國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

垃圾評論的分析與偵測 – 用流出資訊作為標準答案

Opinion Spam Analysis and Detection Leaked Confidential Information as Ground Truth

陳譽仁

Yu-Ren Chen

指導教授：陳信希 博士

Advisor: Hsin-Hsi Chen, Ph.D.

中華民國 103 年 7 月

July 2014

Opinion Spam Analysis and Detection

Leaked Confidential Information as Ground Truth

Yu-Ren Chen, Hsin-Hsi Chen

July 2014

Contents

Abstract

'Opinion spamming' usually refers to the illegal marketing practice which involves delivering commercially advantageous opinions as regular users on review websites. In this research, based on a set of internal records of opinion spams leaked from a shady marketing campaign, we are able to explore the characteristics of opinion spams and spammers to obtain some insights, and then make an attempt to devise features that could be potentially helpful in automatic detection. In the final experiments, we find that our detection model can achieve a decent performance with a set of rather basic features.

1 Introduction

In April 2013, on the Taiwan-based web forum *Mobile01*, a poster submitted a thread[1] in which several confidential documents of a **covert marketing**[2] campaign that had been conducted under the table were disclosed. The campaign instructed hired writers and designated employees to post disingenuous comments on some web forums including *Mobile01*. This revelation created a big stir at that time, since it was the first time such strong evidence supporting what most folks had considered as a 'conspiracy theory' came to light.

Mobile01, also known as *01* in Taiwan, is a web forum which mainly features discussion about mobile phones, hand-held devices, and other consumer electronics. The vast majority of the users on the site are originated in Taiwan, so the posts are mostly written in Traditional Chinese, which is the official script of Taiwan. The site was founded in year 2000 and has become one of the most well known Taiwanese local websites. As reported by *Alexa*, *Mobile01* ranked #10 in terms of website traffic in Taiwan, as of this writing.

The confidential documents, along with relevant articles describing the campaign, are on *Taiwansamsungleaks*[3], a website made by the hacker '0xb'. According to the site, the covert marketing campaign was carried out by a consulting firm that was a subsidiary company of one of the biggest IT companies in the world. In this campaign, hired posters were asked to promote a certain brand and denounce its rivals on web forums such as *Mobile01*, while

[1] `http://www.mobile01.com/topicdetail.php?f=568&t=3284729`

[2] **Covert marketing** is defined as a firm's marketing actions whereby consumers believe that the activities are not those of the firm. (Kaikati and Kaikati, 2004)

[3] `http://taiwansamsungleaks.org`

disguised as normal consumers. Among the disclosed documents, there are two spreadsheets[4] that appear to be internally-kept records of the spam posts incurred by the campaign from 2011 to 2012. Each row in these spreadsheets is a record of an incentivized forum post and consists of the poster's username, the time of posting, the url to the post, the product that was discussed in the post, and some other details.

Generally speaking, web forums provide platforms for people with similar interests to interact and share experiences with each other. Since people normally believe posts on legit forums to be based on genuine personal opinion and experience, it's considered unethical to use them to promote things for personal gain without disclaimer, and take advantage of the inherent mutual trust between forum users. As a matter of fact, such marketing malpractice violated the fair trade law, and the company in charge of the campaign was fined by the *Fair Trade Commision (FTC)* in Taiwan, after the investigation was completed.

In this research, we make an attempt to leverage these spreadsheets to generate ground truth of deceptive forum spams. After some exploration into the data, we try to come up with automatic methods for spam detection and spammer detection, and then conduct experiments to see its performance under various conditions.

2 Related Work

We orgainze the related work section into subsections focusing on different aspects, which include the studies of spam in general (section 2.1), what type of target to detect (opinion spam or opinion spammer) (section 2.2), features proposed in the past (section 2.3), and the difficulty in acquiring the ground truth data of opinion spam (section 2.4).

2.1 'Spam' in General

Spam, whose various definitions usually center around the concept of **unsolicited message** (Hayati et al., 2010), has been bothering Internet users since the rise of the Internet. Because the amount of spam is usually quite formidable, it would be too laborious to identify and remove them one-by-one manually. Therefore, finding an automatic spam detection method has

[4] with file name extension .xlsx

long been a popular research topic due to the strong demand, in addition to the fact that it's an intriguing topic in itself.

