers may rarely post any content but follow many people [JSFT07]. It is observed that indegree and outdegree of Twitter social network follow power law distribution as in other social networks [BKM⁺00].

Usually, a reputation of an account on Twitter is measured in terms of a number of followers of the account. Users who want to increase the follower count may buy followers from Twitter follower market which may use fake accounts. Study has been conducted to understand static and dynamic properties of customers of such market [SWE⁺13]. To tackle blacklisting of such fake accounts, underground market utilizes thousand of host located around the globe. Such a spam account behave as a legitimate for a month before they are utilized in the market. It shows that a small number of groups are involved in creating large numbers of such account [TMG⁺13]. Fake accounts as well as compromised accounts have been utilized in such market.

The customers of such market show a dramatic increase in follower in a short time and gradual decrease in follower overtime. It is observed that such customer has more followers than followees. The distribution of follower to followee ratio is significantly different for normal users and customers of such market [SWE⁺13].

Further, crowdsourcing platform has also been utilized by spammers to post malicious tweets. Study shows that crowdturfing accounts can be more popular compared to legitimate accounts [SLK15]. To better understand crowdturfing activities, types of malicious tasks, properties of requesters, and worker in crowdsourcing site have been analyzed [LTC13]. Online social marketing, search engine manipulation, and web traffic are the main focus of such accounts. Analysis shows that majority of workers for such activities are from western countries but workers are distributed across the globe. Similar, pattern is observed for the buyer of such service [LWG14].

Besides manually posting malicious tweets, automated posting of tweets using social bots is also common on Twitter. Bots try to mimic the behaviors of legitimate users to remain undetected from spam detection system. To understand the characteristic of social bots Freitas *et al.* perform detail analysis of social bots on Twitter [FBGV15]. Different aspects such as specified gender in the profile, frequency interaction with other users, a technique to generate tweets and type of user they interact are analyzed. It is reported that 31% of social bots are detected after a period of one month of automated behavior. The analysis shows that using automated posting and interaction, social bots can get a larger number of followers. The study shows that gender and profile picture of social bots are important factors to gain acceptance from other users [FBGV15].

The review of these studies provides a foundation for spam detection task and further analysis of spam dataset. Analyses of this section are utilized in Chapter 4 to compare and contrast our findings with previous findings.

2.3 Spam on Other Communication Media

Spam is a prevalent problem in all types of online media such as email [Cor08], Web [SH12], video [BRA⁺09], comment [MCL05], and review [JL07], to name a few. In different online platforms, spammers exploit different techniques to evade spam detection system. Moreover, spammers keep on changing their spamming behavior and activities to remain undetected from spam detection systems [CKOF09]. Hence, a wide range of information such as content, link, user behavior, as well as HTTP session information has been utilized for spam detection tasks [SH12].

Email spam. Spam email contains unwanted and malicious information such as advertisement, fraud scheme, promotion, or malware [XYW06, BB08, JKA⁺09]. Such email spam has been identified using text-based spam detection techniques [LT04]. However, to elude text-based spam detection system spammers embed spam text into an image. To address such spamming activities, OCR-based recognition has been utilized to detect spam email containing spam text information embedded in an image [FPR06]. Whitelisting and blacklisting of senders, domains or IP addresses have been done to fight email spam aggressively [Cor08]. Further, hashing approach has also been used for spam email detection [JKA⁺09].

Web spam. Web spamming can be in the form of link spamming, cloaking, click spamming and comment spamming. To understand challenges on web spam, detailed study of spam on web has been conducted [HKG07]. Important features identified from detailed analysis are utilized in standard machine learning algorithms to design a robust web spam detection system [MCL05, NNMF06]. Similarly, web graph has been exploited to effectively combat web spam [GGMP04]. Spammers keep on changing their techniques to remain undetected from spam detection systems. To tackle the changing spamming activities ensemble approach has been proposed [CKOF09]. As in email spam detection system, whitelisting and blacklisting of senders, domains or IP addresses have also been used to address spamming issues on web [SH12].

Product review spam. Online product reviews provide opinions of other users about products and services to potential customers. As product review is an important piece of the information referred by potential customers, spammers are attracted to promote their products and defame competitors' products. User behavior as well as content information has been utilized to detect spam reviews [JL07, LNJ⁺10, MLW⁺11]. Unsupervised approach has also been used to identify review spam [XZ15b, XZ15a]. Likewise, graph-based approach has also been used for review spam detection [XZCL13]. Topic model based approach has also been used for spam product review detection [LCL13]. Spam review detection techniques have also utilized features derived from part-of-speech tag, n-gram, and sentiment of the reviews [LHYZ11, OCCH11]. As spammers try to promote their products by posting similar content for multiple times, near-duplicate detection techniques have been exploited to identify spammy content and spammers [Bro97, CFGM02, WJL⁺07]. Duplicate detection and classification techniques have also been used for spam review detection [JL07, JL08].

2.4 Hashtag Recommendation

In this dissertation, we use hashtag recommendation as a case study to analyze the impact of spam on hashtag recommendation. In this context, we also review studies on hashtag recommendation.

It is important to understand why people tag any online resources. Studies have been done to understand the motivation of user to annotate online resources [AN07, NNY08, NY10]. They have reported that broadly social and functional dimensions motivate people to tag online content. The social dimension is categorized into self, family & friends, and the general public. Similarly, the functional dimension is further categorized to organization and communication [AN07]. Cunha *et al.* reported that men and women have distinct hashtag usage behavior [CMA⁺12]. Such a detailed understanding of the motivation of users for tagging is crucial for designing a robust hashtag recommendation system.

Hashtag recommendation is a well-studied area in the context of platforms such as Flickr, Delicious [WYYH09, TSD08]. Users on Flickr and Delicious tag content with a large number of hashtags hence for these platforms co-occurrence is an important information for tag recommendation task. Co-occurrence based hashtag recommendation systems are reported to be effective in these platforms [SvZ08, WYYH09]. Similarly, topic modeling based approach is also used for hashtag recommendation task on Flickr and Delicious [IYU09]. However, due to the 140-character limitation on Twitter, hashtag co-occurrence and hashtag per tweet are significantly different from these platforms. Hence, hashtag recommendation methods suitable for Flickr or other online resources can not be directly adopted on Twitter.

Recently, hashtag recommendation on Twitter has attracted the attention of research community. Zangerle *et al.* analyze the effect of different similarity measures on hashtag recommendation task for micro-blog message [ZGS13]. The study shows that cosine similarity with TF-IDF weighted entries is the most appropriate measure to retrieve similar micro-blog posts. Kywe *et al.* utilize TF-IDF weighted similarity measure for personalized hashtag recommendation task [KHLZ12]. They exploit user preferences and tweet content for hashtag recommendation. Personalized hashtags are recommended by combining hashtags obtained from similar users and similar tweets and rank them by the frequency of occurrence.

Topic model has been used for recommending general hashtags related to the topic of a tweet [GSDN⁺13, SC14]. General hashtags are recommended to a tweet based on the topic distribution evaluated using Latent Dirichlet Allocation (LDA). Further, supervised topic model based approach is proposed to cover the relationship among words, hashtags, and topics of tweets [SC14]. Ding *et al.* also utilized LDA-based model to recommend hashtags [DQZH13]. They combine the LDA model with a translation model, to address the vocabulary gap between tweets and hashtags. PLSA-based topic model has also been used to model the hashtag annotation behavior in Twitter [MSYC14].

Model inspired from word-alignment model in language translation task [BPPM93] has been utilized for hashtag recommendation. Statistical machine translation techniques are exploited for hashtag recommendation on Twitter [DQZH13,LCS11]. It is based on the assumption that both content and hashtags are about the same themes but written in different languages. Zhang *et al.* incorporated temporal and personal attributes in the translation based model. They also employ hashtag extraction technique to recommend the hashtags that are not present in training dataset [ZGSH14]. Inspired by the language translation model we have also used named entities in our case study. We also consider that hashtag and named entities are about the same theme but written in different languages. We have also used the language translation inspired technique to generate candidate hashtag.

Learning to rank technique has been used to model hashtag relevance. In this approach, candidate hashtags are selected using different methods and they are re-ranked using learning to rank technique. Recently, learning to rank approach is used to recommend hashtags on real-time for news articles shared on Twitter [SIH16]. Further, temporal information is also incorporated in learning to rank framework to design hashtag recommendation system [GYK15].

Neural network based approach has also been utilized for hashtag recommendation task [TGV⁺14,DWP⁺15,GZ16]. Word2vect based word embedding has also been utilized for hashtag recommendation on Twitter [TGV⁺14]. They have used Skip-gram model to learn the vector representation of words. In their approach, feature engineering is not required as an unsupervised approach is utilized for feature extraction.

Similarly, Convolutional neural network(CNN) is utilized for hashtag recommendation task in [WWY14]. The CNN takes d-dimensional vector representation of each word in a document using word lookup table. A document having l words is represented by $l \times d$ input matrix. They use a sliding window of size $K = 5$ and also consider the padding vector at both ends. After convolution, they utilized *tanh* nonlinearity followed by max pooling operation. Finally, another *tanh* non-linearity

followed by a fully connected linear layer is applied to represent the entire document in the original embedding space of d -dimensions. Similarly, a hashtag is also represented in d -dimensional space using a lookup table. A score of the hashtag and document is computed based on the cosine similarity between the document vector and hashtag vector. Hashtags having top scores for a document are the hashtags recommended for the document.

Recently, motivated by the word trigger based model, Gong *et al.* have proposed attention based CNN model [GZ16]. The model consists of two channels namely, local attention channel and global channel. The global channel utilizes all the words for each tag whereas local attention channel focus only on a small subset of words for each hashtag. Trigger words are selected based on the attention layer. Convolution layer combined local attention channel and global channel. Finally, fully-connected layer with softmax output is utilized to formulate the hashtag recommendation task as a multi-class classification problem. Results show the superior performance of the attention-based model for hashtag recommendation task.

2.5 Summary

This chapter reviewed previous studies that provide a foundation for our spam detection and spam analysis work. We discussed previous studies about spam detection on Twitter and other communication media. We also reviewed the spam dataset analysis and hashtag recommendation studies. From here onward, each chapter focuses on our work on spam analysis and detection.

Chapter 3

HSpam14 Dataset Creation

In this chapter, we discuss data collection and statistics of the dataset and present a detailed discussion of the tweet annotation technique. We propose an efficient approach to create a dataset for spam detection task on micro-blogging platform. Without loss of generality, we discuss dataset labeling technique in the context of Twitter. To the best of our knowledge, there is no any publicly available Twitter dataset for tweet-level spam detection task. This is one of the reasons there are limited studies on content-based spam detection on Twitter. Further, trending hashtag on Twitter are heavily exploited to promote the content, so we used trending hashtags to create hashtag oriented dataset. We labeled 14 million tweets with the proposed approach and made the dataset publicly available which is named as HSpam14. As the dataset is publicly available, researchers working on spam detection on Twitter can utilize the dataset to design spam detection system and they can also analyze the dataset to understand better about spamming activities on Twitter.

3.1 Data Collection and Statistics

3.1.1 Data Crawling

To collect data for hashtag-oriented spam tweet research, a key issue is to identify candidate hashtags. The tweets containing these candidate hashtags are then collected and labeled. Without the luxury of accessing all tweets or all hashtags, we rely on the trending topics (*i.e.*, hashtags) reported on Hashtags.org.¹ Hashtags.org reports trending hashtags in three categories: *trending up*, *trending down*, and *most popular hashtags*. As expected, we observe that the hashtags in *most popular hashtags* category do not change much for many days. To cover more varieties of hashtags in our data collection, we used the hashtags in *trending up* and *trending down* categories as query keywords to search for tweets on the daily basis. On each day, we collect the trending hashtags in these two categories and we then use each collected hashtag as a query keyword to collect tweets using Twitter’s streaming API for that day.² A tweet is collected through the API if it contains the query keyword as a *hashtag*, *word*, or *link* in its content. On average 135 hashtags were used as query keywords on each day. The data collection process last for two months, May and June 2013.

In total, we collected 24.36 million tweets, which were published by 11.97 million users. The collected tweets contain 20.21 million hashtags and 6.97 million hyperlinks. After resolving the short URLs, there are 3.43 million distinct URLs. Ignoring the deadlinks and the links that require authentication, 2.05 million web pages were successfully downloaded. However, we have not fully utilized these web pages for this labeling process.

Among the collected tweets, 14.07 million tweets are in English.³ In our following discussion, we mainly focus on the 14.07 million (or simply 14 million in approximation)

¹<https://www.hashtags.org/trending-on-twitter/> Note that the trending hashtags identified by Hashtags.org are based on the site’s proprietary algorithm.

²We used Twitter database server code available at <http://140dev.com/>.

³We used language detection library for Java available at <http://code.google.com/p/language-detection/>

Figure 3.1: User frequency distribution

tweets in English. The same set of tweets will be annotated with spam and ham labels in the next section.

3.1.2 Data Statistics

User Distribution. The 14 million English tweets were posted by 7.25 million users and the distribution of the number of tweets per user is plotted in Figure 3.1. Shown in the figure, the data demonstrates a typical power-law distribution that is widely observable in many social-related studies. While more than 4.83 million users each contributed only one tweet, the most active user contributed 754 tweets to this collection. Recall that this collection covers two-month period and the tweets were collected by using trending hashtags as queries (*i.e.*, biased sampling). The few extremely active users may post a larger number of tweets regularly or post many tweets targeting on some trending hashtags. Because we do not have the full profiles of the users in this data collection, the underlying reasons cannot be verified. In our following analysis, we mainly focus on the content of the tweets, particularly the usages of hashtags and mentions.

Figure 3.2: Hashtag/mention frequency distribution

Usages of Hashtags and Mentions. Figure 3.2 plots the frequency distributions of hashtags and mentions. This figure shows that the usages of both hashtags and mentions follow power-law distributions with slightly different slopes in the log-log plot. On the one hand, this figure suggests that hashtags are more commonly used than mentions in tweets, particularly the extremely popular ones. On the other hand, a small set of usernames is frequently being mentioned in tweets. A mention in a tweet is also a form of hyperlink, and clicking a mention leads to the user profile page where much more information about this user is displayed (*e.g.*, user description and recent tweets). For this reason, mention can be intentionally used for spamming purposes, similar to hashtags⁴.

Table 3.1 lists the top-20 most popular hashtags in the 14 million tweets. Observe that the most popular hashtag `#TEAMFOLLOWBACK` appears in more than half million tweets out of the 14 million. Moreover, 40% of the top-20 most popular hashtags contain the word “follow”. Many of the usernames among the popular mentions contain the word “follow”. In fact, a quick check of the top-20 most mentioned usernames

⁴<https://blog.twitter.com/2012/shutting-down-spammers>

Figure 3.3: No. of hashtags/mentions per tweet

shows that 11 usernames⁵ have been suspended by Twitter. Among the suspended usernames, nearly half contain the word “follow”.

Figure 3.3 plots the per-tweet distribution of hashtags and mentions. Note that the y-axis is in log-scale. Both distributions show the same interesting pattern: (i) the number of tweets having 1 to 4 hashtags/mentions drops significantly along the increase of the number of hashtags/mentions; (ii) the rate of dropping becomes much smaller for tweets containing 4 to 10 hashtags/mentions; and (iii) the number of tweets again drop significantly with the increase of contained hashtags/mentions from 10 onwards. As expected, very few tweets contain more than 14 mentions or more than 18 hashtags, given the 140-character length limit of tweets.

3.2 Tweet Annotation

There are three major challenges in manually labeling millions of tweets: (i) *large number of instances*, *i.e.*, it is infeasible to manually check every single tweet to determine its label as in many other labeling efforts [CTH⁺09], (ii) *limited amount of*

⁵In Twitter, a user may change her username at any time. Mentioning the same username at different time points may refer to different users.

Table 3.1: The top-20 most popular hashtags. All these 20 hashtags appeared as trending up/down hashtags which were used for sampling the tweets.

Hashtag (#)	Freq.
TEAMFOLLOWBACK	666,328
TFBJP	527,176
gameinsight	510,504
android	341,240
OPENFOLLOW	332,857
FF	293,748
androidgames	286,706
RETWEET	250,992
RT	235,1373
IPADGAMES	232,335
SougoFollow	197,525
ipad	195,375
FOLLOWBACK	177,109
THF	165,762
FOLLOWNGAIN	149,916
500aday	146,005
AUTOFOLLOW	141,214
MUSTFOLLOW	138,040
TEAMHITFOLLOW	136,043
MUSIC	129,852

information carried by each tweet because of its shortness, and (iii) *lack of standard criteria* for the classification of spam and ham tweets.

To address the first two challenges, we propose to group the near-duplicate tweets into clusters. Spammers keep changing their behavioral patterns to avoid detection; however, they keep on posting similar promotional tweets exploiting popular keywords. Hence, clustering of near-duplicate tweets captures promotional tweets effectively despite the change in behavioral patterns of spammers. Because the tweets in each near-duplicate cluster are nearly identical to each other, the same label applies to all the tweets in the same cluster. The near-duplicate clustering also helps in partially addressing the second challenge. One of the major characteristics of spammers is to post near-duplicate content in a repetitive manner. If a larger number of near-duplicate tweets are from very few users, then it is relatively easier for us to label spam tweets

Figure 3.4: Overview of the four major steps in labeling spam/ham tweets

even if each tweet contain very limited semantics. During the labeling process, we may also access all tweets by a particular user in our data collection to get more contextual information. Regarding the last challenge, we reference to the existing studies on spam detection (*e.g.*, tweets about quick money gain, adult content are considered spammy). In the context of hashtag-oriented spam detection, we also consider tweets containing semantically unrelated hashtags and quick followership gain as spam tweets.

Figure 3.4 illustrates the four main steps in the annotation process. We now brief the four major steps. More details of the steps and the labeling criteria are described in the next sub-sections.

Heuristics-based Tweet Selection. It is reasonable to assume that majority of tweets are ham. To minimize human labeling effort, it is necessary to label only the tweets that are more likely to spam. To this end, we selected the tweets that contain most popular hashtags (our assumption is that these hashtags are likely to be hijacked for spamming). We also select the tweets that contain money gain, adult content related keywords, and promoting purposes.

Cluster-based Manual Annotation. Selected tweets are grouped as clusters of near-duplicate tweets. A subset of clusters are then labeled with spam or ham, and all the tweets in the same cluster share the same label. The near-duplicate clusters are obtained by *MinHash* algorithm [Bro97]. With an initial set of clusters labeled, we adopt nearest neighbor approach to “grow” the labels as in [CTH⁺09]. A cluster is

labeled by kNN if the prediction of the label is very confident (based on the percentage of the nearest neighbors in the same class), and manually labeled otherwise. Spam tweets may be posted by some spam user accounts. For the top 10,000 users who posted the largest number of tweets in our dataset, we conduct user-based tweet clustering (*i.e.*, all tweets in each cluster are posted by the same user) and label the clusters. Similarly, based on the domain names of the links embedded in tweets, we cluster the tweets that contain links from the same domain, and label the clusters. The top 10,000 most frequent domains in our dataset are considered.

Reliable Ham Tweet Prediction. By the assumption that majority tweets are ham, a large number of tweets, particularly the unselected tweets left by the heuristics based selection, remain unlabeled. Here, we adopt the approach in *learning from positive examples only* in classification [LDL⁺03] to determine reliable ham tweets.