2.1.1 Email Spam

Email spam is one of the most prevalent types of spam that could be dated back to long ago. In our experience, when people mention 'spam' without giving a more specific context, it can be assumed that they're referring to email spam on most occasions. The topic of email spam detection has been extensively studied, and there is rich literature that covers in-depth exploration of this topic available. For an thorough overview, Blanzieri and Bryl (2008) surveyed the state-of-the-art-at-the-time machine learning applications for email spam filtering.

2.1.2 Web Spam

Another form of spam is web spam, whose objective is to game the ranking algorithm of the search engine in order to get an undeserved high ranking. It is usually applied as a *Black Hat SEO*[5] technique to pursue the lucrative profit that could be brought by the search engine traffic. However, as the major search engine *Google* keeps refining the ranking algorithm, nowadays simple link spams are no longer able to cheat the search engine, and could instead incur some penalty in ranking. Gyongyi and Garcia-Molina (2005) presented a comprehensive taxonomy of the web spamming techniques. Gyöngyi et al. (2004) proposed a semi-automatic method to separate good and reputable pages from web spams.

2.1.3 Social Network Spam

Just as the rise of search engine leads to web spam, as social media gained its popularity in recent years, social network spam came along. Sometimes concisely called as 'social spam', social network spam has many variants. One of the variants involves throwaway accounts created in batch to somehow bait regular users to clicks certain link for personal gain. McCord and Chuah (2011) and Benevenuto et al. (2010) both discussed the techniques of spammer detection on Twitter.

[5] use of aggressive SEO tactics without following the terms of service of search engines

2.1.4 Opinion Spam

The kind of spam we want to detect in this research is usually referred to as **opinion spam** or **review spam**. Opinion spam is related but still different from other kinds of spam from various perspectives. One of the most prominent differences is that opinion spam is arguably the most 'subtle' kind of spam, since it is not only completely ineffective, but also very harmful to the reputation of a brand (or a store, a restaurant, etc.) when got caught. Therefore, opinion spammers would generally try their best to disguise their opinion spam as genuine opinion. Carefully-written opinion spams have caused great challenges in manually identifying the spams and annotating the ground truth, which is in concert with the finding that human are poor judge of deception (Vrij et al., 2008). Ott et al. (2011) reported a very low annotator agreement score when annotating the opinion spams from a review corpus. In contrast, most of the email spams, web spams, or social network spams are fairly easy to spot by an experienced user of the respective platform.

One of the earliest researches on opinion spam is Jindal and Liu (2008), in which they attempt to detect fake product reviews on *Amazon*. Since then, this topic has been drawing increasing attention. Mukherjee et al. (2011), Lim et al. (2010), Jindal et al. (2010), Xie et al. (2012), Wang et al. (2011) are some of the following researches.

In the later parts of this paper, when we mention 'spam' or 'spammer' without specifying its type, we're referring to 'opinion spam' or 'opinion spammer', respectively.

2.2 Target of Detection

The task of opinion spam detection can be seen as a binary classification problem where we want the detection model to detect whether a given instance is a spammy or not. Naturally, each instance would be a post, and spam (post) and non-spam would be the two classes. Alternatively, each instance could be an user account, where spammer and non-spammers would be the classes, when we only care about which of the users are the black sheep. In our research, we attempt to construct the models and conduct some experiments for both types of targets.

2.2.1 Spam Post Detection

In spam (post) detection, the detection model's job is to identify whether a forum post (or a product review, a store review, etc.) is a spam post. Many of the previous researches on opinion spam aimed at detecting spam reviews, which can be seen as a type of post (Jindal and Liu, 2008; Harris, 2012; Jindal et al., 2010; Ott et al., 2011). Nonetheless, even if the target of detection is spam, we can still utilize features derived from information about the corresponding spammers, and vice versa.

2.2.2 Spammer Detection

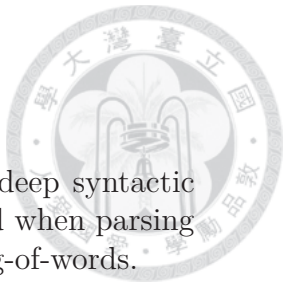
Lim et al. (2010) and Wang et al. (2011) are two of the previous researches that focused on identifying spammers, while Mukherjee et al. (2011) created a variation by making groups of spammer who worked together to write fake reviews as their detection target. In our research, we define 'spammers' to be users who have ever submitted a spam post. Under this definition, in some sense, spammer detection is not harder than spam detection, as a spammer will be identified if any of his/her spam posts is identified.

2.3 Proposed Features

A good amount of features has been proposed for the use with commonly-applied supervised learning models such as *SVM (Support Vector Machine)* (Cortes and Vapnik, 1995), or alternatively, in some ad-hoc models designed for the specific purpose. Most of these features fall into the two categories: the ones derived from the **textual contents** of opinion spams, and the ones not directly related to them.

2.3.1 Content-centric Features

In terms of features derived from contents, Jindal and Liu (2008) counted the percentage of opinion-bearing words, brand name mentions, numerals, capitals etc. Mukherjee et al. (2011) computed content similarity between reviews to examine if there are duplicate or near-duplicate reviews, which are suspicious of being spam reviews. Ott et al. (2011) used bag-of-n-grams and slightly improved the performance with psychologically meaningful dimensions in *LIWC* (Pennebaker et al., 2007). Harris (2012) took cues such as word diversity, proportion of first person pronouns and mention of brand



names. Feng et al. (2012) went a step further by adopting deep syntactic features, which are derived from the production rules involved when parsing the contents based on the PCFG, in addition to the basic bag-of-words.

2.3.2 Non-content-centric Features

Speaking of features not directly related to contents, Lim et al. (2010) and Feng et al. (2012) both made extensive use of various characteristics of user rating patterns on *Amazon*. Mukherjee et al. (2011) derived features from bursts in the amount of reviews, how early the reviews was post, and rating deviation, with respect to either groups or individuals. Wang et al. (2011) iteratively computed the trustiness of reviewers, honesty of reviews and reliability of stores based on a graph model which utilized non-content-centric features such as average rating and number of reviews. Since our dataset is obtained from a web forum, some information from product review site, such as user ratings, is unavailable, which might limit some possibilities here.

2.4 Ground Truth Acquisition

One of the major obstacles in studies of opinion spam is the difficulty in acquiring ground truth, since it's in spammer's best interest to keep it secret, and manual annotation is ineffective because of the subtlety nature of opinion spam mentioned in section 2.1.4.

A lot of effort had been put into obtaining ground truth in studies of opinion spam. Jindal and Liu (2008) assumed near-duplicate reviews were likely to be spam and followed this heuristic to build an annotated dataset. More recently, collecting annotations using crowdsourcing platform like *Amazon Mechanical Turk* had become a popular approach. Gokhman et al. (2012) discussed various techniques of obtaining ground truth in studies of deception, and argued that realistic deceptive contents could be generated from crowdsourcing, if the context of deception in practice is replicated on the crowdsourcing platform. On the other hand, it's ineffective to annotate existing deceptive contents. In fact, one of the quality indicators of fabricated opinion spams is that they shouldn't be recognizable by crowdsourced annotators. Ott et al. (2011) scraped truthful opinions from *TripAdvisor* and synthesized deceptive opinion with the help of *Amazon Mechanical Turk*.

Thus far, most of the previous researches on opinion spams appeared to adopt some sort of approximations of the actual ground truth, due to the difficulties stated in this section. On the contrary, in our research, we extract ground

truth from the confidential records leaked directly from a covert advertising campaign, which assures its 'trueness'.

3 Dataset

There are three major sources of our dataset:

1. The leaked spreadsheets disclosed by the anonymous hacker '0xb' provide ground truth of which posts are spam. (section 3.1)
2. The actual contents and various meta information on *Mobile01* compose the 'body' of our corpus. (section 3.2)
3. Product information is scraped from a phone review website name *SOGI*[6] to aid analysis requiring knowledge about the products. (section 3.3)

3.1 Leaked Spreadsheets

The leaked spreadsheets *HHP-2011.xlsx* and *HHP-2012.xlsx* keep the histories of the opinion spam posts made in 2011 and 2012, respectively. Several discussion platforms were spammed, but for simplicity, we consider only the opinion spams and the corresponding spammers on *Mobile01*, which make up the majority of the records contained in the spreadsheets.