Label Prediction by EM Algorithm. After getting a reasonably large number of reliable labels of both spam and ham tweets, the EM algorithm by Nigam *et al.* [NMTM00] for learning from labeled and unlabeled data is adopted to predict the labels of the remaining tweets.

3.2.1 Heuristics-based Tweet Selection

We heuristically target on the tweets that are most likely to be spam, with the assumption that most tweets are ham. More specifically, we select three sets of tweets: (i) tweets containing any of the top-100 most popular hashtags in our collection, (ii) tweets containing adult content related keywords, and (iii) tweets containing keywords related to quick money gain, lucky draw, free gift *etc.*.

The first set of tweets is selected based on the heuristic that spammers are likely to exploit popular hashtags to make their content visible to a broader audience. Table 3.1 lists the top-20 most popular hashtags, which is a subset of the 100 selected hashtags. The second set of the selected tweets aligns with most other spam detection tasks in email and web domains. A list of keywords related to adult content is used for this

purpose.⁶ For the third set of tweets selection, we used the following 23 keywords to capture the tweets related to quick money gain, lucky draw, and free gift: **awesome, business, hurray, earn, bargain, deal, adventure, money, click, facebook, want, incredible, gift, money, amazon, ebay, amazing, prize, lucky, check out, shopping, sale, giveaway**. Note that, there are tweets repeatedly posted by users with similar content, for promoting purposes related to major websites like Facebook, Amazon, and eBay. Therefore, these website names are included as keywords.

As the result, 7.10 million tweets posted by 3.93 million users were selected by keyword match. A tweet is selected if the tweet contains any of the selected hashtags or keywords in its content.

3.2.2 Cluster-based Manual Annotation

3.2.2.1 Near-Duplicate Clustering

To label the tweets more effectively, we group the near-duplicate tweets into clusters using MinHash algorithm [Bro97]. The preprocessing of the tweets includes removal of links, removal of mentions (*i.e.*, @username), removal of the ‘#’ symbols but keep the words in the hashtags, and removal of stop-words using the stop-word list provided in Lucene⁷.

Both MinHash and SimHash are widely used for detecting near-duplicate documents [SL14]. Here, we adopt the MinHash algorithm on tweet’s shinglings for duplicate detection [MRS08]. We consider two tweets t_1 and t_2 are (near-)duplicates if minimum hash values of their *uni-gram*, *bi-gram*, and *tri-gram* representations are the same. More specifically, let $h(\cdot)$ be a hash function, and let tweet t be a n -word sequence $\langle w_1, w_2, \dots, w_n \rangle$. The minimum hash value of uni-gram representation of t is $h(w_i)$ iff $h(w_i) \leq h(w_j)$ for any $1 \leq i, j \leq n$ and $i \neq j$. The minimum hash value of bi-gram representation of t is $h(w_i w_{i+1})$ iff $h(w_i w_{i+1}) \leq h(w_j w_{j+1})$ for any

⁶We used the stem version of the keywords under pornographic category listed at <http://nymphs.org/RevealDirtyWordList.txt>. In total there are 80 keywords in the list, and after stemming, there are 65 word stems.

⁷<http://lucene.apache.org/core/>

Figure 3.5: Size distribution of the near-duplicate clusters

$1 \leq i, j \leq n - 1$ and $i \neq j$. The minimum hash value of tri-gram representation is computed in a similar manner. For simplicity, let $h_1(t)$, $h_2(t)$, and $h_3(t)$ be the minimum hash values of uni-gram, bi-gram, and tri-gram representations of tweet t . Then t_1 and t_2 are near-duplicates iff $h_1(t_1) = h_1(t_2)$, $h_2(t_1) = h_2(t_2)$ and $h_3(t_1) = h_3(t_2)$. In our Java implementation, we used the `hashCode()` method in `String` class to get the hash value of a n-gram (*i.e.*, a word or a shingling). The clustering process is computationally effective, and can be achieved in one scan of the tweets.

Result and Evaluation. Because our purpose is to label tweets more effectively, the clusters containing fewer than 10 tweets are ignored from the manual annotation. The remaining 40,549 clusters contain 3.13 million tweets posted by 1.52 million users. Figure 3.5 plots the size distribution of these 40,549 clusters. Again, the plot demonstrates a power-law distribution with many clusters each contains 10 or slightly more tweets and very few extremely large clusters. The largest cluster contains 127,559 near-duplicate tweets, mainly due to hashtags `#gameinsight`, `#android`, and `#androidgames`. An example tweet is “RT @username: I’ve collected 103,501 gold coins! [LINK1](#) `#android` `#androidgames` `#gameinsight`”.⁸

⁸We mask the usernames and links in all example tweets.

Table 3.2: Example tweets from two near-duplicate clusters

-
1.
 - RT @username: #Money SEC chief White remains mum on money market fund reforms [LINK1](#) #TonyRocha
 - #Forex: SEC chief White remains mum on money market fund reforms [LINK1](#) [LINK2](#)
 - UPDATE 1-U.S. SEC chief White remains mum on money market fund rules [LINK1](#) #news
 - SEC Chief White Remains Mum On Money Market Fund Rules: The top securities regulator remained tight-lipped on ... [LINK1](#)
 2.
 - RT @username: [Retweet Only If You Want New Follower] #TEAMFAIRYROSE #OpenFollow #Follow #TFBJP #500aday #RT #R.Family #MaxVIP
 - RT @username: [Retweet Only If You Want New Followers] #TEAMFAIRYROSE #FOLLOWNGAIN #Follow #TFBJP #500aday #RT #R.Family #THF
-

Table 3.3: Measures of intra-/inter-cluster samples of tweets

Measure	Intra-cluster	Inter-cluster
Jaccard coefficient	0.9169 ± 0.0026	0.0187 ± 0.0006
Cosine similarity	0.9510 ± 0.0017	0.0350 ± 0.0012
Length difference	0.3933 ± 0.0196	5.5895 ± 0.0673

In manual annotation, all the tweets in the same near-duplicate cluster will be annotated with the same label (*i.e.*, spam or ham). For this purpose, we must ensure that the tweets in the same cluster are indeed similar to each other, or in other words, are near-duplicate. Table 3.2 shows two randomly selected near-duplicate clusters containing 4 and 2 example tweets respectively, from each cluster. The examples show that tweets in the same clusters are indeed very similar to each other. However, there is a need to quantitatively evaluate the quality of the clusters in terms of tweet similarity. To this end, we randomly selected 4000 clusters (*i.e.*, about 10% all clusters) and measured the intra-cluster and inter-cluster tweet similarities. For intra-cluster tweet similarity, two tweets are randomly selected from each cluster; for inter-cluster tweet similarity, two tweets are randomly selected from two different clusters without replacement.

Table 3.3 reports tweet similarity in terms of Jaccard coefficient and cosine similarity. Cosine similarity is computed based on the bag-of-words model with *TFIDF* weighting. The length differences between the randomly selected pairs of tweets in number of words are also reported. Shown in Table 3.3, tweets from the same cluster show high similarity in terms of both Jaccard coefficient and cosine similarity with values above 0.91 and 0.95 respectively. On the other hand, inter-cluster similarity of tweets is dramatically lower at 0.019 and 0.035 respectively for Jaccard coefficient and cosine similarity. In terms of length difference, tweets from the same cluster are nearly identical with the length difference of 0.39 word, while a pair of tweets from different clusters differ on average 5.59 words.

In summary, the clustering process is able to produce clusters with high intra-cluster similarity and low inter-cluster similarity. Tweets in the same cluster are of very similar length.

3.2.2.2 Manual Annotation of Near-Duplicate Clusters

Labeling tweets to be spam or ham is non-trivial for two reasons. First, there is no very specific definition of “spam”. However, many definitions, including the description in Wikipedia⁹, contain the phrase “unsolicited message”, and the words “advertisement” and “repeatedly”. Second, because of the length limit, tweets are very short, and often do not come with context. The notion of near-duplicate clusters partially addresses these two challenges. Each near-duplicate cluster lists the near-“repeated” tweets and it is easy to identify whether the repeated tweets are from very few users or from some specific apps. Because of the grouping of similar tweets, it is also relatively easy to identify whether these similar tweets were posted because of some important events/topics (*e.g.*, the 4 tweets in the first cluster in Table 3.2) or were posted for advertisement purposes (*e.g.*, the 2 tweets in the second cluster in Table 3.2). In other words, we are able to infer the context of the tweets in a collective manner. By considering the tweets in each near-duplicate cluster, the users who posted these

⁹<http://en.wikipedia.org/wiki/Spamming>

tweets, and also the usage of hashtags and links, we are able to label the near-duplicate clusters.

We label *tweets in a duplicate-cluster to be spam* if the tweets in this cluster satisfy *one of the five conditions*:

1. For quick money gain or quick gain of followers
2. Repeatedly posted for advertisement, promoting products/services, and request for retweet or offering benefits
3. About adult content
4. Contain many unrelated but popular hashtags
5. Automatically posted by bots/apps for multiple times with nearly identical content

Note that, many bots/apps post legitimate tweets. For example, by linking Twitter and Foursquare accounts, users have the option to share check-ins made in Foursquare to Twitter. Such tweets will not be labeled as spam because these tweets are informative (*i.e.*, reporting location information of users). On the other hand, recall that in Section 3.2.2.1, we report that the largest cluster contains 127,559 near-duplicate tweets in the form of “RT @username: I’ve collected 103,501 gold coins! [LINK1](#) #android #androidgames #gameinsight”. The only difference between any two tweets is the number of gold coins and the links. Moreover, we observe that many users post only tweets in this form in his/her full user profile in Twitter. We consider such tweets less informative and label them as spam.

With the above criteria, we *selectively* label the near-duplicate clusters, because labeling more than 40 thousand near-duplicate clusters is infeasible. The labeling is conducted in two phases. In the first phase, the top-1000 largest clusters ranked by number of tweets in each cluster are labeled. Because of the power-law distribution (see Figure 3.5), the labeling of the 1000 largest clusters cover a large number of tweets.

Table 3.4: Initial manual annotation of near-duplicate clusters

Label	No. of clusters	No. of tweets	No. of users
Spam	1,104	1.63 million	0.61 million
Ham	1,000	0.19 million	0.17 million

In the second phase, we aim to cover more diverse tweets by considering the keywords we used in Section 3.2.1. For each of the keywords used for tweet selection, we label around 10 largest clusters containing this keyword if the keyword is not covered in the top-1000 clusters labeled earlier. As examples, most clusters containing keywords “love” and “news” are ham clusters while clusters containing “gameinsight” and “iphonegame” are mostly spam. In the labeling process, we also encounter some cases where we cannot confidently label the tweets to be ham or spam even after manual inspection. In this case, we assign an “unknown” label to this cluster.

Result and Analysis. After the two-phase labeling, 1.82 million tweets in 2,104 near-duplicate clusters are manually labeled, reported in Table 3.4. Observe that, although the number of spam clusters is similar as the number of ham clusters, the number of spam tweets contained in these clusters is significantly larger than the number of ham tweets. This is expected as spam tweets are mostly posted repeatedly. More specifically, the median number of tweets in the labeled spam clusters is 437 while the median number of tweets in ham clusters is 26. The median number of users in the two sets of clusters are 383 and 25 respectively. It is also observed that 84% of labeled spam clusters have at least one user posting the same tweet multiple times. On the other hand, only 25% of labeled ham clusters have at least one user posting the same tweet multiple times. More interestingly, 47,454 users who posted at least one spam tweet have 0 followers. Without any followers, these users do not pollute the timelines of other users. However, their tweets may have adverse effect on keyword-based or hashtag-based tweet search. The number of users who posted at least one ham tweet and have 0 followers is 4,822, about 10% of those who posted at least one spam tweet.

3.2.2.3 k NN Annotation

Using the 2,104 manually labeled clusters as ground truth, we are able to “grow” labels in a more effective manner with the help of k NN classifier. A similar approach has been adopted in creating benchmark dataset for a web image dataset [CTH⁺09].

The most important issue here is to ensure the quality of labels predicted by the k NN classifier. We address this issue by assigning a label to a cluster only if 80% of its k -nearest neighbors are from the same class (*i.e.*, spam or ham). Moreover, if this newly labeled cluster c_k causes misclassification of a manually labeled cluster c_m if the same k NN rule were applied to c_m , then we consider this newly labeled cluster c_k to be a *difficult cluster*. The *difficult clusters* are then manually inspected and labeled. If a newly labeled cluster does not cause misclassification of any existing manually labeled cluster, then the label of this cluster assigned by k NN is considered as a *confident* label. The labeling process is conducted in batch mode and thus confidently labeled clusters in each batch are added as labeled examples for the next batch.

Algorithm 1 summarizes the main steps in k NN-based near duplicate cluster labeling. The algorithm takes the set of manually labeled cluster C_ℓ and unlabeled clusters C_u as input, and returns clusters with confident labels C_k as output, in four main steps.

Step 1. A subset of clusters $C_{mk} \subset C_\ell$ is obtained such that within C_{mk} each cluster’s label is consistent with the label predicted based on the k NN rule (Lines 2 - 7). More specifically, for each cluster c , its k nearest neighbors are obtained from the labeled set C_ℓ using function $C_{NN} = kNN(c, k, C_\ell)$, where C_{NN} denotes the set of c ’s nearest neighbors, $kNN(c, k, C_\ell)$ finds the k nearest neighbors of c within the set C_ℓ . In Line 4, the function $majorityLabel(C_{NN})$ returns the *number of neighbors* from the majority class in C_{NN} (*e.g.*, either spam or ham). The clusters in C_{mk} will be used to identify the difficult clusters labeled by k NN in later steps.

Step 2. The algorithm classifies unlabeled clusters in a batch mode (Lines 11 - 17). Each batch is set to be 20% of the size of the manually labeled examples (Line 12).

Algorithm 1: k NN Annotation

Input : Set of manually labeled clusters C_ℓ
 Set unlabeled near-duplicate clusters C_u
 Number of nearest neighbors k
Output : Set of labeled near-duplicate clusters C_k

```

1  $C_{mk} = \{\}$  // clusters consistent with  $k$ NN result
2 foreach cluster  $c \in C_\ell$  do
3    $C_{NN} = kNN(c, k, C_\ell)$  // find nearest neighbors
4   if  $majorityLabel(C_{NN}) \geq 0.8 \times k$  then
5      $C_{mk} = C_{mk} \cup \{c\}$ 
6   end
7 end
8  $C_d = \{\}$ 
9  $C_t = C_\ell$  // set of training examples
10  $C_b = \{\}$  // a batch of labeled clusters
11 for unlabeled cluster  $c \in C_u$  do
12   if  $|C_b| < 0.2 \times |C_\ell|$  then
13      $C_{NN} = kNN(c, k, C_t)$ 
14     if  $majorityLabel(C_{NN}) \geq 0.8 \times k$  then
15        $C_b = C_b \cup \{c\}$ 
16     end
17   end
18   else // verify the current batch
19     foreach cluster  $c \in C_{mk}$  do
20        $C_{NN} = kNN(c, k, C_t \cup C_b)$ 
21       if  $majorityLabel(C_{NN}) < 0.8 \times k$  then
22         foreach neighbor cluster  $c_n \in C_{NN}$  do
23           if  $c_n \in C_b$  then
24              $C_b = C_b - \{c_n\}$ 
25              $C_d = C_d \cup \{c_n\}$ 
26           end
27         end
28       end
29     end
30      $C_t = C_t \cup C_b$  // add as training examples
31      $C_b = \{\}$  // start a new batch
32   end
33 end
34 Manually label clusters in  $C_d$  and add to  $C_t$  in Line 9
35 Repeat the labeling process (Lines 11 - 34) till no more clusters can be labeled
36 Return  $C_k = C_t - C_\ell$ ;

```

For the first batch, the set of training examples is initialized to be the set of manually labeled clusters C_ℓ (Line 9). In the subsequent batches, the training examples will include the newly labeled examples (Lines 30 - 31).

Step 3. The algorithm verifies whether the newly labeled examples cause misclassification of some clusters in C_{mk} (Lines 19 - 29). More specifically, the k nearest neighbors of each cluster in C_{mk} is now searched from the set of training examples C_t and the newly labeled batch C_b using the function $kNN(c, k, C_t \cup C_b)$. If the number of neighbors from majority class is smaller than 80% of k , then we consider that the neighbors which are from the newly labeled batch cause the misclassification. These clusters will then be removed from the batch and considered as *difficult clusters*. The difficult clusters are then manually labeled in the next step. The remaining confidently labeled clusters in C_b will be added to C_t for the next batch labeling (Line 30).

Step 4. Observe that the predicted labels of unlabeled clusters in C_u are dependent on the order of labeling because the training examples C_t are updated after each batch of labeling. For this reason, we run the labeling process (Lines 11 - 33) multiple times. Before each run, we manually label the difficult clusters resulted from the last run (Line 34) and add them to C_t in Line 9 because these labels are manually assigned. Note that, for concise presentation, we do not explicitly add this “outer loop” in Algorithm 1. Instead, this step is stated in Line 35 and the iteration continues until there is no unlabeled cluster which has 80% of neighbors from the same class. Note that some detailed control structures in the algorithm are ignored, for example, the last batch in each run may not reach the size of $0.2 \times |C_\ell|$.

For nearest neighbor search, the similarity between two clusters is the cosine similarity with *TFIDF* weighting of the word feature vectors for the two clusters, by aggregating all the tweets in each cluster. The value of k is set to 19 empirically in our kNN based annotation process. The size of each batch is about 420.

Annotation Result. As the result of kNN -based annotation, 15,571 spam clusters and 15,913 ham clusters are labeled. In the labeling process, 1,178 clusters were

identified to be difficult clusters and were then manually labeled. Among the latter, 826 clusters were labeled ham and 275 were labeled spam clusters; the remaining 77 clusters were difficult to label even after manual inspection. These 77 clusters are labeled “unknown” in our data collection.

After manual and k NN-based annotation, we have labeled more spam tweets than ham tweets. Because spam clusters often contain much repeated tweets, the number of spam tweets is significantly larger than the number of ham tweets. The reason is that, the manual annotation process was targeted on spam tweets by using the most suspicious keywords and most popular hashtags. The k NN-based annotation was guided by the labeled clusters. In other words, most ham tweets are left untouched in the dataset.

3.2.2.4 User- and Domain-based Cluster Annotation

We now label repeated tweets posted by spammers and tweets promoting specific links. Instead of restricting to the tweets that match keywords (Section 3.2.1), we consider all tweets in this step.

User-based Cluster Annotation. In this step, tweets posted by the same user are grouped into near-duplicate clusters. Then the clusters with more than 10 tweets are labeled. The top 10,000 users who posted the largest number of tweets in our dataset are considered. We observe that most user-based clusters are small clusters with fewer than 10 tweets. That is, most of the users do not post many similar tweets. As the result, 980 user-based clusters each has more than 10 tweets need to be labeled, consisting of about 25 thousand tweets in total. Among these 980 clusters, 723 are labeled spam (containing about 20 thousand tweets) and 120 clusters are labeled ham (about 2 thousand tweets). Note that, highly similar ham tweets may be posted by the same user repeatedly (*e.g.*, updates of bus services and weather reports). The remaining 137 are labeled unknown, after looking into the content of the tweets and user profiles (*e.g.*, number of followers/followees, number of tweets) .

Because we consider all tweets in this step, some of the tweets in user-based cluster may have already have been labeled in the earlier steps. If the labels are all from the manual annotation, then the cluster is considered labeled. If the labels are given in the kNN annotation, these labels are considered as reference and may be corrected during this manual annotation process.

Domain-based Cluster Annotation. Recall that a tweet may embed links to external sources. Here we aim to label the spam tweets that promote links from specific domains. To this end, the tweets that contain links from any of the top 10,000 most frequent domains in our collection are selected and then grouped into near-duplicate clusters based on domains (all the tweets in the same cluster contain links from the same domain). There are 9,038 such clusters each contains at least 10 tweets. Out of these 9,038 clusters, 4,398 are spam clusters (containing about 221 thousand tweets), and 3,785 are ham clusters (about 121 thousand tweets). The remaining 855 clusters could not be labeled even after manual inspection. Similar to that in user-based cluster annotation, the labels from earlier steps are utilized in this annotation process.

During the annotation process, we observe that domain-based cluster annotation is effective in capturing distributed spam activities in promoting online resources. Examples include links related to topics of working from home, best deals, and coupon codes. Note that, in our labeling process, clusters with tweet content that is clearly ham and links that are from major trustworthy sites (*e.g.*, BBC, CNN, NBA) are labeled ham without looking the content of the web pages. Clusters containing links from less popular domains are labeled by looking into the user profiles and the content of the linked pages. However, it is found that in most cases, tweets provide good descriptions of the links. It is uncommon to have similar tweets linking to very different web pages.

Until this step, most ham tweets are left untouched in the dataset. In the next section, we aim to detect the ham tweets from the remaining unlabeled data.

3.2.3 Reliable Ham Tweet Detection

The remaining unlabeled tweets in our tweet collection may contain a large number of ham tweets, but also contain many spam tweets yet to be identified. Given a large number of remaining unlabeled tweets, this problem can be formulated as a *PU Learning problem*, i.e., learning from Positive and Unabeled examples [LDL⁺03].

Inspired by the approach for building text classifiers from positive and unlabeled examples proposed in [LDL⁺03], we utilized the *positive words* in spam tweets to identify reliable ham tweets. Positive words are the words that appear more frequently in spam tweets than other tweets. Example positive words are ‘followers’, ‘gain’, ‘gameinsight’, ‘please’, ‘win’, ‘iphone’ etc. More specifically, let $rdf(w, D_s) = \frac{df(w, D_s)}{|D_s|}$ be the relative document frequency of word w in the set of spam tweets D_s labeled so far, where $df(w, D_s)$ denotes the number of tweets containing word w in set D_s . Let $rdf(w, D_h \cup D_u)$ be the relative document frequency of word w in the set of labeled ham tweets D_h and the set of unlabeled tweets D_u . Then w is a positive word if $rdf(w, D_s) > rdf(w, D_h \cup D_u)$. Reliable ham tweets are those tweets that do not contain any positive words. In our implementation, words that are shorter than 3 characters in length are ignored. Note that, number of characters in a hashtag does not include the ‘#’ symbol.

Result and Evaluation. This method is simple but found to be very effective in identifying ham tweets. In total 4.093 million tweets are identified as reliable ham tweets. We evaluated the quality of the labeled tweets by manually inspecting a randomly selected sample of 1000 reliable ham tweets. Among them, 32 tweets are incorrectly labeled as ham, hence the precision of the labels in reliable ham detection is 0.968. Following are three examples of reliable ham tweets detected in this step.

- Why can't I sleep
- The first thunder sound scared me
- @username hey :)) do u know what channel is CNN on???? Lol i forgot it...

Table 3.5: Statistics of the labeled tweets in HSpam14 dataset

Annotation step	Spam (million)	Ham (million)
Manual annotation	1.644	0.226
k NN-based annotation	0.502	0.455
User-based annotation	0.020	0.002
Domain-based annotation	0.221	0.121
Reliable ham tweet detection	-	4.093
EM-based annotation	0.951	5.790
Total	3.338	10.688

3.2.4 Annotation by EM-Algorithm

Out of the 14 million tweets in our data collection, more than 7 million tweets are now labeled in the earlier steps. With the 2.387 million spam tweets and 4.897 million ham tweets, we have a sizable collection of labeled examples to build a reliable classifier for the prediction of labels of the remaining unlabeled tweets. To this end, we adopt the Expectation-Maximization (EM) algorithm proposed for text classification using labeled and unlabeled documents [NMTM00]. EM has also been used to reduce human effort in evaluating IR systems [GWO14]. A Naïve Bayes classifier is firstly constructed by utilizing the existing labeled tweets. In the E-Step, this classifier is used to predict the labels of the remaining unlabeled tweets. In the M-Step, the probabilities of the word features are re-estimated based on the predicted labels. Then the classifier in the E-Step is updated accordingly and is used to re-classify the tweets in the M-Step. This process continues until the number of changes in the predicted labels is below 0.01% of the total unlabeled tweets.

As the result, 5.790 million ham tweets and 0.951 million spam tweets are labeled in this step. To evaluate the quality of the labeling, we randomly selected 500 ham and 500 spam tweets labeled by EM for manual inspection. The precisions for spam and ham labels are 0.94 and 0.96 respectively.

3.2.5 Summary

After all these steps, we have labeled almost all the 14 million English tweets. Table 3.5 reports the number of tweets labeled as ham and spam in each step. Note that, the numbers reported under manual annotation are slightly larger than that reported in Table 3.4 because the numbers here include the manual annotations conducted in k NN-based annotation step as well as the annotations made during the evaluation of subsequent steps. In summary, 3.338 million spam tweets and 10.688 million ham tweets are labeled in the HSpam14 dataset. There are two possible reasons for the relatively large number of spam tweets. First, the tweets in our collection were collected based on trending hashtags, which are more likely to be hijacked for spamming purpose. Second, many of the spam tweets are repetition of similar content, *e.g.*, 127,559 near-duplicate tweets are due to the hashtag `#gameinsight`.

3.3 Conclusion and Limitation

In this chapter, we detailed the HSpam14 labeling process. This is the first step for tweet-level spam detection focusing on hashtag-oriented tweet spam. The main contribution of this work is the creation of HSpam14 data collection, which contains 3.338 million spam tweets and 10.688 million ham tweets, collected in two months' time guided by trending hashtags. While the annotation is not made per tweet basis, the overall quality of the labels is good enough to perform analyses and studies on hashtag-oriented tweet spam detection.

A limitation of this data collection is the lack of full profiles of the users. As the tweets in this collection were collected with the guide of trending hashtags, we do not have complete user profiles (*e.g.*, all tweets published by each user, a list of her followers/followees) in HSpam14 due to the restriction of Twitter streaming APIs. This may limit the study on user-level spam detection with this dataset. Another limitation could be the limited keywords used in heuristic-based tweet selection. The existing keywords may not catch spam in specific domains (*e.g.*, healthcare and medical).

Despite some limitations, HSpam14 dataset is a valuable resource which can be utilized to understand different aspects of spam and ham tweets. There are limited studies to understand spamming behavior on Twitter. In the next chapter, we present a detailed fine-grained analysis of spam and ham tweets based on HSpam14 dataset which is crucial to design robust fine-grained spam detection system on Twitter.

Chapter 4

HSpam14 Dataset Analysis

In this chapter, we report a detailed analysis of the HSpam14 dataset to get a better insight of spamming activities on Twitter. The analysis is focused on similarities and differences between spam and ham tweets/users in various aspects based on hashtag, tweet content, and user profile. For hashtag, we compare the usage of hashtags in spam and ham tweets based on frequency, position, orthography and co-occurrence. For content-based analysis, we consider the differences in word usage, and content-based metadata, and also near-duplicate tweets. The user-based analysis aims to get a better understanding of spamming activities on Twitter. There are limited studies to understand fine-grained spamming behavior. However, we have compared and contrasted finding of our study with analysis of the previous studies to get the deeper insights of spamming activities on Twitter.

There are a few studies focused on the analysis of spam and ham tweets [GTPZ10, TGSP11]. Similarly, there are limited spam-related studies [CWW12, GVK⁺12, YRSB10] though their focus is not in spam analysis they are related to our work in a sense that they have done some analyses of their datasets. These studies have reported the findings and it would be useful to validate those findings using independently collected dataset. Further, it would be interesting to compare the analysis based on earlier datasets and detect evolving behaviors of spammers. To this end, we have

Figure 4.1: Cumulative fraction of spam/ham tweets with different number of hashtags per tweet

presented a detailed analysis of the dataset based on hashtags, tweet content and user profile and compare our observation with the finding of previous studies.

4.1 Hashtag Usage in Spam and Ham

In this section, we make a thorough comparison of hashtag usage patterns in spam and ham tweets. We analyze hashtags from three different perspectives: number of hashtags in each tweet, orthographic features of the hashtags, and hashtag co-occurrence. Some of the features have been used to predict hashtag usage in early studies [TR12].

Number of Hashtags in Tweets. We first report the number of hashtags per tweet. Figure 4.1 shows the cumulative fraction of tweets containing different numbers of hashtags, ranging from 0 to 10, for spam and ham tweets. Here, we observe that spammers use many hashtags per tweet, probably with the aim of reaching more users through hashtags. We can also observe that nearly 80% of legitimate tweets have

Table 4.1: Top 20 most used hashtags in ham/spam tweets. Hashtags are in descending order of their frequencies. The most frequent ham hashtag #FF appears in 43K tweets, while the 20th ham hashtag #SS5INADay2 appears in 9K tweets. The most frequent spam hashtag #TEAMFOLLOWBACK appears in 661K tweets, and the 20th spam hashtag #HITFOLLOWSTEAM appears in 127K tweets

Ham	#FF, #NowPlaying, #NP, #SoundCloud, #jobs, #news, #BELIEVEtour, #tbt, #MUFC, #Oomf, #Love, #Music, #occupygezi, #Taurus, #QnA, #Gemini, #Leo, #Virgo, #RT, #SS5INADay2
Spam	#TEAMFOLLOWBACK, #TFBJP, #gameinsight, #androidgames, #OPENFOLLOW, #androidgames, #FF, #RETWEET, #IPADGAMES, #RT, #SougoFollow, #ipad, #FOLLOWBACK, #THF, #FOLLOWNGAIN, #500aday, #AUTOFOLLOW, #MUSTFOLLOW, #TEAMHITFOLLOW, #HITFOLLOWSTEAM

no hashtags, and very few ham tweets have more than two hashtags. In contrast, more than 60% of spam tweets have one or more hashtags, and approximately 40% of them have two or more hashtags. This indicates that spam tweets tend to have more hashtags compared with ham tweets.

In our dataset, we observed that popular hashtags such as #android, #ipad are heavily exploited by spammers to promote their applications and games. Table 4.1 lists the top 20 most frequently used hashtags in spam and ham tweets in our dataset. The frequent hashtags in spam tweets are intended to quickly gain followers, whereas the frequent hashtags in ham tweets are about horoscope, music, love and news. We observe that the frequencies of hashtags in spam tweets are at least one order of magnitude higher than in ham tweets. Interestingly, many of the hashtags that are frequently used in spam tweets are written in capital letters.

Orthographic Features of Hashtags. In this section, we compare spam and ham tweets by analyzing three aspects, namely, (i) the usage of uppercase letters in hashtags, (ii) the usage of numbers, and (iii) the position of hashtags in tweets.

Capitalization is probably the simplest way to emphasize a tweet. In the first three columns of Figure 4.2, we observe that more than 35% of spam tweets contain hashtags written in all capital letters. For instance, #MUSTFOLLOW and #IPADGAGMES

Figure 4.2: Orthography and position of hashtags in spam and ham tweets

are listed in the top 20 most frequent hashtags in spam tweets (See Table 4.1). In contrast, nearly 90% of ham tweets contain hashtags written in lowercase letters or mixed cases. The fourth column in Figure 4.2 shows that hashtags in spam tweets contain slightly more numbers than those in ham tweets.

Now we discuss the position of hashtags, *i.e.*, whether a hashtag appears as the first, middle or last word in a tweet. In Figure 4.2, we observe that hashtags are more likely to appear in the middle of the spam tweet. This observation suggests that spammers are likely to use hashtags as a part of a message. In ham tweets, hashtags are more likely to appear at the end of the tweet content, often used as a topical indicator of the tweet.

Hashtag Spammy Index. We propose a measure named *Spammy Index* to quantify the extent a hashtag is being used in spam tweets. Given a hashtag t , its spammy index, denoted by $si(t)$, is defined in the following equation.

$$si(t) = \log_2(df(t, D)) \times \frac{df(t, D_s)}{df(t, D)} \quad (4.1)$$

Table 4.2: Top-20 hashtags with the highest spammy-index. All hashtags listed in these two tables have frequency larger than 5000, and the hashtags are case-sensitive.

Hashtag	$df(t, D_s)$	$df(t, D)$	$si(t)$
TEAMFOLLOWBACK	661,058	666,328	19.18
TFBJP	526,709	527,176	18.99
gameinsight	509,760	510,504	18.93
OPENFOLLOW	331,643	332,857	18.27
androidgames	286,101	286,706	18.09
android	335,699	341,240	18.06
IPADGAMES	231,725	232,335	17.78
SougoFollow	197,421	197,525	17.58
FOLLOWBACK	176,269	177,109	17.34
ipad	192,706	195,375	17.32
THF	165,407	165,762	17.30
FOLLOWNGAIN	149,708	149,916	17.17
500aday	145,696	146,005	17.12
RETWEET	240,128	250,992	17.10
TEAMHITFOLLOW	136,039	136,043	17.05
AUTOFOLLOW	140,686	141,214	17.04
MUSTFOLLOW	137,502	138,040	17.00
RT	224,640	235,137	16.98
HITFOLLOWSTEAM	127,262	127,264	16.96
Follow	122,575	126,720	16.35

In this equation, $df(t, D)$ denotes document frequency, or the number of tweets containing hashtag t in tweets collection D ; D_s is the set of spam tweets where $D_s \subset D$. A hashtag is considered to be more spammy if (i) its document frequency is high, and (ii) the probability of being used in spam tweets is high.

The top-20 hashtags with the highest spammy indexes are reported in Table 4.2. For comparison, Table 4.3 lists the 20 hashtags with the lowest spammy indexes among all hashtags with frequency more than 5,000. The two tables show that spammy hashtags are more likely to use capital letters, probably to draw attention. Frequent words appear in spammy hashtags are “follow”, “fb” (follow back) and “game”. Hashtags with low spammy indexes are less likely to contain all capital letters and many of them are topic specific on music, sports, or functional tags like #job and #jobs. Comparing Table 4.2 and Table 4.1, most extremely popular hashtags are hijacked for spamming

Table 4.3: Top-20 hashtags with the lowest spammy-index. All hashtags listed in these two tables have frequency larger than 5000, and the hashtags are case-sensitive.

Hashtag	$df(t, D_s)$	$df(t, D)$	$si(t)$
musicjournals	4	6408	0.00
QnA	32	11868	0.01
occupygezi	21	13521	0.01
iCanAdmit	12	6714	0.01
thankyousiralex	25	5788	0.02
Aquarius	43	7270	0.03
blessed	42	6206	0.04
job	80	9786	0.05
BELIEVEtour	206	22758	0.07
Pisces	82	8011	0.07
nbaplayoffs	66	5596	0.07
Capricorn	104	7636	0.09
CFC	125	9271	0.09
jobs	333	26402	0.11
Sagittarius	202	9706	0.16
tbt	382	18157	0.18
TCOT	207	8553	0.19
MUFC	410	17791	0.20
Gemini	283	11156	0.21
Libra	160	5425	0.22

purpose. Popular hashtags such as `#android`, `#ipad` are hijacked for spamming purpose so they appeared as top 20 spammy hashtags. The popularity of these hashtags may attribute to the large number of near-duplicate tweets automatically generated by bot like `#gameinsight`.

For the completeness of the analysis, we also computed the Odds Ratio (OR) of the words (including both case-insensitive hashtags after removing the `#` symbol and words appear in the content of tweets) in distinguishing spam tweets from ham tweets. The top-20 words with highest OR values are reported in Table 4.4. The words identified by high OR values are consistent with the hashtags of high spammy indexes, with the dominating word “follow”.

Table 4.4: Top-20 most spammy words by Odds Ratio

Rank	Word		Rank	Word
1	hitfollowsteam		11	followers
2	tribez		12	ifollowngain
3	maxvip		13	followbacksegur
4	r_family		14	teamfollowwacky
5	follow2befollowe		15	smurfberries
6	teamhitfollow		16	myretweets
7	extraretweets		17	reignd
8	onlyifyouare		18	mesiguestesigo
9	rt2club		19	90sbabyf
10	rt2supergain		20	harvested

Hashtag Co-occurrence. Nearly 90% of ham tweets contain zero or one hashtag while more than 40% of spam tweets have two or more hashtags (see Figure 4.1). This suggests that many co-occurring hashtags are likely to be exploited by spammers.

We propose the *Spammy Index* to quantify the extent a hashtag is used in spam tweets [SS15]. Given a hashtag t , its spammy index, denoted by $si(t)$, is defined in Equation 4.2.

$$si(t) = \log_2(df(t, D)) \times \frac{df(t, D_s)}{df(t, D)} \quad (4.2)$$

Where $df(t, D)$ denotes hashtag t 's document frequency (*i.e.*, the number of tweets containing hashtag t) in tweet collection D ; and D_s is the collection of spam tweets and $D_s \subset D$

A hashtag has a high spammy index if (i) its document frequency is high, and (ii) its probability of being used in spam tweets is high. Figure 4.3 plots the hashtag co-occurrence graph. Hashtags that appear more than 25,000 times in our dataset are included in this graph. The size of a node is proportional to its frequency in our dataset and width of an edge is proportional to co-occurrence frequency. The spammy index of a node is represented by the color of the node, where green is the least spammy, blue is moderately spammy and red is the most spammy.

Figure 4.4: Semantic similarity of co-occurring hashtags and the fraction of times the pair of hashtags appear in spam tweets. The number of hashtag pairs in each hexagon is indicated by its color.

the profile of any of the hashtags. By using tweets containing exactly one hashtag, we aim to more precisely define the meaning of each hashtag. The semantic distance between a pair of hashtags is estimated using the Jensen-Shannon divergence between hashtag profiles.

There are 3204 hashtags in our dataset that each appears at least 50 times in the tweets having only one hashtag. We profile these 3204 hashtags. Among these hashtags, there are 3531 pairs of hashtags that co-occur more than 50 times. The mean Jensen-Shannon divergence of these co-occurring hashtag pairs is 0.490 with standard deviation 0.169.