Among the columns in the spreadsheets, **urls to the spam posts** and **usernames of the spammers**[7] are extracted. Some typos and inconsistent ways of presenting the usernames (e.g., lowercase vs uppercase, confusion between similar looking Unicode characters) are manually checked and fixed. Furthermore, recorded urls linked to pages on *Mobile01* might appear in different forms. To be able to reliably match the posts we scraped later, a 3-tuple $(fid, thid, pnum)$ are extracted from each of the urls, where $fid, thid$ and $pnum$ refers to **forum id**, **thread id** and **page number**, respectively. These 3-tuples serve as unique identifiers of a page in a thread on *Mobile01*. For the example *Mobile01* page url below, the extracted 3-tuple identifier would be $(566, 4009283, 2)$.

`http://www.mobile01.com/topicdetail.php?f=566&t=4009283&p=2`

[6] `http://www.sogi.com.tw`

[7] Whenever the word **spammer** or **user** is used hereafter without further details, we're talking about **spammer account** or **user account**, respectively, since no way can we find out who the actual human poster behind an user account is.

Since we regard any user who has ever post a spam post as **spammer**, any account which is contained in the spreadsheets is considered to be spammer. Thereafter, we have a set of 2-tuples that each consists of a spammer's username and a nested 3-tuples identifier leading to the page containing one or more spamming posts of the spammer. An example snippet of data extracted from leaked spreadsheets is listed in table 1. As a matter of fact, the spreadsheets didn't specify exactly which post in the page the url points to is spam, so if a linked page contains multiple posts by the poster with the recorded username, we simply consider all of them as spam posts.

username	**thread page id**		
	fid	**thid**	**pnum**
amberwangtw	568	2378318	1
nickliu623	568	2682497	1
jackR	14	1977960	1
popstyle	568	2661837	1
kk8928166	568	2636349	2
賈蘇林	568	2400890	1
CBR600RR2007	217	2399752	1
QQ_578	61	2605621	1

Table 1: data extracted from leaked spreadsheet

3.2 Mobile01 Corpus

A large portion of previous related studies used dataset scraped from product or store review websites such as *Amazon* or *TripAdvisor*, whereas our corpus is scraped from a **web forum**. Another difference is that the contents on *Mobile01* are mostly written in **Traditional Chinese**, with little bit of English phrases scattered around, rather than predominantly written in English as in previously used corpora.

Mobile01 works just like a typical web forum, such as the ones based on *phpBB* or *VBulletin*, and here we assume the readers to have a basic understanding in how web forums work.

Since more than **70%** of the recorded spams were submitted to the *Samsung (Android)* board on *Mobile01*, we decide to focus our analysis on this board. By *SSH* tunneling through *Linux* workstations maintained by the department, within a reasonably short period of time we were able to fetch all the threads along with the contained posts accessible by a regular member on the *Samsung (Android)* board on May, 2014. In addition, profiles of users

who have ever post in this board are also retrieved. To get the relevant information we need out of the retrieved web pages, we parse *HTML* with the help of *BeautifulSoup*[8]. After the laborious tasks of mangling with the raw data, the cleaned data are all stored into a *SQLite* database, for the ease of later accesses and possible modifications. For instance, each post is stored as a record in the **POSTS** table, which has a column for each attribute of a post.

Basic counts, scraped attributes, and randomly-selected snippets of the *SQLite* tables **POSTS**, **PROFILES**, and **THREADS** are shown in Tables 2 to 7.

It should be noted that the data we scraped from *Mobile01* is the 'May 2014 version', while the spam activity we want to investigate happened during 2011 and 2012. Ideally, a snapshot at the end of the 2012 may suit our need best. By the time we collected the dataset, some posts could have been edited or removed, and profiles could have evolved with time had the users stayed active. In table 7, we can see 4 out of 10 randomly picked profiles have the last login time '1399075200', which represents 5 May 2014, the date when the profile data were scraped from *Mobile01*.

table	row counts
posts	632234
threads	41759
profiles	58531

(a) Scraped data

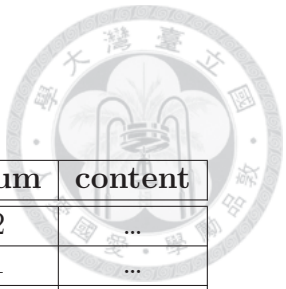
target type	row counts
spammers	300
spam posts	3116

(b) Labeled data

attribute	description
thid	id of the thread to which the post belong
time	submission time of the post
uid	id of the poster who made the post
uname	username of the poster
nfloor	position relative to other posts in the thread
pnum	page number on which the post is
content	structured content in *HTML*