Figure 4.4 plots the relation between the semantic distance of a co-occurring hashtag pair (*i.e.*, Jensen-Shannon divergence on x-axis) and the fraction of times the pair appears in spam tweets (y-axis). We observe that spam and ham tweets contain semantically similar as well as semantically dissimilar hashtags. In a legitimate tweet

Table 4.5: Ten co-occurring hashtag pairs in spam and ham tweets having low and high Jensen-Shannon divergence

	Low Divergence	High Divergence
Spam	(#followback, #TFB) (#BlackBerry, #Apple) (#ipadgames, #iphone) (#Entertainment, #Travel) (#Get, #followback) (#life, #dead) (#win, #competition) (#Entertainment, #fashion) (#London, #USA) (#MustFollow, #FollowMe)	(#threewords, #NBAFinals) (#Shoutout, #MUFC) (#BETAwards, #SingleBecause) (#ff, #androidgames) (#TeenageBio, #TheOlderIGet) (#music, #Retweet) (#New, #deal) (#nbafinals, #BornSinner) (#MUSIC, #ANDROID) (#IWishIWas, #GRIZZLIES)
Ham	(#nature, #green) (#party, #happy) (#crazy, #love) (#lovely, #cute) (#yum, #food) (#CNN, #BBC) (#MONEY, #business) (#cute, #lol) (#tech, #technology) (#food, #foodporn)	(#thankyousiralex, #mufc) (#heartbreaker, #musicjournals) (#OccupyGezi, #CNN) (#CFC, #AFC) (#photography, #music) (#24seven, #Rushers) (#BBMA, #swifties) (#Photo, #beautiful) (#Best, #friends) (#beautiful, #Art)

semantically dissimilar hashtags may provide more precise meaning by providing the intersecting topics of two hashtags. For example, `#mufc` is a hashtag about football club Manchester United, which is used in many contexts, whereas two hashtags `#mufc` and `#thankyousiralex` together make it easier to understand that it is a farewell tweet thanking the manager of the club Manchester United, Sir Alex. In contrast, in spam tweets, spammers exploit popular hashtags without considering their semantics, probably to reach a wider audience. For example, `#threeword` is a popular hashtag used with three-word sentences such as ‘i love you’, and `#NBAFinals` is another popular hashtag; manual inspection of the tweets containing the hashtags shows that these hashtags do not provide any additional meaning to the tweets.

Popular hashtags such as #ANDROID, #music, and #NBAFinals have been heavily exploited in spam tweets, even though these hashtags are not only semantically different from the co-occurring hashtags but also found to be unrelated to the tweets from manual inspection. This observation is consistent with earlier studies on spam datasets [GTPZ10]. However, legitimate tweets may also contain redundant hashtags to emphasize a particular topic. Hashtag pairs such as (#party, #happy), (#nature, #green) are used to emphasize the topic. Spam tweets also use semantically similar hashtags with the same motivation and malicious intention. In Table 4.5, we list more examples of hashtag pairs with low and high Jensen-Shannon divergence used in spam and ham tweets. In short, both spam and ham tweets may contain semantically similar and semantically dissimilar hashtags. However, hashtags present in a spam tweet act as a channel to propagate the tweet to a wider audience but do not provide any additional information to the tweet.

4.2 Content Analysis

In this section, we perform content based analysis by analyzing frequent words and the usage of sentiment words, links, exclamation points, question marks, capitalization, positive emoticons, negative emoticons and the money sign in spam and ham tweets. These content-based metadata have also been used in many applications such as spam detection [BMRA10], social bot detection [FVD⁺14], and credible tweet detection [CMP11].

4.2.1 Words and Content-based Metadata

Frequent Words. The usage of frequent words in spam tweets is different from that in ham tweets. Table 4.6 reports that spam tweets in the dataset often use spammy words such as ‘follow’, ‘teamfollowback’ and ‘openfollow’, whereas these words are less likely to appear in ham tweets. Furthermore, spam tweets exploit globally popular

Table 4.6: The top 20 most frequently used words in ham and spam tweets

Ham	you, my, me, so, your, love, just, have, like, i'm, all, go, up, when, get, out, new, lol, we, now
Spam	follow, followers, retweet, you, teamfollowback, new, tfbjp, gameinsight, me, want, android, i've, openfollow, my, androidgames, ff, gold, coins, collected, more

Figure 4.5: Differences in content metadata in spam and ham tweets

words such as ‘android’ and ‘androidgames’, whereas legitimate tweets mostly contain words commonly used in day-to-day conversations.

Content-based Metadata. In this section, we analyze content-based metadata: sentiment words,¹ links, exclamation points, question marks, capitalization, positive/negative emoticons² and the money sign. Figure 4.5 shows the presence of these attributes in spam and ham tweets. The figure indicates that ham tweets are more likely to contain sentiment words compared with spam tweets. More than 35% of

¹We used the sentiment word list available at http://neuro.imm.dtu.dk/wiki/A_new_ANEW:_evaluation_of_a_word_list_for_sentiment_analysis_in_microblogs

²The emoticons are listed at <http://www.datagenetics.com/blog/october52012/index.html>

spam tweets contain links whereas only 18% of the legitimate tweets contain links. The significantly greater usage of links in spam tweets is probably to promote the associated Web pages.

We also observe that exclamation points are more likely to appear in spam tweets than in ham tweets. In contrast, question marks are more likely to appear in ham tweets. We refer to a tweet containing more than 25% of characters in uppercase as a capitalized tweet. Approximately 30% of the spam tweets in our data set are capitalized tweets and only 4% of ham tweets are capitalized. Usage of money signs is slightly greater in spam tweets than in ham tweets. Spammers use more exclamation points, capitalized words, and money signs, probably to draw the attention of users. For emoticons, spam tweets are more likely to contain positive emoticons than ham tweets. Ham tweets contain relatively more negative emoticons. One reason for the lesser usage of negative emoticons in spam tweets could be that negative emoticons are less pleasing to users.

4.2.2 Near-duplicate Tweets

To understand the collective behavior of spammers, we analyze spam and ham tweet clusters formed by near-duplicate tweets. Near-duplicate clusters are manually labeled when annotating the HSpam14 dataset (see Table 3.5). The near-duplicate tweets are identified by using the MinHash based algorithm [SS15], and only those near-duplicate clusters that are manually labeled are used in this analysis and not the clusters labeled by k NN. There are 11,130 manually labeled clusters, of which 6,225 are spam clusters, and the remaining are ham clusters. One of the reasons for having near-duplicate spam tweets could be the collective spamming activities of users (*e.g.*, similar tweets promoting the same product/service that are posted by many spammers). Ham tweet clusters, however, are probably due to the intrinsic feature of Twitter, *e.g.*, using few words to express a statement or report events.

Previous studies show that there are spam campaigns on social networks [GHW⁺10, GTPZ10]. Spammers distribute spamming tasks among spam accounts to remain

hidden from spam detection systems [CWW12]. In our dataset, we also observe that spammers post content aggressively, and they distribute such activities among many users.

Aggressive Content Promotion. In our dataset, we observe that spam content is posted aggressively, probably to draw the attention of users. Of the top 10 largest clusters based on the number of tweets, there is only one cluster of ham tweets. The one cluster of ham tweets is from automated posts of Facebook on behalf of its users. The largest cluster contains 127,559 promotional tweets posted over 42 days. During this time span, there were two days in which more than 10,000 tweets were posted; on most of the remaining days, more than 1,000 tweets were posted. Similar aggressive posting of tweets is observed in the other top 10 largest clusters. Furthermore, of the top 10 clusters of spam tweets, 9 clusters contain links, and each cluster contains thousands of near-duplicate tweets with links from one particular domain. Such aggressive posting of tweets containing links from a domain is probably to promote the website.

In the HSpam14 dataset, there are 3.3 million spam tweets, of which 71.5% are detected by the near-duplicate detection approach. Such a significant fraction of near-duplicate tweets is due to the aggressive posting of similar tweets by spammers. In contrast, only 7% of ham tweets in this dataset are labeled based on near-duplicates. Near-duplicate ham tweets are due to the retweet of popular posts or tweets from legitimate online services such as Facebook, Foursquare, or Instagram. Hence, aggressively posted promotional tweets could be effectively detected by the near-duplicate detection approach.

Distributed Activity. We now analyze the near-duplicate clusters of spam tweets based on domains. There are 4398 clusters of spam tweets, where tweets in each cluster contain links to one particular domain. In other words, all the tweets in the same cluster are promoting the same website. Figure 4.6 plots the number of such clusters against the number of users (or tweet accounts) in each cluster who posted

Figure 4.6: The distribution of the number of clusters against the number of users posting these tweets in each cluster. Tweets in each cluster contain links to one particular domain.

the tweets. The figure shows that approximately 800 clusters contain near-duplicate spam tweets that are posted by at least 50 Twitter accounts. It is less likely that legitimate users post such spam tweets. Manual inspection of the users posting these tweets shows that these accounts are likely to be created solely for posting promotional tweets. In most of these clusters, the tweets are posted by more than 10 users, whereas only approximately 5% of the clusters contain tweets by a single account. Previous studies show that multiple accounts are exploited to promote spam content which is consistent with our observation [CWW12, GTPZ10]. Because creating a Twitter account is relatively easy, spammers may have created multiple accounts for posting spam tweets.

Figure 4.7: Follower distribution of ham users

4.3 User-based Analysis

Yardi *et al.* [YRSB10] reported that the follower-to-followee ratio of spam and legitimate users is not significantly different. However, the total number of followers and followees of spammers is three times that of legitimate users. In contrast to their findings, we observe that a large number of spammers have fewer than 10 followers and/or followees which may be a strategy of spammers to remain undetected.

We consider a user to be a *spam user* if the user has posted at least five tweets and all of them are spam tweets. Similarly, a user is considered as a *ham user* if the user has posted at least five tweets and all of them are ham tweets. There are 32,581 ham users and 36,662 spam users in the HSpam14 dataset. With these two sets of users, we analyze followers, followees and the information provided in user profiles (*e.g.*, number of tweets, location, URL, time zone, and profile description).

Figure 4.8: Follower distribution of spam users

4.3.1 Followers and Followees

Previous studies reported that spammers try to have as many followers as possible. In our dataset, we observe that there are spammers who have thousands of followers. However, in Figure 4.8 we notice that almost 40% of spam users have fewer than 10 followers. In general, spammers with fewer followers and followees have less chance to be detected or reported as spammers, even though they post many spam tweets. Spammers may intentionally create many accounts each with a small number of followers/followees to promote their content. These spammers are difficult to identify by graph-based or social-honeypot-based spammer detection systems because they are disconnected from the social graph and do not follow many users, but content-based methods could identify these spammers.

In an earlier analysis based on a dataset of 2009 [YRSB10], the authors reported that spammers are likely to have more followers and followees compared with legitimate users. However, recent studies on a spam dataset of 2013 (which was collected around the same time as our dataset) show that approximately 40% of spammers have no

Figure 4.9: Followee distribution of ham users

follower and followee, which is consistent with our result [TLGP14]. This indicates that spammers may have changed their strategy from having more followers to hiding under the radar. Understanding such evolving behavior of spammers is crucial for the effective detection of Twitter spam.

Figures 4.7 and 4.8 show that there is an order of magnitude more spam users than ham users having fewer than 10 followers. Figures 4.9 and 4.10 show a similar pattern for the number of followees of spam and ham users. From Figures 4.7 and 4.8, it can be inferred that, in the range of having 0 to 70 followers, the number of spam users drops rapidly compared with ham users. A similar pattern is observed for the number of followees of the spam and ham users. A user having very few followers and followees could be a spammer, although a new or inactive legitimate user may also have a small number of followers/followees. In Figures 4.9 and 4.10, we observe a sudden increase in the number of users having approximately 2000 followees; similar behavior was observed in a previous study [TGSP11]. A Twitter user following other users is likely to be followed back by those users. Using this trick, a user may obtain a

Figure 4.10: Followee distribution of spam user

large number of followers. However, Twitter has a restriction on the number of users that a user can follow. If a user does not have enough followers, the user will not be able to follow more than 2000 users. The abrupt increase in the number of users having approximately 2000 followees is probably due to this restriction. Unexpectedly, Figures 4.9 and 4.10 show that both spammers and legitimate users are affected by this restriction. Further analysis of such users utilizing their profile information may give more insights.

4.3.2 User Profile-based Analysis

In this section, we perform user profile analysis from different perspectives, namely, the number of tweets, location, URL, time zone, and description, that are commonly listed in user profiles. It is reported that spammers try to mimic the behavior of legitimate users to avoid the spam detection system [LEC11]. However, we observe that spam

Figure 4.11: Distributions of number of tweets of spam user. The number of tweet-user pair in a bin is indicated by its color.

and ham users have significant differences based on the information provided in user profiles.

Figures 4.11 and 4.12 report the distributions of spam and ham users, which are grouped by the number of tweets posted per user. We observe that there are a significantly large number of spammers posting 500-5000 tweets. The large number of tweets may have been posted for content promoting purposes. It can be observed that, spammers post more tweets than legitimate users. However, Figure 4.12 shows that there are also legitimate accounts that post significantly more tweets than spam users. Manual inspection of these accounts reveals that most of them are associated with news agencies, radio stations, or weather reports. Such accounts usually post the most up-to-date information, resulting in a large number of tweets.

Furthermore, Figure 4.13 shows that more than 70% of legitimate users provide location information, while less than 50% of spam users provide location information in their profiles. About half of ham users provide URLs in their profiles and only

Figure 4.12: Distributions of number of tweets of spam and ham users. The number of tweet-user pairs in a bin is indicated by its color.

approximately 20% of spam users do the same. Similarly, approximately 80% of ham users provide time zone information, and the percentage of spam users doing so is less than 60%. It is also observed that more than 35% of legitimate users provide complete user profile information (*i.e.*, location, URL, time zone and description), whereas only 10% of spam users provide full information in their user profile pages. Legitimate users are more likely to provide profile descriptions than spammers. Figure 4.13 shows that more than 90% of legitimate users provide profile descriptions, which is much higher than the 56% of spam users. In Table 4.7, we list the top 20 most commonly used words in their user profiles, which shows that our observation is consistent with the observations of a previous study [GVK⁺12]. Although there are many words in common among the user profile descriptions of spam and ham users, the two words ‘follow’ and ‘followback’ are certainly more frequent in the descriptions of spam user profiles.

Table 4.7: Top-20 most commonly used words in ham and spam users' profile descriptions, ordered by their frequencies.

Ham	love, follow, news, all, life, from, just, music, like, about, don't, fan, one, world, get, twitter, have, who, more, instagram
Spam	follow, love, back, mention, life, fallback, teamfollowback, roleplayer, one, instagram, followers, twitter, music, always, all, more, followback, family, que, god

Figure 4.13: Fraction of spam/ham users providing information in user profile page

4.4 Case Study: Spam Detection

The analyses of the various aspects of spam and ham tweets provide insights about the differences in the behavior of spammers and legitimate users. The next question is, *are the observed differences useful in identifying spam tweets from ham tweets?* Based on our analysis, in this section, we design features and evaluate their effectiveness in the spam detection task. Note that, as a case study, the spam detection task here is formulated as a simple binary classification task for the purpose of evaluating the effectiveness of the features.

Table 4.8: The 15 features derived from this study and ranked by Gini coefficient

Rank	Tweet feature	Gini coefficient
1	Contains more than 2 hashtags	0.0405
2	Contains spammy hashtag	0.0175
3	Has less than 5 percentile followers	0.0147
4	Followers-followees ratio	0.0114
5	User profile contains description	0.0110
6	Has less than 5 percentile followees	0.0096
7	Contains capitalized hashtag	0.0062
8	Fraction of upper case characters	0.0060
9	Contains URL in the tweet	0.0046
10	Contains exclamation sign	0.0037
11	Percentile of followers of the user	0.0034
12	User profile contains time zone info	0.0032
13	Contains negative sentiment words	0.0029
14	Contains URL in user profile	0.0023
15	Suffix hashtag	0.0019

We first evaluate the features based on Gini coefficients with the tweets posted on a randomly selected day (17th May 2013) in the HSpam14 dataset. There are 22,185 ham tweets (only manually labeled and reliable ham tweets are used here) and 48,849 spam tweets on the selected day. We evaluated 39 features based on our analysis, such as the usage of hashtags and user profile information. Table 4.8 reports the top 15 features ranked by their Gini coefficients. We observed that spammers exploit many hashtags, which are found to be discriminative features for spam tweet detection (Feature 1). As shown in Figure 4.1, we observe that having more than two hashtags is an important feature for identifying spam tweets. We also observe in our analysis that to remain undetected, the spammer does not follow many other users (Features 4 and 6). This insight is found to be useful for spam detection on Twitter using the Gini coefficients of the features. Similarly, having user profile information such as a profile description (Feature 5), time zone information (Feature 12) and a URL on the profile page (Feature 14), are found to be discriminative features. The presence of exclamation point (Feature 10), URL (Feature 9), and negative sentiment words (Feature 13) in tweet contents are also found to be important attributes in identifying spam tweets.

Table 4.9: The top-15 terms (words and bi-/tri-grams) ranked by Gini coefficients and the top-15 hashtags ranked by spammy index. The matched terms and hashtags in both lists are highlighted in boldface.

Terms	Hashtags
follow	#TEAMFOLLOWBACK
follow back	#TFBJP
back	#gameinsight
ipad ipadgames	#OPENFOLLOW
please follow	#androidgames
please	#android
followers	#IPADGAMES
retweet	#SougoFollow
tfbjp	#FOLLOWBACK
teamfollowback	#ipad
ipad	#THF
follow me	#FOLLOWNGAIN
collected	#500aday
followback	#500aday
gameinsight	#TEAMHITFOLLOW

It is observed that the orthographic features of hashtags are discriminative features for spam detection. We observe that spammers use significantly more capitalized hashtags compared with legitimate users, shown in Figure 4.2. Interestingly, a 2012 study [TR12] reported that the orthography of hashtags plays an important role in better spreading ideas in the microblog community. In our dataset, where the tweets were collected in 2013, the same technique has been heavily exploited by spammers.

We further computed the Gini coefficients of the terms (*i.e.*, words and bi-grams and tri-grams) in our dataset. Table 4.9 lists the top 15 terms ranked by Gini coefficients and the top 15 hashtags ranked by spammy index. In the two lists, six of them are exact matches, highlighted in boldface in the table. This observation suggests that the spammy index does capture discriminative hashtags for spam detection.

To evaluate the effectiveness of these features, we trained a logistic regression classifier using the tweets posted on the selected day (17th May 2013), and then used the classifier to detect spam tweets from the tweets posted on the following day (18th

May 2013). In the training dataset, we used manually labeled tweets and reliable ham tweets. There are 0.071 million training tweets and 0.371 million test tweets in this case study. From the test tweets, we randomly selected 400 tweets for manual annotation. The simple classifier achieves precision, recall, and F_1 of 0.96, 0.77, and 0.86, respectively. The high precision and relatively good recall show that spam detection at the tweet-level can be achieved with reasonable accuracy.

4.5 Discussion

In this section, we compare the observations from our analyses with those reported in previous studies. We observe that spammers use popular hashtags in their tweets, which is consistent with the findings reported in [GTPZ10]. Tsur *et al.* [TR12] report that hashtag capitalization helps to promote tweets, and we observe that spammers heavily exploit capitalized hashtags to attract attention. Based on a dataset of 2009, Yardi *et al.* [YRSB10] report that spammers are likely to have more followers and followees than legitimate users. However, a recent study [TLGP14] based on a dataset of 2013 shows that approximately 40% of spammers have no followers or followees, which is consistent with our observation. The change in the behavior of spammers over time could be due to their strategy to evade graph-based spam detection system. Similar to the finding reported in [TGSP11], we also observe that both spammers and legitimate users are affected by the limit of 2000 friends on Twitter.

Our analysis shows that spammers provide less profile information compared with legitimate users. However, Ghosh *et al.* [GVK⁺12] reported that link farmers make heavy use of their profile information, which is inconsistent with our observation. It would be interesting to conduct a separate behavior analysis for different types of spammers such as promoters, link farmers, crowdturfers, and social bots. However, Ghosh *et al.* [GVK⁺12] also reported that spammers are likely to use more promotional words, which is also observed in our study. Likewise, as in the previous

studies [CWW12, GVK⁺12], our analysis also shows that spammers exploit multiple accounts to promote their content.