Table 2: attributes of the table **POSTS**

[8] `http://www.crummy.com/software/BeautifulSoup`



thid	time	uid	uname	nfloor	pnum	content
2753035	1337723880	2135371	ZSKOR	20	2	...
2007342	1297610580	151149	湯尼小	1	1	...
3060762	1353922620	2369024	pinckstraw	12	2	...
2550662	1331902200	1817096	iamfishfis	1074	108	...
1830978	1288783020	185736	wunit	26	3	...
3841427	1396355820	2253702	bluestaral	1	1	...
2741044	1337137440	1448858	wei700818	9	1	...
2467506	1322415300	160444	nella76327	4	1	...
1899252	1291632600	1426406	cloud2211	9	1	...
3227368	1362157920	2212133	jinshun000	187	19	...

Table 3: a snippet of the table **POSTS**

attribute	description
thid	id of the thread
fid	id of the forum (board) in which the thread is
title	title of the thread
pages	number of pages in the thread
clicks	number of clicks (views) on this thread
time	submission time of the thread (=first post's)

Table 4: attributes of table **THREADS**

thid	fid	title	pages	clicks	time
3851232	568	請問沒有參加預購的人 4/11 哪裡比較能買到 s5?	1	880	1396908300
1710011	568	有住台中的神人大大能幫忙 root I9000 嘛?	5	6941	1282155000
2301390	568	S2 的 5.1 聲道怎麼比不開還不太好聽!!	1	1474	1313325000
2810711	568	S 3 嚴重收訊問題～有同樣問題的還說說吧	2	4972	1340631900
3029682	568	我的 note ii 32gb 沒貼神腦或聯強的貼紙	1	319	1352040120
3015440	568	越南 NOTE2 開箱之尋寶圖 ????	1	1518	1351294380
2582007	568	Samsung Galaxy mini s5570 使用 Kies 程式的問題	1	154	1328747100
2848181	568	你們的 note 會這樣嗎？	1	2288	1342674000
2258807	568	再跟新 9100 的新版本的時候出現不可預期的錯	1	231	1310961000
3287201	568	遊戲的背景音樂破破的	1	143	1364908860

Table 5: a snippet of the table **THREADS**

attribute	description
uid	id of the user
reg_time	time of registration on the site
login_time	last time the user logged in
n_threads	number of threads initialized by the user
n_eff_posts	number of effective posts
n_posts	number of all posts
n_replies	number of replies, which is equal to (n_posts - n_threads)
score	'karma' given by other users to the threads the user make
p_phone	%proportion of posts made on the smart phone section

Table 6: attributes of table **PROFILES**

uid	reg_time	login_time	n_posts	n_eff_posts	n_threads	n_replies	score	p_phone
1873586	1295136000	1395878400	18	18	0	18	0	88
820575	1192406400	1292976000	5	5	0	5	0	60
2495668	1367107200	1387756800	16	16	0	16	0	100
2500546	1367798400	1397692800	15	15	2	13	0	6
941418	1204934400	1398816000	9	9	1	8	0	22
1850046	1292457600	1398643200	50	44	3	47	5	42
2678858	1397174400	1399075200	4	4	1	3	0	50
636450	1172620800	1399075200	71	66	1	70	0	18
814919	1191801600	1399075200	328	176	14	314	0	0
165425	1125964800	1399075200	561	540	13	548	4	74

Table 7: a snippet of the table **PROFILES**

3.3 Product Information

We scrape product information of all cell phone and tablet of the top brands listed in the front page of *Sogi*. The scraped attributes of each product are shown in the table below.

attribute	description
brand	brand name of the product
specs	product specifications
price	estimated price
release	release date of the product
description	product description

Table 8: attributes of products

People use a wide variety of aliases to refer to cell phone or tablets products on *Mobile01*, and very rarely would call the products by their full exact names. To be able to match with as many product mentions as possible, we take every **1-word or 2-word fragments** from the full name as the aliases. And if the name contain Roman numerals, we convert them to the respective Arabic numbers to create more aliases. There ought to be a lot of false aliases when they're constructed in such a loose manner, such as general terms like '3D', 'pro', so we just remove these by sifting through the matching results, as the false aliases without any match most likely won't do any harm.