Our observations on the dataset may lead to more effective spam detection in Twitter. Our analysis on the features and case study on spam detection show that spam detection at tweet-level can be achieved with reasonably good accuracy by using an off-the-shelf classifier. By combining with the user-level spam detection, a more robust spam detection system could be developed. Particularly, we observe that many spammers have very few followers and may not follow a large number of accounts. Furthermore, many spammers do not provide much information on their user profile pages. The lack of user information (*e.g.*, URL) and social graph participation make the detection of user-level spam more challenging. However, spammers tend to attract the attention of users to reach a large audience by including more hashtags, capitalized words, and so on. Spammers also collectively post near-duplicate tweets because of the recency ranking of the tweets that appear in user timelines. These observations suggest using features derived from tweet content and also spam detection at tweet cluster level (*e.g.*, near-duplicates of tweets). Based on the comparison with earlier studies, spammers have indeed changed strategies to avoid detection. Therefore, a semi-supervised spam detection system should be designed to detect such evolving spammers effectively.

4.6 Conclusion

In this chapter, we analyze various aspects of spam and ham tweets based on the HSpam14 dataset. We observe that spam tweets contain many hashtags whereas a majority of ham tweets do not contain any hashtags. Furthermore, spammers tend to use popular hashtags to reach a wider audience beyond their followers. Orthographic features of spam and ham hashtags are also found to be significantly different. Spammers tend to highlight their tweets by using capitalized words and exclamation points. Spammy words such as ‘teamfollowback’ and ‘openfollow’ are common in

spam tweets, whereas such spammy words are rare in ham tweets. We observed that spammers aggressively post near-duplicate tweets to promote their tweets. Moreover, multiple accounts are engaged to promote the same content. Spammers may also remain disconnected from the social graph to avoid social graph-based spam detection. We also observed that legitimate users tend to provide detailed information on their profile pages compared with spammers.

We note that the HSpam14 dataset does not contain the full tweet history of all users. Because of the limited user information, spam and ham users are defined based on the labels of their tweets available in the dataset. Hence, performing analysis using the complete timeline and social graph of the users may give better insights regarding spammers and legitimate users. Because of the limitation of the dataset, we could not conduct analysis on the social graphs, and thus our analyses are mainly focused on the content of tweets.

Comparing our analysis with results of previous studies, we observed that the spammers are exploiting some of the tricks such as using popular hashtags, capitalization of hashtags for a long time. Similarly, it is also observed that spammers have changed their strategy over time. We noted that spammers have modified their approach from having more followers and followees to few followers and followees to evade social graph based spam detection system. We also evaluated the discriminative power of the features discussed in this chapter. Based on our observations and findings, in the next chapter we propose a real-time semi-supervised framework for spam detection on Twitter.

Chapter 5

Real Time Spam Detection

In this chapter, we present a semi-supervised real-time fine-grained spam detection framework for Twitter. Twitter is an effective platform for information propagation as a tweet of a user is broadcasted to all her followers. Moreover, it is reported that average path length among Twitter users is 4.12 [KLPM10]. Because of the well-connected social graph, a few retweets may expose a tweet to a large number of users. It is also reported that a tweet propagates faster than the earthquake [SOM13]. Spam tweets also have the same advantage of quick propagation on Twitter. Further, the click-through rate of links present in a tweet has two orders of magnitude higher than email [GTPZ10]. Hence, real-time spam detection is crucial to deal with spam on Twitter.

Most of the existing studies on Twitter spam focus on account blocking, which is to identify and block spam users, or spammers. Similarly, information extracted from user's tweets, demographics, shared URLs, and social connections are utilized as features in standard machine learning algorithms to detect spam users [LCW10]. However, this approach is less effective for spammers who may act as legitimate users by posting non-spam content regularly. Blocking spammers may even hurt a legitimate user who happens to grant permission to a third-party application that posts spammy tweets under her username. Spammer detection system usually detects a spammer only after the spammer post many spam tweets. Spammers also exploit trending

hashtags to reach a wider audience. Trending hashtags pollute not only the timeline of the followers of the users but also Twitter search results which degrade Twitter search experience. Hence, tweet-level spam detection is increasingly important to deal with such spam tweets.

There are a few studies on spam tweets detection; however, these approaches use standard classifiers to classify spam tweets [SML⁺13, MRA13, WZLP15]. These spam tweet detection systems are reported to be effective but we can further incorporate domain blacklisting, near-duplicate tweets detection and reliable ham tweet detection in such spam detection system to achieve a more robust system. Further, spammers keep on evolving over time hence semi-supervised approach is suitable for tracking such changing spamming activities. Semi-supervised spam detection method has been utilized to identify spam on voice-over-IP call [WBSW09]. Similarly, a semi-supervised approach is reported to have better performance than supervised approach for malware detection task [SNB11]. Semi-supervised model continuously learns new spamming activities which is effective to identify evolving activities. Chen *et al.* suggested that training dataset should be continuously updated to deal with changing distribution of features in tweet stream [CZX⁺15]. However, to the best of our knowledge semi-supervised approach has not been utilized to detect spam tweets. To this end, we have proposed fine-grained semi-supervised spam detection system that detects spam in real-time.

5.1 Semi-supervised Spam Detection

The proposed S³D contains two main modules as shown in Figure 5.1. Assuming that we have all the information (*e.g.*, a blacklist of spamming domains, and trained classification models), the tweets are labeled as spam and non-spam (or ham) tweets using the four detectors in real-time (the left hand side of the figure). The required information is updated periodically based on the confidently labeled tweets from previous time window. Next, we detail the two main modules.

Figure 5.1: System Overview

5.1.1 Spam Tweet Detection in Real Time

For efficiency reason, the tweets are labeled by four light-weight detectors from four perspectives, in an order of the easiest to hardest in terms of difficulty in detection. Once a label is assigned by one of detectors, the tweet need not pass to the next detector.

Blacklisted Domain Detector. Spammers promote their services/products by posting links in their tweets [GHW⁺10]. An effective way of spam detection is to detect tweets containing links from blacklisted domains. The list of blacklisted domains is to be updated at the end of each time window utilizing confidently labeled tweets, during the batch update.

Near-Duplicate Detector. Tweets that are near-duplicate versions of pre-labeled spam/ham tweets are assigned the same labels accordingly. The near-duplicate tweets are detected by using the *MinHash* algorithm [Bro97], which has shown effectiveness for labeling spam tweets [SS15]. More specifically, a signature is computed for each tweet by concatenating the three minimum hash values computed from the tweet's *uni-gram*, *bi-gram*, and *tri-gram* representations, respectively. If two or more tweets have the same signature, then the tweets are considered near-duplicates. If the cluster of near-duplicate tweets having the same signature have been labeled as spam or ham

tweets, the new tweet having the same signature receives the same label. This detector deals with the spam tweets that are posted by content promoters through different user ids or social-bots.

Reliable Ham Tweet Detector. Tweets that are posted by legitimate users can be considered as ham tweets, however; spammers may pretend as legitimate users to post spam tweets [YHG11]. For this reason, we consider a tweet to be a *reliable ham tweet* if it satisfies two conditions: (i) the tweet does not contain any spammy words, and (ii) the tweet is posted by a trusted user.

Spammy words are words whose probability of occurrence is larger in spam than in ham tweets. For example, word *followme* is likely to appear in spam tweets but the word may appear in ham tweet as well. Specifically, let $p_s(w)$ be the probability of word w appearing in the spam tweets, and $p_h(w)$ be the probability of w in ham tweets. Then w is a spammy word if $p_s(w) > p_h(w)$. In our implementation, words that are shorter than 3 characters in length are ignored. The number of characters in a hashtag does not include the ‘#’ symbol.

A trusted user is a user who has never posted any spam tweet and has posted at least 5 confident ham tweets. A tweet is a confident ham tweet if the tweet does not contain any spammy words and is predicted to be ham by all the three classifiers in the tweet classification detector. To be detailed later, the clusters of near-duplicate tweets will also be predicted as clusters of spam or ham by multiple classifiers. The tweets in a cluster which is predicted to be ham cluster by all classifiers are also considered confident ham tweets.

During the batch update, the list of trusted users will be updated, *i.e.*, adding new users who have now accumulated at least 5 confident ham tweets and removing the users who have posted any spam tweets. The list of spammy words is updated as well.

Multi-classifier based Detector. Tweets that are not labeled in any of the previous steps are processed and labeled in this step. Here, we develop a spam detector by using three efficient classifiers, namely Naïve Bayes (NB), Logistic Regression (LR),

and Random Forest (RF). The three classifiers use different classification techniques, *i.e.*, generative, discriminative, and decision tree-based classification models. A full spectrum of features is extracted to represent each tweet. Listed in Table 5.1 in the column titled “Features for tweet representation”, the features include hashtag-based features, content-based features, user-based features, and domain-based features. Most features are self-explanatory and we only elaborate two features, categorical hashtag and top domains. Categorical words are the words used in one of the top-level categories in Yahoo! hierarchy, or words used to categorize content in four Web sites: bbc, cnn, nytimes and reddit. There are 75 categories including sports, technology, business, movie, jobs etc. The binary feature is 1 if the hashtag is one of the categorical words. The domain feature is based on the domain of the URLs contained in tweets. Domain ranking is from alexa.com. A tweet is labeled as spam if at least two of the three classifiers predict the tweet to be spam; otherwise the tweet is labeled as ham.

5.1.2 Model Update in Batch Mode

We now discuss the update module. The time window for an update is set to be *one day* in our experiments. The key desideratum is to identify the confidently labeled data of the previous time window.

Confidently Labeled Tweets. Similar to that in co-training, confidently labeled tweets are automatically labeled tweets with high confidence, such that including these tweets to train new classifiers help to improve system performance. Tweets that are labeled by the first three detectors (*i.e.*, blacklisted domain, near-duplicate, and reliable ham tweet) are considered as confidently labeled tweets. For the classifier based detector, recall that we use three classifiers each is based on a different classification technique. Tweets that are labeled as spam by all the three classifiers are considered as confidently labeled spam tweets. Similarly, tweets that do not contain any spammy words and are labeled as ham by all three classifiers are confidently labeled ham tweets.

Table 5.1: Features used to represent tweets and clusters for classification; **FoT** means *Fraction of Tweets* and **FoU** means *Fraction of Users*. The top-15 most effective features for tweet classification and cluster classification based on the Gini impurity score are indicated by the numbers (x) following the features ($1 \leq x \leq 15$).

Type	Features for tweet representation	Features for cluster representation
Hashtag	Contains hashtag Contains more than 2 hashtags (1) Contains spammy hashtag (2) Contains categorical hashtag (7) Contains capitalized hashtag (8)	FoT having hashtag (3) FoT having more than 2 hashtags (5) Hashtag per tweet (12) FoT having spammy hashtag (14) FoT having categorical word as hashtag FoT having capitalized hashtag (4)
Content	Fraction of words that are spammy Contains question mark Contains money sign Contains exclamation sign (13) Contains positive emoticons Contains negative emoticons Contains positive words Contains negative words Fraction of uppercase characters (9) Contains URL (11) Is retweet Contains mention (14) Contains first person pronoun Contains second person pronoun Contains third person pronoun Normalized length of the tweet in word Normalized length of the tweet in character (10) Contains top 10,000 uni-/bi-/tri-gram Day of the week in which the tweet is posted	FoT having spammy words FoT having question mark FoT having exclamation mark (10) FoT having money sign FoT having positive emoticons (7) FoT having negative emoticons FoT having positive words FoT having negative words (8) Fraction of capitalized tweets (2) FoT that are retweet (6) FoT having URL Mentions per tweet (15) FoT tweets having first person pronoun FoT tweets having second person pronoun FoT tweets having third person pronoun Median tweet length in word/(max tweet length) Median tweet length in characters /140 Contains top 10,000 uni-/bi-/tri-gram Ratio of spam tweets in the cluster (1)
User	Has less than 5 percentile followers (3) Has less than 5 percentile followees (6) Has more than 50 percentile total tweets Percentile followers of the user (15) Percentile followees of the user Percentile total tweet count of the user User profile contains description (5) User profile description contains spammy words User profile has url User profile has location info User profile has time-zone info Followers-followees ratio (4) User's normalized age (12)	FoU having less than 5 percentile followers FoU having less than 5 percentile followees FoU having more than 50 percentile total tweets Median percentile of followers of users Median percentile of followees of users Median percentile of total tweets of the users (11) FoU having description in profile FoU having spammy words in description FoU having URL in profile FoU having location info in profile FoU having timezone info in profile FoU having followers greater than followee Median normalized age of users Fraction of post by the dominating user (9) Percentile followers of the user tweeting the most Percentile total tweets of the user tweeting the most Tweets per user Standard deviation of normalized age of users (13) Percentile followers of the most followed user Percentile followees of the most followed user Percentile total tweets of the most followed user
Domain	URL from top 100 domains URL from top 1000 domains URL from top 10000 domains	FoT having URL from top 100 domains FoT having URL from top 1000 domains FoT having URL from top 10000 domains

Excluding ham tweets containing spammy words (*e.g.*, *followme*) helps to prevent the deviation of classifier from a burst of spammy words in ham tweets.

The identified confidently labeled spam tweets are utilized to update *blacklist domains*, and confidently labeled ham tweets are utilized to identify *trusted users*.

Near-duplicate Cluster Labeling. Recall that the near-duplicate detector computes a signature for each tweet for detecting whether the tweet is a near-duplicate version of previously labeled tweet-cluster. If a tweet’s signature does not match any pre-labeled cluster, then the tweet is passed to the next level detectors.

After each time window, all the tweets that do not match pre-labeled clusters but having the same signature are grouped into a new cluster; *i.e.*, each cluster is a collection of near-duplicate tweets. Next, we label the clusters each containing at least 10 tweets and if the labels are of high confidence, then the signatures of these newly labeled confident clusters will be used by the near-duplicate detector in the next time window. Recall that all the tweets have been labeled as spam and ham tweets (see Figure 5.1), an easy approach to label these clusters is to perform a majority voting. Specifically, if there are more spams in a cluster than ham tweets, then the cluster is labeled as a spam cluster. However, the majority voting approach solely relies on the predicting power of the detectors and may not capture the new spamming patterns in the most recent time window. Moreover, because tweets in a cluster are near-duplicates, their labels assigned by the detectors are mostly the same. For this reason, we also employ a feature-based classifier.

Each cluster is represented with hashtag-based features, content-based features, user-based features, and domain-based features, as listed in Table 5.1, the third column. Many of the features used here are adopted from existing studies [CMP11, FVD⁺14]. Different from tweet classification (features listed in the second column), the cluster-level features represent the collective information obtained from all the tweets in the cluster. The clusters represented in feature-space are classified using a logistic regression classifier.

We consider a cluster to be a confidently labeled cluster if the labels predicted by the feature-based cluster classifier and the majority voting of the tweet labels are the same.

Update Detector Models. After finding the confidently labeled tweets and clusters, the models used by the detectors are updated accordingly including blacklisted domains, labeled clusters, trusted users and tweet classification models. Blacklisted domains are updated by including domains having at least 5 tweets in the last time window and at least 90% of the tweets are confidently labeled as spam tweets. A user having at least 5 tweets and all tweets are confidently label ham tweets is considered as a trusted user. The classification models of the three classifiers are retrained by including the newly labeled confident tweets of the last time window.

By updating the detection models in batch mode, the proposed semi-supervised spam detection framework is capable of capturing new vocabulary and new spamming behaviors, which makes the framework robust and adaptive to deal with the dynamic nature of spamming activities.

5.1.3 Computational Efficiency

All the four spam detectors are computational effective, hence the proposed framework is capable of labeling tweet stream in real-time. More specifically, the first two detectors simply match the tweet's domain and signature to lookup tables. The reliable ham tweet detector labels tweets in constant time by checking user list and spammy words. The most computationally costly part of the detection process is the feature extraction for tweet classifications.

We conduct the experiment on a desktop PC with octa-core Intel processor of 3.70GHz and 16 GB RAM. In our experiments, all the detectors are carried out on a single-core of the processor, except random forest classifier which utilizes all the cores of the processor. Empirically, we found that on average it takes 0.495 ms to label a tweet where more than 50% of the time is used for feature extraction. Note that, our

code is not carefully optimized for a real-time setting. More importantly, efficiency can be further improved by parallelising the detectors. The most costly part of the model update is feature extraction for near-duplicate tweet classification.

5.2 Experiment and Discussion

We used 15 days of data from HSpam14 dataset [SS15] in our experiments. HSpam14 contains 14 million tweets, collected by using the trending topics on Hashtags.org for two months, May and June 2013. In this research, we use 15 days of tweets, May 17 - 31 2013 where each day has more than 35 thousand tweets. The time window for batch mode update is set to be a day. Almost all tweets in HSpam14 are labeled to be spam and ham and the remaining small portion are labeled as unknown for not being able to determine their labels even with manual inspection. Note that, more than 80% of tweets in HSpam14 are labeled automatically and the manually labeled tweets are bias to spams.

We simulate a tweet stream in our experiments. On the first day, the detectors in S³D are trained using the manually labeled tweets and the reliable ham tweets in the HSpam14 dataset (the released HSpam14 dataset contains the detailed labels of the tweets, *i.e.*, on which step a tweet was labeled during dataset construction). These training tweets are utilized to create the initial set of blacklisted domains, labeled clusters, trusted users, labeled tweets and spammy words. There are 48849 spam tweets and 22185 ham tweets. The remaining tweets on the first day and all the tweets of the remaining 14 days are used for testing purpose. Because not all tweets in HSpam14 are manually labeled, to ensure the accuracy of the evaluation in our experiments, we manually label 300 randomly selected tweets from each time window to evaluate the performance of the system.¹ The performance of the spam detection is evaluated using the commonly used metric: Precision, Recall, and F_1 .

¹We have also evaluated the results using the manually labeled tweets in the HSpam14, similar results were obtained.

Table 5.2: Fraction of tweets detected by each detector

Step	Detector	Coverage
1	Blacklisted domain detection	5.55%
2	Near-duplicates detection	6.61%
3	Reliable ham tweets	0.64%
4	Tweet classification	87.20%

Supervised spammer detection on Twitter such as [LCW10, FVD⁺14] focus on spammer detection whereas our work is on tweet-level spam detection. Hence, these spammer detection systems can not be compared with our proposed method. Previous tweet-level spam detection studies used supervised standard classifiers namely, Naïve Bayes, Logistic Regression and Random Forest classifier. We have used these methods to compare the performance of the proposed system. Tweet classification using logistic regression reported in this work is similar to the model proposed on information credibility [CMP11] and also similar to the method reported in [MRA13]. Most of the features described in the information credibility paper except propagation related features are used in the S³D as well. Propagation related features are not available in HSpam14 dataset so those features could not be used. Similarly, Naïve Bayes and Random Forest classifiers are used in [SML⁺13, MRA13, WZLP15] to evaluate the system. Performance of Random Forest classifier is found to be superior among all the other methods [SML⁺13, WZLP15]. These widely used techniques for content based spam tweet detection are used to compare the performance of the proposed system. As reported in [WZLP15] we have also found that uni-gram and bi-gram features are important features to classify tweets.

5.2.1 Result and Discussion

S³D has four detectors as shown in Figure 5.1. Table 5.2 reports the percentage of tweets labeled by each detector. It shows that 5.55% of the tweets are labeled by the blacklisted domain detector and 6.61% of the tweets are labeled by the near-duplicate detector. Reliable ham tweet detector has very low coverage of 0.64%. The low

Figure 5.2: Precision of supervised and semi-supervised approach

coverage is due to the fact that the HSpam14 dataset was collected based on popular hashtags, not on user basis [SS15]. In other words, the dataset does not contain all tweets of any user. Because a trusted user should have at least 5 ham tweets, only a small set of users can be identified as trusted users. Remaining 87.20% of tweets are labeled by the last detector, tweet classifier. Next, we report the spam detection performance of S³D with more focus on the tweet classifier detector.

We now report the performance of the following five methods for spam tweet detection task.

NB This method reports the prediction results of the Naïve Bayes classifier rely on the training data of the first day.

LR This method reports the prediction results of the Logistic Regression classifier rely on the training data of the first day.

RF This method reports the prediction results of the Random Forest classifier rely on the training data of the first day.

Figure 5.3: Recall of supervised and semi-supervised approach

S³D-Update The results of the S³D framework without a batch update. That is, the detectors in the framework fully rely on the training data of the first day.

S³D The results of the proposed S³D framework.

Figures 5.2, 5.3, and 5.4 plot the Precision, Recall and F_1 scores respectively of the five methods. Observe S³D achieves the best F_1 scores. The significant better F_1 scores against S³D-Update over all days shows that semi-supervised approach is suitable for real-time spam detection in Twitter as it learns new spamming patterns continuously. Comparing S³D with the F_1 scores of NB, LR and RF shows that proposed method is superior to standard supervised methods. Observe F_1 scores of S³D is consistent over time compare to other methods. It is observed that precision of RF is best for 3 days but recall is the lowest of all. In contrast, NB has good recall at the expense of lower precision. The results show that proposed S³D method is effective to capture spam tweets effectively.

The sudden rise of the F_1 scores on the 12th day is due to the presence of a large number of relatively easy to detect spam tweets in that time window. In HSpam14 dataset, the tweets were collected by using trending keywords of each day which leads

Figure 5.4: F1-score of supervised and semi-supervised approach

to change in the distribution of words of each day in the dataset. The significant fluctuation of the performance of S^3D may be due to the changing distribution of dataset in each time window. However, as S^3D continuously learns new patterns and vocabulary, its performance is found to be more consistent compared to other methods. More specifically, Figure 5.5 plots the number of blacklisted domains, confidently labeled near-duplicate clusters and trusted users. It shows that S^3D keeps on utilizing new knowledge obtained from earlier labeled tweets and clusters to improve the capability of spam tweets detection. Furthermore, we have also used top 10,000 frequent uni-/bi- and tri-grams computed at the end of each time window to update the model to deal with vocabulary change (see Table 5.1).

In S^3D , we identify confidently labeled tweets and confidently labeled near-duplicate clusters. The confidently labeled tweets and clusters are utilized to learn new models for the detectors. The quality of these confidently labeled tweets and clusters are therefore crucial for the performance of S^3D . Here, we evaluate the quality of these tweets and clusters, plotted in Figure 5.6. The confident clusters are evaluated by manually labeled 47 randomly selected clusters on each day, which is the smallest

Figure 5.5: Accumulated numbers of blacklisted domains, confident clusters and trusted users

number of confidently labeled clusters produced over the 15 days. Figure 5.6 shows that the precision of confident clusters is almost perfect for both spam and ham clusters. The figure also shows that the precision of confidently labeled spam and ham tweets are consistently above 95%. Adding such clusters and tweets in the training process makes S³D capable of capturing emerging spamming activities as well as new vocabulary.

5.2.2 Feature Analysis

There are four types of features used to represent tweet and cluster for classification (see Table 5.1). In our experiments, we observe that normalization of features gives better performance than without normalization. Because users' followers, followees, and total tweets exhibit power law distributions. The features derived from these values are normalized based on percentile. Features such as length of tweet in characters and words show normal distribution which are normalized by the maximum value. Based on the Gini impurity score, we identify the top-15 most effective features for tweet classification and cluster classification respectively. These features are highlighted in Table 5.1 in “(x)” format, where x is the top ranking position.

Figure 5.6: Precision of confidently labeled clusters/tweets

It is observed that 10 out of the top-15 most effective features are vocabulary-based features for tweet classification whereas in the case of cluster classification only 3 out of the top-15 features are vocabulary-based features. Meta-data of a tweet contains information only about the single tweet which is comparatively less informative. In contrast, a cluster contains a number of tweets, hence meta-data based features represent the collective information of tweets in the cluster and are comparatively more informative. For example, if there is a tweet from a user whose account creation date is known and has very few followers and followees, it is hard to determine that tweet posted by this users is ham or spam. In contrast, if there is a group of users whose accounts are created around the same time and all having very few number of followers and followees, and posting near-duplicate tweets, then the tweets in this cluster are likely to be spam.

Table 5.3 lists the top-15 vocabulary-based features. We have used uni-gram, bi-gram and tri-gram vocabulary as features. However, only uni-gram and bi-gram vocabulary appear in the top ranked list. One possible reason for tri-gram features not in the list may be due to the sparsity of the tri-gram vocabulary in the dataset. It is interesting to note that most of the top words based on Gini-impurity score are the same as the list of hashtags having the highest *spammy-index* reported in [SS15].

Table 5.3: Top-15 most effective vocabulary features

Tweet	follow; follow back; back; ipad ipadgames; please follow; please; followers; retweet; tfbjp; teamfollowback; ipad; follow me; collected; followback; gameinsight
Cluster	follow; love; back; followers; follow me; follow back; retweet; openfollow; team-followback; want; someone; win; please; 500aday; gain

5.3 Conclusion

In this chapter, we proposed a semi-supervised spam detection framework, named S³D. S³D utilizes four light-weight detectors to detect spam tweets on real-time basis and update the models periodically in batch mode. The framework mainly consists of two main modules: (i) four light-weight detectors in the spam tweet detection module for detecting spam tweets in real time, and (ii) updating module to periodically update the detection models based on the confidently labeled tweets from the previous time window. Utilizing the confidently labeled tweets from previous time windows the model continuously learn the new feature distribution and new spamming activities. Further, updating vocabulary makes the framework robust for new words. The experiment results demonstrate the effectiveness of semi-supervised approach in our spam detection framework.

In our experiment, we found that confidently labeled clusters and tweets make the system effective in capturing new spamming patterns. Due to the limited user information in the dataset, we have used a simple technique to deal with user-level spam detection. However, the user-level spam detection can be easily incorporated into S³D, which is part of our future work. As spam is a serious issue, it is interesting to understand the effect of such spam on applications. Hence in the next chapter, as a case study, we analyze the impact of spam on hashtag recommendation.

Chapter 6

Effect of Spam on Hashtag Recommendation

In this chapter, we introduce seven hashtag recommendation methods and utilize those systems for a case study to evaluate the effect of spam on hashtag recommendation. We particularly focused on tweets containing hyperlinks to an external document. In this study, we call a tweet containing one or more hyperlinks to external documents a *hyperlinked tweet*.

Compared with many non-hyperlinked tweets (*e.g.*, those about daily activities), a user is more willing to retweet a hyperlinked tweet with a larger audience. It is reported that many trending topics on Twitter are from headline news and many of such news pieces are shared through tweets with hyperlinks to web pages [HLW⁺12, KLPM10]. Hence, the presence of a link in a tweet is a strong indication of the tweet being more informative [DJQ⁺10]. However, links in Twitter are mostly encoded in short URLs which neither provide any hint about the information source nor the content of the links. Hence, readers do not have any clue about a link unless the tweet provides a good description. Such description often relies on the right context provided by hashtags. For example, hashtags like #SIGIR and #eurocup provide the right context to understand both the tweet and the link. Even though hashtags provide contextual information about tweets, there is a significant fraction of the tweets

that do not contain any hashtags. One of the reasons could be users may not be aware of the relevant hashtags for their tweets due to lack of exposure to the relevant hashtags [RMK11]. Hence, hashtag recommendation system makes hashtag selection process easier and may also motivate to use more hashtags. Moreover, HSpam14 dataset is collected using the trending hashtags which is suitable for hashtag related studies. Hence, as a case study of the effect of spam on applications we use hashtag recommendation for hyperlinked tweets.

Figure 6.1 illustrates a conceptual model consisting of *hashtags*, *hyperlinked tweets*, and the *linked documents*. An edge exists between a hashtag and a hyperlinked tweet if the tweet is annotated with the hashtag (*e.g.*, h_1 and t_1). Similarly, an edge between a tweet and a document exists if the document is linked from the tweet (*e.g.*, t_1 and d_1). We argue that through hyperlinked tweets, the functions of hashtags can be extended to the content of the linked documents. That is, the hashtags convey additional contextual information to the linked documents, facilitate the topical categorization of these documents, and serve as a medium to propagate the documents to more readers on Twitter. Further, hashtags are commonly used as queries to search tweets about a topic or specific event [TRM11]. Having appropriate hashtags in a hyperlinked tweets make the tweets more informative and influential. Hence, Hashtag recommendation is important on Twitter but at the same time, it is an inherently difficult task. Further, due to the increasing spamming activities it is becoming more difficult.

Increasing spam activities on Twitter has an adverse effect on different applications rely on Twitter dataset. Hence, it is important to understand the impact of spam tweets on applications(*e.g.*, tweet search, tweet classification, and event detection). The presence of spam tweets in a dataset may affect the choices of feature selection, algorithm formulation, and system evaluation for many applications. For example, spammers promoting #gameinsight exploit popular hashtags such as #iphone #android in such an extent that the most co-occurring hashtag with these popular hashtags is #gameinsight. Hence, the features derived from such co-occurrence information is severely skewed by such malicious activities. Nevertheless, most existing studies on

Figure 6.1: Hashtag, hyperlinked tweet, and linked document

Twitter have not considered the impact of spam tweets in their data collections. To this end, as a case study we analyse how the presence of spam on a dataset effect hashtag recommendation for hyperlinked tweets.

6.1 Hashtag Recommendation

Given a hyperlinked tweet t containing link ℓ to a web page, our task is to recommend a list of hashtags that are most relevant to the tweet.¹ There are two cosine similarity based hashtag recommendations and another based on frequently used hashtags on the domain. Named entities appearing on a web page are important topic indicators of the page. We also observe that some of the named entities are directly adopted as hashtags in the hyperlinked tweets. Hence, Random Walk with Restart (RWR) model [PYFD04] and Language Translation (LT) model [LCS11] utilized named entities on web page to recommend hashtags. Last two learning to rank based methods leverage results of first five systems to get better recommendation results. We have analyzed and compared the results of all the methods; however, the main focus of this chapter is to study the effect of spam on hashtag recommendation.

¹In our data collection of 24 million tweets, more than 96% of hyperlinked tweets each contains only one link. For easy presentation, we assume each tweet contains one link, although the proposed method can be easily extended to handle multiple links.

Figure 6.2: Entity-hashtag graph

6.1.1 Hashtags from Similar Tweets

Based on the assumption that topically similar tweets are more likely to be annotated by similar hashtags, we search for hashtags by using the similar tweets to the given tweet t . Cosine similarity and TFIDF weighting scheme are adopted to search similar tweets. The top most voted hashtags are selected by using the top 50 most similar tweets. This hashtag recommendation approach is similar to the hashtag recommendation method proposed in [KHLZ12] (not considering the hashtags by similar users).

6.1.2 Hashtags from Similar Web Pages

Based on the discussion that the functions of hashtags can be extended to the linked documents, we also assume that similar web pages are annotated by similar hashtags. Two web pages discussing a particular people or event is likely to use same sets of hashtags. Hence, we search for candidate hashtags by using the similar web pages. Cosine similarity and TFIDF weighting scheme are adopted to search similar web pages. The top most voted hashtags are selected by using the top 50 most similar web pages.

6.1.3 Hashtags based on the Domain of the Link

In our dataset, we observe that hashtags often reflect the topics of the linked website. Let ℓ_d be the domain of hyperlink ℓ . We select the top hashtags by $P(h|\ell_d)$, which is the probability that h is used to annotate a tweet containing links from domain ℓ_d . It is based on the intuition that while sharing links from particular domain, people are likely to use hashtags from a small pool of hashtags that characterize the domain. For example, hashtags `#news`, `#cnn`, and `#breaking` are likely to be used for links from domain *edition.cnn.com*.

6.1.4 Random Walk with Restart (RWR) model

We construct an entity-hashtag graph where there are two sets of nodes N_e and N_h (see Figure 6.2). $N_e = \{e_i\}$ is the set of nodes corresponding to the named entities in the web page linked from tweet t . $N_h = \{h_j\}$ is the set of nodes corresponds to a subset of hashtags in the data collection. A hashtag is included in this graph if it can be reached from any of the named entity node $e_i \in N_e$ through the edges. There are three kinds of directed edges: $e_i \rightarrow h_j$, $h_j \rightarrow e_i$, and $h_j \rightarrow h_k$, weighed by $P(h_j|e_i)$, $P(e_i|h_j)$, and $P(h_k|h_j)$, respectively.² The first two conditional probabilities are computed by the number of times a hashtag h_j is used to annotate a tweet linking to a document containing a named entity e_i , with respect to the frequencies of e_i and h_j respectively. The last conditional probability is the asymmetric hashtag co-occurrence. Only the edges with positive weights are included in the graph.

To apply RWR model, let $\vec{r}$ be the restart vector of dimensionality $|N_e| + |N_h|$, where $|\cdot|$ denotes the cardinality of a set. The entries corresponding to named entities in $\vec{r}$ are set to 1 and the entries corresponding to hashtags are set to 0. Let $\mathbf{P}$ be the transition matrix derived from the weights computed above. Let α be the restart probability. The steady state value of $\vec{v}$ is computed using the following formula. The

²In our experiments, we have also evaluated the graph with edges between two named entities weighted by $P(e_k|e_i)$. This version of the graph leads to poorer recommendation accuracy.

top hashtags with highest corresponding values in $\vec{v}$ are recommended.

$$\vec{v} = (1 - \alpha)\vec{v}\mathbf{P} + \alpha\vec{r}$$

6.1.5 Language Translation (LT) model

In this model, we consider the named entities and hashtags as descriptions of the same content in two different languages. Again, let N_e be the set of named entities appearing in the document linked from tweet t , the score of a hashtag h_j is computed by:

$$Score(h_j) = \sum_{e_i \in N_e} P(h_j|e_i)$$

where, $P(h_j|e_i)$ is the same conditional probability as in the RWR model. Similarly, from this model, hashtags with highest scores are recommended.

6.1.6 Pairwise Learning to Rank

Hashtags recommended by above 5 hashtag recommendation methods can be used as candidate hashtags for learning to rank based hashtag recommendation system. We have used top 20 hashtags from each method as candidate hashtags. We represent each candidate hashtag as a feature vector and use pairwise learning to rank method to find the top ranked hashtags from the candidate set. Given a tweet t , let h_i^+ be a positive hashtag that is selected as a candidate hashtag and is used to annotate tweet t ; let h^- be a negative hashtag that is selected as a candidate hashtag but is not used to annotate t . Then the pair $\langle h^+, h^- \rangle$ is a positive instance and $\langle h^-, h^+ \rangle$ is a negative instance in learning the model. Each pair is represented by the vector difference between the feature vectors of the two hashtags, to be detailed shortly. In the test phase, each candidate hashtag is paired with the rest of the candidate hashtags and then classified by using the learned model. Let H_c be the set of candidate hashtags.

The recommendation score of candidate hashtag h_i is: $f(h_i) = \sum_{h_j \in H_c, h_i \neq h_j} I(h_i, h_j)$, where $I(h_i, h_j) = 1$ if $\langle h_i, h_j \rangle$ is classified as positive and 0 otherwise.

Features. We use two sets of binary features (*i.e.*, the value of each feature is either 1 or 0).³ The first set of 5 features describes the schemes a candidate hashtag is selected. Each of the 5 dimensions is set to 1 if the candidate hashtag is selected by similar tweet, similar web page, domain of the link, named entities with RWR, and named entities by LT, respectively. The second set of 4 features utilizes Wikipedia, category list, and domain related information, detailed below.

1. *Wikipedia.* This feature is set to 1 if the hashtag appears as an anchor text in a Wikipedia entry. The list of anchor texts is parsed from the English Wikipedia dump on January 30, 2010.
2. *Is Category.* This feature is set to 1 if the hashtag is one of the top-level categories in Yahoo! hierarchy, or it matches a category in any of the four web sites: bbc, cnn, nytimes and reddit. After duplicate removal, there are 75 categories including sports, technology, business, movie, jobs etc.
3. *Is Popular Domain.* This feature is set to 1 if the hashtag matches one of the top 5000 most popular domains in our dataset.
4. *Matches Domain.* This feature is set to 1 if the hashtag matches the domain of the link contained in the tweet.

The second set of 4 features is proposed based on the observation that hashtags are often used to topically categorize tweet content. We also observe that many websites themselves are topically indicative (*e.g.*, Yelp for business review, CNN for news).

³We have also evaluated the non-binary weighting scheme of the same set of features (*e.g.*, the feature values are normalized to $[0, 1]$) but observed poorer performance.

Table 6.1: The 7 methods in evaluation

Method name	Description
SimilarTweet	Most frequently used hashtags in similar tweets
SimilarPage	Most frequently used hashtags in similar web pages
DomainFreqTag	Most frequently used hashtags for the given domain
NamedEntity-RWR	Hashtags selected by named entities using RWR
NamedEntity-LT	Hashtags selected by named entities using LT
RankSVM	RankedSVM using the first set of 5 features
RankSVM++	RankedSVM using the all (two sets of) features

6.2 Dataset

In the HSpam14 dataset there are 1.07 million hyperlinked tweets ignoring the dead-links and links requiring authentication (*e.g.*, Facebook links). Among 1.07 million hyperlinked tweets, 0.95 million linked pages have at least one named entity in the page based on StanfordNER.⁴ Out of the 0.95 million tweets, 0.220 million tweets have at least one hashtag. In this study, we use these 0.220 million hyperlinked tweets. Among them, 22 thousand are spam tweets and 198 thousand tweets are ham tweets.

After processing the linked Web pages (*e.g.*, parsing out the textual content and extracting the named entities⁵), we observe that 12.65% of tweets contain named entities from their linked Web pages as hashtags. This accounts only the single-word hashtags without segmentation. Besides named entities, broad categories such as sports, news, business are also widely used hashtags. Moreover, domain names such as NBA, TED, CNN are found to be used as hashtags frequently. Among hyperlinked tweets, more than 96% of tweets each contain only one link.

6.3 Experiment

In this section, we evaluate the performance of seven hashtag recommendation systems discussed in the previous section. We perform hashtag recommendation utilizing all

⁴<http://nlp.stanford.edu/software/CRF-NER.shtml>

⁵Stanford NER: <http://nlp.stanford.edu/software/CRF-NER.shtml>

220 thousand tweets. These tweets contain both ham and spam tweets. First five methods are baseline methods and learning to rank based methods are proposed in [SS14]. Even though the proposed system is significantly better than the baseline systems, the main focus of this chapter is to analyze the effect of spam on hashtag recommendation rather than comparing the system performance.

Table 6.1 shows the list of the hashtag recommendation systems. First five methods are independent of each other whereas for RankSVM based method utilized the first five methods for candidate hashtag selection. RankSVM is a pairwise learning to rank approach which creates the training example based on the order of the pair of candidate hashtags. Representing hashtag pair as a positive and negative example is vital step to learn RankSVM model. For example, `#h1` and `#h2` are two hashtags recommended based on SimilarTweet approach and `#h1` matches with ground truth hashtag hence its correct recommendation whereas `#h2` does not match with ground truth hashtag hence it is an incorrect recommendation. In this example, we can create both positive and negative instances utilizing these two hashtag pairs. Vector difference of `#h1` and `#h2` is a positive instance for RankSVM model whereas vector difference between `#h2` and `#h1` is a negative example. However, we can utilize only one because of symmetric nature of positive and negative examples. Using this procedure, we created training and test instances for RankSVM based learning to rank model. RankSVM uses 5 binary features describing the schemes a candidate hashtag is selected. RankSVM++ uses all 9 binary features listed in Section 6.1.6. Both methods were implemented using the `liblinear` library.⁶

To learn the RankSVM models, we randomly select 40% of the tweets as training and the remaining 60% as test from spam and ham tweets, respectively.⁷ In candidate hashtag selection for both training and testing, we limit the search for similar tweets (resp. web pages) posted one day before the currently processing tweet to simulate inaccessibility of future data in reality.

⁶<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

⁷Results of training/test datasets created by random selection are found to be consistent with the results from partitioning of tweets by time.

Figure 6.3: Hit rate of the 7 methods

Evaluation Metric. We use three metric to evaluate hashtag recommendation accuracy: *Precision@k*, *Recall@k*, and *HitRate@k* $k=\{1, 5, 10\}$ is the number of top-ranked recommended hashtags. Let H_k be the set of top- k recommended hashtags and H_g be the set of ground-truth hashtags of a tweet t . ***Precision@k*** for tweet t is $|H_k \cap H_g|/k$; ***Recall@k*** for tweet t is $|H_k \cap H_g|/|H_g|$; and ***HitRate@k*** for t is 1 if $|H_k \cap H_g| \geq 1$ and 0 otherwise. The values reported for each method are the averaged values over the test tweets.

6.4 Method Comparison

In this section, we evaluate the performance of seven hashtag recommendation systems discussed in previous section. Figure 6.3, 6.4, 6.5 report *Precision@k*, *Recall@k*, and *HitRate@k* of the five baseline methods and the two RankSVM methods for $k = \{1, 5, 10\}$.

Figure 6.4: Precision of the 7 methods

The two RankSVM methods outperform all the five baseline methods on all three evaluation metrics for all k values. The results clearly show the effectiveness of aggregating candidate hashtags selected by exploiting multiple sources. When $k = 1$, RankSVM is slightly better than RankSVM++ on all three metrics. However, RankSVM++ outperforms RankSVM when k is 5 or more, suggesting the effectiveness of the 4 additional features in the learning to rank model. It is observed that learning to rank based model is suitable to combine multiple competing systems and adding other important features to make robust hashtag recommendation system.

Among the five baseline methods, SimilarPage performs slightly better than SimilarTweet, followed by DomainFreqTag, particularly when k is greater than 1 on three evaluation matrices. This suggests that similar web pages are more reliable in recommending hashtags, and the domain of the hyperlink is indeed an important piece

Figure 6.5: Recall of the 7 methods

of information for hashtag recommendation. The results also validate that similar content is annotated by similar hashtags.

Two named entity based approaches namely NamedEntity-RWR and NamedEntity-LR have the lowest performance across all performance measures. Further, NamedEntity-RWR is computationally expensive compared to NamedEntity-LR. However, the result of NamedEntity-RWR is consistently inferior to NamedEntity-LR. It shows that co-occurrence based approach is better than neighborhood-based approach for hashtag recommendation on Twitter. Even though the performance of these systems are inferior to other systems, these two methods complement other remaining method to obtain better hashtag recommendation in learning to rank system.

Analysis of the data shows that a tweet is likely to have fewer than 3 hashtags (see Figure 3.3) hence precision of the system is likely to degrade as we increase k . As expected, we observed that when $k = 5$ or 10 there is a large performance degradation

based on precision. However, recall and hit rate should not decrease as we increase k . We observed the behavior in our dataset, when $k = 10$, RankSVM++ achieves *Recall@10* of 0.53 and *HitRate@10* of 0.63.

In our proposed solution, we select candidate hashtags by exploiting the content of the tweets, the linked documents, and the domain of the hyperlink. Using the learning to rank models, we show that the aggregation and ranking of candidate hashtags achieves promising hashtag recommendation accuracy, evaluated by precision, recall, and hit rate.

6.5 A Case Study

There is a wide range of applications such as hashtag recommendation, sentiment analysis, event detection, election prediction to name a few. These applications assume that all the data on social media are generated by legitimate users. However, such assumption is fundamentally problematic and may adversely affect the system performance and may give misleading results. Hence, it is crucial to understand the effect of spam of applications. To this end, as a case study we perform experiment to understand the effect of spam on hashtag recommendation.

In this section, we perform the experiment on three datasets derived from the dataset used in the previous section. The dataset which contains only ham tweets is named as *ham_{only}*. This dataset consists of 22 thousand hyperlinked ham tweets which is the collection of all the ham tweets in the dataset used in the previous section. Similarly, the dataset which contains only spam tweets is named as *spam_{only}*. There are 198 thousand tweets in *spam_{only}* dataset which is the collection of all the spam tweets in the dataset used in the previous section. All the tweets from *spam_{only}* and *ham_{only}* dataset are merged to create the *merged* dataset which contains of 220 thousand tweets. The *merged* dataset is same as the dataset we used in the previous section.

Figure 6.6: Hit rate of the 7 methods on spam, ham and merged datasets

6.5.1 Result and Discussion

In this section, the analysis is mainly focused on the effect of spam on the hashtag recommendation task. Figures 6.6, 6.7, and 6.8 report $HitRate@k$, $Precision@k$ and $Recall@k$ respectively of all the 7 methods. It is observed that both RankSVM and RankSVM++ are superior to rest of the methods across all the dataset which shows that learning to rank method is effective in combining different ranking methods. However, instead of comparing methods we focus on the analysis of the effect of spam on these systems.

Figure 6.7: Precision of the 7 methods on spam, ham and merged datasets

In all the evaluation measure, the performance of all the recommendation systems are best for the *spam_{only}* dataset. Further, *HitRate@1*, *HitRate@5* and *HitRate@10* of RankSVM++, the best performing method, are at least 15% higher on spam tweets than ham tweets. Similar observations hold for precision and recall. This shows that hashtag recommendation is much easier for spam tweets than ham. Spammers heavily exploit popular hashtags and presence of popular hashtags could be one of the reasons for performance being superior for the *spam_{only}* dataset.

Figure 6.8: Recall of the 7 methods on spam, ham and merged datasets

Similarly, simple methods (SimilarPage and DomainFreqTag) achieve very good recommendation accuracy for spam tweets, probably due to the fact that many spam tweets are posted to promote web pages from a limited number of domains. For instance, a large number of tweets linking to game-insight.com contain 3 hashtags from a small pool of hashtags (*e.g.*, android, androidgame, ipad, ipadgames, iphone, iphonegames). Another simple method, SimilarTweet, also achieves good results due to the presence of near-duplicate tweets in spam. The presence of a large number of near-duplicate spam tweets badly affects any application. In order to address

these kinds of promotional activities, near-duplicate focus spam detection may help to capture such tweets effectively.

Further, the presence of spam on dataset may provide misleading information. In our dataset, there are less than 10% of spam tweets; however, the accuracy on the merged dataset are at least 2% higher than ham tweets on all measures. This shows that without removing spam in a data collection, the results obtained for a hashtag recommendation method may be (misleadingly) better than its actual performance in reality.

RankSVM++ is the best performing method on all datasets by all evaluation measures. However, the percentage of improvement by using RankSVM++ over simple methods (*e.g.*, SimilarPage, DomainFreqTag) on spam dataset is much smaller than that on ham dataset. That is, features/methods that are effective on ham may not necessarily be effective on spam tweets.

Different activities of spammers on Twitter have a direct impact on applications based Twitter dataset. Due to the promotional tweets, there are a large number of spammy near-duplicate tweets. Hence, if we recommend hashtags correctly for one of the near-duplicate tweets then for the rest of the near-duplicate tweets recommendation is likely to be true. Similarly, aggressive use of the popular hashtags by spammers makes hashtag recommendation simpler for spam dataset. Further, it is observed that presence of spam content can skew the feature selection process.

6.6 Conclusion

Spam is a prominent problem on Twitter, which has adversely affected many applications. Hashtag recommendation is an important application which helps users to easily identify appropriate hashtags for their tweets. Hence, as a case study we used hashtag recommendation system to analyze the effect of spam on an application. To this end, we introduced seven hashtag recommendation approaches. We analysed the performance of these recommender system for three datasets namely *spam_{only}*, *ham_{only}*

and *merged* datasets. Analysis of the result on three datasets shows that presence of spam has an adverse effect on hashtag recommendation on Twitter.

Our experiment shows that spam on Twitter skews the performance evaluation. It is observed that simple methods such as DomainFreqTag give very good accuracy for hashtag recommendation for spam tweets. Due to the presence of near-duplicate tweets, methods like similar tweets and similar web pages also achieve good results. In the literature, similar item based recommendation has been considered as a strong baseline, which may be heavily affected by the presence of spam. We have also observed that features and methods which are effective for ham tweets may not be equally effective for spam. Hence, the presence of spam in the dataset may affect feature selection process in some applications. It is necessary to perform spam filtering before conducting any analysis or evaluation on Twitter dataset.

Analyses of effects of spam on different applications such as sentiment analysis, event detection may provide better insights. Further, there are different types of spammers such as social bot, crowdturfers, link farmers to name a few. Hence, analyses of the effect of different types of spammers on microblogging platform is a potential direction to explore.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Microblogging sites act as platforms for exchanging ideas and sharing information and Twitter is by far the most popular microblogging site. Hence, without the loss of generality, Twitter dataset is utilized for all the studies in the dissertation. Besides being popular, information propagation is very effective on Twitter and click-through rate of URLs shared on Twitter is orders of magnitude more than in email. Due to these reasons, there are increasing spamming activities on Twitter which has deteriorated the user experience on Twitter. Further, spammers exploit popular hashtags to promote their content. Hence, data is collected using trending hashtags of each day provided by [hashtags.org](https://trends.hashtags.org/) to perform hashtag oriented spam research. In order to address these issues on Twitter, we need to detect spam tweet on time which can be done using fine-grained spam detection approach. In this dissertation, we take a fine-grained approach to deal with the spam on Twitter. Experimental results show that fine-grained approach is effective and may complement other social graph-based and user-based spam detection on Twitter.

One of the challenges on fine-grained spam detection is the lack of benchmark dataset. Further, manual labeling of a large dataset is an expensive and tedious task. To this end, in Chapter 3 we propose an effective way to label millions of tweets with

minimum human effort. We utilize a hashing based efficient unsupervised approach to create near-duplicate clusters. Analysis of the clusters shows that near-duplicate clusters are of high quality where intra-cluster similarity is more than 94% based on cosine similarity as well as Jacquard similarity whereas inter-cluster similarity is less than 5%. In order to reduce the human effort, a small set of the clusters are labeled manually and remaining large number of the near-duplicate clusters are labeled by using a modified nearest neighbor approach. Then, we detected reliable ham tweets based on learning from positive only examples. Finally, remaining tweets are labeled using expectation maximization algorithm. Manual evaluation of the automatically labeled tweets shows that annotations of tweets are of high quality which can be utilized for studies of hashtag oriented spam detection on Twitter.

In Chapter 4, we perform detailed analysis of the HSpam14 dataset prepared in Chapter 3 to understand spamming activities on Twitter. The analysis is focused on similarities and differences between spam and ham tweets/users in various aspects based on hashtag, tweet content, and user profile. From the analysis, it is observed that spam tweets contain more hashtags compared to ham tweets. Position and orthography of hashtags used in spam and ham tweets are also found to be significantly different. Our analysis shows that spammers use popular hashtags to promote their content and they also emphasize tweets by using exclamation sign and capitalized words. Further, spammers post same content multiple times and distribute the spamming activities among many accounts. It is observed that profile information of spammers is significantly different from profile information of legitimate users. Comparing with previous studies, it is observed that spammers have changed their strategy, from having more followers to no or fewer followers, to remain undetected. The detailed analysis of the dataset and comparisons with previous studies show that there are significant differences between spam and ham tweets and users posting them, and such tweet can be detected utilizing the insights gained from this study.

In Chapter 5, we utilized the HSpam14 dataset detailed in Chapter 3 and utilized the insights from Chapter 4 to design robust real-time spam detection framework. We

utilized a semi-supervised approach which is found to be effective to capture changing spamming activities. S³D framework consists of four light-weight detectors which can identify spam on real-time. In the four detectors, near-duplicated cluster-based detector and reliable ham detector are based on our understanding of HSpam14 dataset. The classifiers in the framework utilize discriminative features identified in our analysis as well as in previous studies. It is observed that content-based approach is effective for real-time spam detection. Further, all the light-weight detectors are updated over time due to which the system can capture changing pattern of spamming activities. Results of the framework show that semi-supervised approach is suitable to detect spam on tweet stream.

In Chapter 6, we study the effect of spam on applications and as a case study, we analyzed the effect of spam on hashtag recommendation for hyperlinked tweet. It is observed that features and methods which are effective for spam dataset may not be effective for ham dataset. It is also observed that evaluation of any system based on a dataset containing spammy content may give misleading information. Hence, it is necessary to perform spam filtering before conducting any analysis or evaluation on Twitter dataset.

In summary, our research is mainly focused on fine-grained spam detection on microblog. It is observed that with the better understanding of data and utilizing both supervised and unsupervised approaches, millions of tweets can be labeled with good accuracy. Similarly, analyzing and comparing findings with previous studies give valuable insight which is useful to design robust spam detection system. We utilized the finding of the analysis to design a real-time semi-supervised spam detection framework. The analysis shows that presence of spam on a dataset have adverse effects on applications. Hence, it is crucial to perform spam filtering before utilizing a dataset for any application.

7.2 Future Work

Spam detection on Twitter is a challenging task and removing them is crucial to improve user experience on Twitter. Hence, a number of spam detection techniques have been proposed based on social graph, user profile, and tweet content. However, spam is still a problem on Twitter which can be better addressed by unifying the all the complementary spam detection techniques. Based on our studies on spam detection and analysis research, we recommend potential research directions.

The proposed HSpam14 labeling process shows that million of tweets can be labeled efficiently and effectively. Despite accurate labeling of tweets in HSpam14 dataset, there are potential approaches to be explored which may improve label accuracy. Active learning based approach could be helpful while labeling the clusters. Priority of labeling the clusters can be set based on the cluster-sized and cluster neighborhood. Similarly, discriminative features based on content, hashtags, users and social graph can be exploited to improve the labeling accuracy.

In our study, analysis of the users is mainly based on the profile information of the users due to the limitation of HSpam14 dataset. Hence, it would be interesting to analyze the behavior of a user utilizing all the tweets of the user. Utilizing all the tweets of a user, near-duplicate based approach can be exploited to identify content promoters. For example, a user whose majority of historical tweets are near-duplicate tweets is likely to be a spammer compared to another user whose historical tweets have very less overlap. In this context, it would be interesting to analyze users based on historical tweets of the users. Similarly, malicious domain detection based on near-duplicate clusters could be the interesting direction for future studies.

Our study shows that spammers remain disconnected from a social graph to remain undetected spam detection system. However, new users may have fewer followers and followees hence users who are disconnected from social graph may not necessarily be spammers. It is observed that spammers heavily exploit this trick to evade spam

detection system. Hence, focused spam detection study for users having a few followers and followees is a direction for further exploration.

Further, it is observed that spam tweets have adverse effects on applications based on Twitter dataset. There are different types of spammers like promoter, social bots, crowdturfers. Hence, it would be insightful to analyze the effect of each type of spammer on different applications. It would be interesting to perform comparative studies of the effect on various types of spammers on applications.

Further, we would like to explore a holistic approach for spam detection by utilizing social graph, user profile, posting behavior and tweet content. Even though a tweet is a short and unstructured text, tweet can be better understood by utilizing all the available information. In the future study, we would like to work on the unified framework that incorporates diverse information for effective spam detection on Twitter.

List of Publications

Chapters 3 - 6 of this dissertation are adapted based on the following publications during my PhD study.

- **Surendra Sedhai**, Aixin Sun. An Analysis of 14 Million Tweets on Hashtag-Oriented Spamming. *Submitted to Journal of the Association for Information Science and Technology*. (**Under Review**).
- **Surendra Sedhai**, Aixin Sun. Semi-Supervised Spam Detection in Twitter Stream. *Submitted to IEEE Transactions on Computational Social Systems*. (**Under Review**).
- **Surendra Sedhai**, Aixin Sun. Effect of Spam on Hashtag Recommendation for Tweets. *Proceedings of the 25th International Conference Companion on World Wide Web (WWW 2016)*. Montreal, Canada. April, 2016: 97-98, DOI: 10.1145/2872518.2889404
- **Surendra Sedhai**, Aixin Sun. HSpam14: A Collection of 14 Million Tweets for Hashtag-Oriented Spam Research. *Proceedings of the 38th International Conference on Research & Development in Information Retrieval (SIGIR 2015)*. Santiago, Chile. August, 2015: 223-232, DOI: 10.1145/2766462.2767701
- **Surendra Sedhai**, Aixin Sun. Hashtag recommendation for hyperlinked tweets. *Proceedings of the 37th International Conference on Research & Development in Information Retrieval (SIGIR 2014)*. Gold Coast, Queensland, Australia. August, 2014: 831-834, DOI: 10.1145/2600428.2609452

Bibliography

- [AK15] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, February 2015.
- [AN07] Morgan Ames and Mor Naaman. Why we tag: Motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 971–980, 2007.
- [ANC10] S. Abu-Nimeh and T. Chen. Proliferation and detection of blog spam. *IEEE Security Privacy*, 8(5):42–47, 2010.
- [APK⁺11] Demetris Antoniadis, Iasonas Polakis, Georgios Kontaxis, Elias Athanassopoulos, Sotiris Ioannidis, Evangelos P. Markatos, and Thomas Karagiannis. We.b: The web of short urls. In *Proceedings of the 20th International Conference on World Wide Web*, pages 715–724, 2011.
- [ARY⁺13] Amit A. Amleshwaram, A. L. Narasimha Reddy, Sandeep Yadav, Guofei Gu, and Chao Yang. Cats: Characterizing automation of twitter spammers. In *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–10, 2013.
- [ATS12] Pramod Anantharam, Krishnaprasad Thirunarayan, and Amit Sheth. Topical anomaly detection from twitter stream. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 11–14, 2012.

- [AXV⁺11] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38, 2011.
- [BB08] Enrico Blanzieri and Anton Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1):63–92, 2008.
- [BKM⁺00] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *The International Journal of Computer and Telecommunications Networking*, 33:309–320, June 2000.
- [BMRA10] Fabrcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virglio Almeida. Detecting spammers on twitter. In *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, 2010.
- [BPPM93] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993.
- [BRA⁺09] Fabrício Benevenuto, Tiago Rodrigues, Virgílio Almeida, Jussara Almeida, and Marcos Gonçalves. Detecting spammers and content promoters in online video social networks. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 620–627, New York, NY, USA, 2009.
- [Bro97] A. Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences 1997*, pages 21–29, 1997.
- [BXG⁺13] Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. Copycatch: Stopping group attacks by spot-

- ting lockstep behavior in social networks. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 119–130, 2013.
- [CABK11] Sidharth Chhabra, Anupama Aggarwal, Fabrício Benevenuto, and Ponurangam Kumaraguru. Phi.sh/\$ocial: the phishing landscape through short urls. In *The 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, CEAS 2011, Perth, Australia, September 1-2, 2011, Proceedings*, pages 92–101, 2011.
- [CC15] Cheng Cao and James Caverlee. Detecting spam urls in social media via behavioral analysis. In *Proceedings of the 37th European Conference on IR Research*, pages 703–714, 2015.
- [CCAB12] Giovanni Comarella, Mark Crovella, Virgilio Almeida, and Fabricio Benevenuto. Understanding factors that affect response rates in twitter. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, pages 123–132, 2012.
- [CFGJ02] Abdur Chowdhury, Ophir Frieder, David Grossman, and Mary Catherine McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems (TOIS)*, 20(2):171–191, April 2002.
- [CGWJ10] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on twitter: Human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 21–30, New York, NY, USA, 2010.
- [CGWJ12] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824, 2012.
- [CKOF09] Deepak Chinavle, Pranam Kolari, Tim Oates, and Tim Finin. Ensembles in adversarial classification for spam. In *Proceedings of the 18th ACM*

- Conference on Information and Knowledge Management*, pages 2015–2018, 2009.
- [CMA⁺12] Evandro Cunha, Gabriel Magno, Virgilio Almeida, Marcos Andre Gonçalves, and Fabricio Benevenuto. A gender based study of tagging behavior in twitter. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, pages 323–324, 2012.
- [CMP11] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 675–684, New York, NY, USA, 2011.
- [Cor08] Gordon V. Cormack. Email spam filtering: A systematic review. *Foundations and Trends in Information Retrieval*, 1(4):335–455, April 2008.
- [CTH⁺09] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: A real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 48:1–48:9, 2009.
- [CWW12] Zi Chu, Indra Widjaja, and Haining Wang. Detecting social spam campaigns on twitter. In *Proceedings of the 10th International Conference on Applied Cryptography and Network Security*, pages 455–472, Berlin, Heidelberg, 2012.
- [CYYP14] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 477–488, 2014.
- [CZX⁺15] Chao Chen, Jun Zhang, Yi Xie, Yang Xiang, Wanlei Zhou, Mohammad Mehedi Hassan, Abdulhameed Alelaiwi, and Majed A. Alrubaian. A performance evaluation of machine learning-based streaming spam

- tweets detection. *IEEE Transactions on Computational Social Systems*, 2(3):65–76, 2015.
- [DJQ⁺10] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303, 2010.
- [DLSL09] Bertrand De Longueville, Robin S. Smith, and Gianluca Luraschi. ”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 *Proceedings of the 2009 International Workshop on Location Based Social Networks*, pages 73–80, 2009.
- [DQZH13] Zhuoye Ding, Xipeng Qiu, Qi Zhang, and Xuanjing Huang. Learning topical translation model for microblog hashtag suggestion. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2078–2084, 2013.
- [DWP⁺15] Emily Denton, Jason Weston, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. User conditional hashtag prediction for images. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1731–1740, 2015.
- [ESKV13] Manuel Egele, Gianluca Stringhini, Christopher Krügel, and Giovanni Vigna. COMPA: detecting compromised accounts on social networks. In *NDSS*, 2013.
- [FBGV15] Carlos Freitas, Fabricio Benevenuto, Saptarshi Ghosh, and Adriano Veloso. Reverse engineering socialbot infiltration strategies in twitter. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 25–32, 2015.

- [FPR06] Giorgio Fumera, Ignazio Pillai, and Fabio Roli. Spam filtering based on the analysis of text information embedded into images. *The Journal of Machine Learning Research*, 7:2699–2720, 2006.
- [FVD⁺14] Emilio Ferrara, Onur Varol, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *CoRR*, abs/1407.5225, 2014.
- [GGMP04] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large*, pages 576–587, 2004.
- [GHW⁺10] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47, New York, NY, USA, 2010.
- [GSDN⁺13] Frederic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 593–596, 2013.
- [GTPZ10] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: The underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pages 27–37, 2010.
- [GVK⁺12] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web*, pages 61–70, 2012.

- [GWO14] Ning Gao, William Webber, and Douglas W. Oard. Reducing reliance on relevance judgments for system comparison by using expectation-maximization. In *Proceedings of the 36th European Conference on IR Research on Advances in Information Retrieval*, pages 1–12, 2014.
- [GYK15] Toni Gruetze, Gary Yao, and Ralf Krestel. Learning temporal tagging behaviour. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1333–1338, 2015.
- [GZ16] Yuyun Gong and Qi Zhang. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.
- [HKG07] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.
- [HLW⁺12] Mengdie Hu, Shixia Liu, Furu Wei, Yingcai Wu, John Stasko, and Kwan-Liu Ma. Breaking news on twitter. In *CHI Conference on Human Factors in Computing Systems*, pages 2751–2754, 2012.
- [HP09] Amanda Lee Hughes and Leysia Palen. Twitter adoption and use in crisis twitter adoption and use in mass convergence and emergency events. In *Proceedings of the 6th International ISCRAM Conference*, 2009.
- [HTGL14] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Social spammer detection with sentiment information. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, pages 180–189, 2014.
- [HTL14a] Xia Hu, Jiliang Tang, and Huan Liu. Leveraging knowledge across media for spammer detection in microblogging. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 547–556, 2014.

- [HTL14b] Xia Hu, Jiliang Tang, and Huan Liu. Online social spammer detection. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 59–65, 2014.
- [HTZL13] Xia Hu, Jiliang Tang, Yanchao Zhang, and Huan Liu. Social spammer detection in microblogging. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 2013.
- [HW11] Jonathan Hurlock and Max L. Wilson. Searching twitter: Separating the tweet from the chaff. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.
- [IYU09] Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. Modeling social annotation data with content relevance using a topic model. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, pages 835–843, 2009.
- [JHTS11] Timothy Jones, David Hawking, Paul Thomas, and Ramesh Sankaranarayanan. Relative effect of spam and irrelevant documents on user interaction with search engines. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2113–2116, 2011.
- [JKA⁺09] Attenberg Joshua, Weinberger Kilian, Dasgupta Anirban, Smola Alex, and Zinkevich Martin. Collaborative email-spam filtering with the hashing-trick. In *Proceedings of the 6th International Conference on Email and Anti-Spam*, 2009.
- [JL07] Nitin Jindal and Bing Liu. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*, pages 1189–1190, 2007.

- [JL08] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 219–230, 2008.
- [JSFT07] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, New York, NY, USA, 2007.
- [KHLZ12] Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. On recommending hashtags in twitter networks. In *Proceedings of the 4th international conference on Social Informatics*, pages 337–350, 2012.
- [KLPM10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600, 2010.
- [KWM11] Efthymios Kouloumpis, Theresa Wilson, and Johanna D. Moore. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.
- [LCL13] Jiwei Li, Claire Cardie, and Sujian Li. Topicspam: a topic-model based approach for spam detection. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 217–221, 2013.
- [LCS11] Zhiyuan Liu, Xinxiong Chen, and Maosong Sun. A simple word trigger method for social tag suggestion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1577–1588, 2011.
- [LCW10] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: Social honeypots + machine learning. In *Proceedings of the 33rd*

- international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442, 2010.
- [LDL⁺03] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining*, pages 179–186, 2003.
- [LEC11] Kyumin Lee, Brian David Eoff, and James Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.
- [LFK12] Cristian Lumezanu, Nick Feamster, and Hans Klein. #bias: Measuring the tweeting behavior of propagandists. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*, 2012.
- [LHYZ11] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. Learning to identify review spam. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, pages 2488–2493, 2011.
- [LNJ⁺10] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 939–948, 2010.
- [LT04] Chih-Chin Lai and Ming-Chi Tsai. An empirical performance comparison of machine learning methods for spam e-mail categorization. In *Hybrid Intelligent Systems, 2004. HIS '04. Fourth International Conference on*, pages 44–48, 2004.

- [LTC13] Kyumin Lee, Prithivi Tamilarasan, and James Caverlee. Crowdturfers, campaigns, and social media: Tracking and revealing crowdsourced manipulation of social media. In *Proceedings of the Seventh International Conference on Weblogs and Social Media*, 2013.
- [LWG14] Kyumin Lee, Steve Webb, and Hancheng Ge. The dark side of micro-task marketplaces: Characterizing fiverr and automatically detecting crowdturfing. In *Proceedings of the Eighth International Conference on Weblogs and Social Media*, 2014.
- [MCL05] Gilad Mishne, David Carmel, and Ronny Lempel. Blocking blog spam with language model disagreement. In *AIRWeb 2005, First International Workshop on Adversarial Information Retrieval on the Web, co-located with the WWW conference, Chiba, Japan, May 2005*, pages 1–6, 2005.
- [MDD⁺14] Zachary Miller, Brian Dickinson, William Deitrick, Wei Hu, and Alex Hai Wang. Twitter spammer detection using data stream clustering. *Information Science*, 260:64–73, March 2014.
- [MLW⁺11] Arjun Mukherjee, Bing Liu, Junhui Wang, Natalie Glance, and Nitin Jindal. Detecting group review spam. In *Proceedings of the 20th international conference companion on World wide web*, pages 93–94, 2011.
- [MRA13] Juan Martinez-Romo and Lourdes Araujo. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, 40(8):2992–3000, 2013.
- [MRS08] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008.

- [MSOB13] Johnnatan Messias, Lucas Schmidt, Ricardo Oliveira, and Fabrício Benvenuto. You followed my bot! transforming robots into influential users in twitter. *First Monday*, 18, 2013.
- [MSYC14] Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 999–1008, 2014.
- [NGKA11] Nasir Naveed, Thomas Gottron, Jérôme Kunegis, and Arifah Che Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proceedings of the 3rd International Web Science Conference*, pages 8:1–8:7, 2011.
- [NMTM00] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, pages 103–134, 2000.
- [NNMF06] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, pages 83–92, 2006.
- [NNY08] Oded Nov, Mor Naaman, and Chen Ye. What drives content tagging: The case of photos on flickr. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1097–1100, 2008.
- [NY10] Oded Nov and Chen Ye. Why do people tag?: Motivations for photo tagging. *Communications ACM*, 53(7):128–131, July 2010.
- [OCCH11] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational*

- Linguistics: Human Language Technologies - Volume 1*, pages 309–319, 2011.
- [PYFD04] Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, and Pinar Duygulu. GCap: Graph-based automatic image captioning. In *CVPR Workshop*, pages 146–154, 2004.
- [PZP⁺11] Huan-Kai Peng, Jiang Zhu, Dongzhen Piao, Rong Yan, and Ying Zhang. Retweet modeling using conditional random fields. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 336–343, 2011.
- [QRRM11] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599, 2011.
- [RCM⁺11] Jacob Ratkiewicz, Michael Conover, Mark R. Meiss, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Detecting and tracking political abuse in social media. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.
- [RMK11] Daniel M. Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide*, pages 695–704, 2011.
- [SC14] Jieying She and Lei Chen. Tomoha: Topic model-based hashtag recommendation on twitter. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 371–372, 2014.

- [SH12] Nikita Spirin and Jiawei Han. Survey on web spam detection: Principles and algorithms. *SIGKDD Explorations Newsletter*, 13(2):50–64, May 2012.
- [SIH16] Bichen Shi, Georgiana Ifrim, and Neil Hurley. Learning-to-rank for real-time high-precision hashtag recommendation for streaming news. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1191–1202, 2016.
- [SKV10] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9, 2010.
- [SL14] Anshumali Shrivastava and Ping Li. In defense of minhash over simhash. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 886–894, 2014.
- [SLK11] Jonghyuk Song, Sangho Lee, and Jong Kim. Spam filtering in twitter using sender-receiver relationship. In *Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection*, pages 301–317, 2011.
- [SLK15] Jonghyuk Song, Sangho Lee, and Jong Kim. Crowdtarget: Target-based detection of crowdturfing in online social networks. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, pages 793–804, 2015.
- [SML⁺13] Igor Santos, Igor Miñambres-Marcos, Carlos Laorden, Patxi Galán-García, Aitor Santamaría-Ibirika, and Pablo Garcia Bringas. Twitter content-based spam filtering. In *International Joint Conference SOCO’13-CISIS’13-ICEUTE’13 - Salamanca, Spain, September 11th-13th, 2013 Proceedings*, pages 449–458, 2013.

- [SNB11] Igor Santos, Javier Nieves, and Pablo G. Bringas. *Semi-supervised Learning for Unknown Malware Detection*, pages 415–422. Berlin, Heidelberg, 2011.
- [SOM10] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860, 2010.
- [SOM13] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):919–931, April 2013.
- [SS14] Surendra Sedhai and Aixin Sun. Hashtag recommendation for hyperlinked tweets. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 831–834, 2014.
- [SS15] Surendra Sedhai and Aixin Sun. Hspam14: A collection of 14 million tweets for hashtag-oriented spam research. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 223–232, 2015.
- [SvZ08] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336, 2008.
- [SWE⁺13] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y. Zhao. Follow the green: Growth and dynamics in twitter follower markets. In *Proceedings of the 2013 Internet Measurement Conference*, pages 163–176, 2013.

- [SYW⁺14] Guowei Shen, Wu Yang, Wei Wang, Miao Yu, and Guozhong Dong. *Detecting Anomalies in Microblogging via Nonnegative Matrix Tri-Factorization*, pages 55–66. Berlin, Heidelberg, 2014.
- [TAH⁺13] Ke Tao, Fabian Abel, Claudia Hauff, Geert-Jan Houben, and Ujwal Gadiraju. Groundhog day: Near-duplicate detection on twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1273–1284, 2013.
- [TGC⁺13] Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. UNIK: Unsupervised social network spam detection. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 479–488, 2013.
- [TGM⁺11] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time url spam filtering service. In *2011 IEEE Symposium on Security and Privacy*, pages 447–462, 2011.
- [TGP12] Kurt Thomas, Chris Grier, and Vern Paxson. Adapting social spam infrastructure for political censorship. In *Presented as part of the 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2012.
- [TGSP11] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 243–258, 2011.
- [TGV⁺14] A. Tomar, F. Godin, B. Vandersmissen, W. D. Neve, and R. V. d. Walle. Towards twitter hashtag recommendation using distributed word representations and a deep feed forward neural network. In *Advances in Computing, Communications and Informatics (ICACCI), 2014 International Conference on*, pages 362–368, 2014.

- [TLGP14] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. Consequences of connectivity: Characterizing account hijacking on twitter. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 489–500, 2014.
- [TMG⁺13] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *Proceedings of the 22Nd USENIX Conference on Security*, pages 195–210, 2013.
- [TN10] K. Thomas and D. M. Nicol. The koobface botnet and the rise of social malware. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*, pages 63–70, 2010.
- [TR12] Oren Tsur and Ari Rappoport. 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*, pages 643–652, 2012.
- [TRM11] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. #twittersearch: A comparison of microblog search and web search. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 35–44, 2011.
- [TSD08] Marta Tatu, Munirathnam Srikanth, and Thomas D’Silva. RSDC’08: Tag Recommendations using Bookmark Content. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 96–107, 2008.
- [TSSW10] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G. Sandner, and Isabell M. Welp. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM*, 2010.

- [VG13] John-Paul Verkamp and Minaxi Gupta. Five incidents, one theme: Twitter spam as a weapon to drown voices of protest. In *3rd USENIX Workshop on Free and Open Communications on the Internet*, 2013.
- [Wan10] A. H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10, July 2010.
- [WBSW09] Y. S. Wu, S. Bagchi, N. Singh, and R. Wita. Spam detection in voice-over-ip calls through semi-supervised clustering. In *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, pages 307–316, 2009.
- [WHMW11] Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 705–714, 2011.
- [WJL⁺07] Zhe Wang, William K. Josephson, Qin Lv, Moses Charikar, and Kai Li. Filtering image spam with near-duplicate detection. In *CEAS 2007 - The Fourth Conference on Email and Anti-Spam*, 2007.
- [WLJH10] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twiterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270, 2010.
- [WNL⁺13] De Wang, Shamkant B Navathe, Ling Liu, Danesh Irani, Acar Tamersoy, and Calton Pu. Click traffic analysis of short url spam on twitter. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, pages 250–259. IEEE, 2013.

- [WWY14] Jason Weston, Ron J. Weiss, and Hector Yee. Affinity weighted embedding. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1215–1223, 2014.
- [WWZ⁺12] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and turf: Crowdturfing for fun and profit. In *Proceedings of the 21st international conference on World Wide Web*, pages 679–688, 2012.
- [WWZZ14] Gang Wang, Tianyi Wang, Haitao Zhang, and Ben Y. Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, pages 239–254, 2014.
- [WYYH09] Lei Wu, Linjun Yang, Nenghai Yu, and Xian-Sheng Hua. Learning to tag. In *Proceedings of the 18th international conference on World wide web*, pages 361–370, 2009.
- [WZLP15] Bo Wang, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Making the most of tweet-inherent features for social spam detection on twitter. *CoRR*, abs/1503.07405, 2015.
- [XYW06] Mengjun Xie, Heng Yin, and Haining Wang. An effective defense against email spam laundering. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 179–190, 2006.
- [XZ15a] Chang Xu and Jie Zhang. Combating product review spam campaigns via multiple heterogeneous pairwise features. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 172–180, 2015.
- [XZ15b] Chang Xu and Jie Zhang. Towards collusive fraud detection in online reviews. In *2015 IEEE International Conference on Data Mining*, pages 1051–1056, 2015.

- [XZCL13] Chang Xu, Jie Zhang, Kuiyu Chang, and Chong Long. Uncovering collusive spammers in chinese review websites. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, 2013.
- [YHG11] Chao Yang, RobertChandler Harkreader, and Guofei Gu. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In *Recent Advances in Intrusion Detection*, volume 6961, pages 318–337. 2011.
- [YHG13] C. Yang, R. Harkreader, and G. Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8:1280–1293, 2013.
- [YRSB10] Sarita Yardi, Daniel M. Romero, Grant Schoenebeck, and Danah Boyd. Detecting spam in a twitter network. *First Monday*, 15(1), 2010.
- [ZGS13] Eva Zangerle, Wolfgang Gassler, and Gnther Specht. On the impact of text similarity functions on hashtag recommendations in microblogging environments. *Social Network Analysis and Mining*, pages 1–10, 2013.
- [ZGSH14] Qi Zhang, Yeyun Gong, Xuyang Sun, and Xuanjing Huang. Time-aware personalized hashtag recommendation on social media. In *COLING 2014, 25th International Conference on Computational Linguistics*, pages 203–212, 2014.
- [ZSC11] Linhong Zhu, Aixin Sun, and Byron Choi. Detecting spam blogs from blog search results. *Information Processing and Management*, 47(2):246–262, Mar 2011.
- [ZZL12] Xianchao Zhang, Shaoping Zhu, and Wenxin Liang. Detecting spam and promoting campaigns in the twitter social network. In *2012 IEEE 12th International Conference on Data Mining*, pages 1194–1199, 2012.

- [ZZZY16] Jinxue Zhang, Rui Zhang, Yanchao Zhang, and Guanhua Yan. The rise of social botnets: Attacks and countermeasures. *CoRR*, abs/1603.02714, 2016.