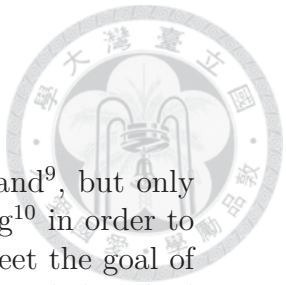
Many of the products of the same brand have some of the aliases shared. For example, people often use *Note* to refer to a *Samsung* product, but there's a dozen of the *Samsung* products having such alias. Since what we ultimately care about is the **brand** the product belonged to, we just deem it as a mention of an arbitrary *Samsung* product of the *Note* series.

4 Data Exploration

In this section, we will inspect the dataset to get a grasp of what is going on in this organized campaign of covert marketing. It should be noted that some of characteristics might **not** be manifested by other similar marketing campaigns, since each of them might have its own 'game plan' that involves a different objective in a different context.

4.1 Subtlety

One of the basic properties we observed is that most of the spam posts don't really look suspicious, which echoes the discussion in section 2.1.4. Spammers usually deliver their opinion about brands in a subtle way which blends them into the discussion, not to mention that a portion of the spam



posts (mostly replies) don't even carry opinion about any brand[9], but only serve the purpose of keeping the discussion alive and bumping[10] in order to attract more attention to specific topics of the thread that meet the goal of the campaign. Moreover, even before the whole story was revealed, it had been rumored for years that some of the posters on *Mobile01* are part-time paid writers, which may cause the spammers to be extra careful to avoid backfiring from the community.

The following is some examples of the 'subtle' spam posts selected from the dataset. Without the ground truth available, we may not be able to identify these posts as spams beforehand. The dark gray parts contain the titles of the thread the posts are within, and the *nfloor* of the posts (position of the posts in the threads); the light gray parts are the contents of the posts.

Xperia mini pro 值得入手嗎 #45

17000 你可以買更好的阿例如再加個 1000 去買 HTC Sensation
MINI 我記得訂價很便宜
搞不好你還有機會買兩隻咧

This post subtly mocked the higher price of a *HTC* product via a price comparison with *MINI*, where *MINI* was not even referring to a *Samsung* product but a *Sony* one.

Galaxy note 退訂 +1 #42

印象中，之前三星的產品預購好像都沒有公佈售價啊？
其實一個願打一個願挨啦！眞的沒辦法接受的話
就等上市再買囉

< 刪文 > #8

明明可以刁你卻無條件讓你換機…
這樣的服務還會讓你特地在這裡發文表示心痛，

[9]In this paper, we still deem these spam posts as 'opinion spam', but probably should have come up with a more appropriate name for this specific type of spam posts.

[10]Replying to a thread would 'bump' it to the top place of the board, since threads are ordered by the time of the last reply on most web forums including *Mobile01*.

那也難怪會有人想把自己送修的事情鬧上新聞了……

S2 訊號旁的 3G 符號不見了 = = #9

打去電信業者客服問問看
會不會是你家那邊收訊有問題？

The above three spam posts defended *Samsung*'s service and products with a seemingly unbiased and rational tone. Judging solely by the content of these posts, it's by no mean easy to tell if they're spam.

有用 I9003 的大大, 分享一下使用心得吧 #2

| 你 lag 了 |i9003 我沒用過
看有沒有其他大大可以跟你分享
不然爬文找找應該也有
| 請爬文 |

GALAXY S2 進化了 #43

可能不到記者會
也不能確定台灣上市的版本呀

These posts have little actual contents, and seem to be there only to heat up the discussion about some *Samsung* products.

請問各位神人 NOTE 的白色版本預計何時會有? #10

聖誕節前…好像蠻應景的？
香港都上了，台灣應該也快了吧？

兩個禮拜本來很順怎麼遊戲應用軟體越多
開始越來越慢那個讀取符號出現越來越久 = = #9

還是常常整理程式比較好啦
用不到的程式就刪掉吧

These two posts are similar to the previous ones, but they managed to provide

somewhat helpful answers to fit into the conversation.

NDSL 的觸控筆和 GALAXY NOTE 可否共用？#1

我發現 NDSL 的觸控筆和 GALAXY NOTE 的觸控筆很像，
他們可否共用？原理也類似嗎？

This is the first post in a thread. It intended to initiate discussion about a *Samsung* product with a question about it. It doesn't seem to contain opinion about any brand, like many of the previous posts.

誰說女生就一定要裝的粉可愛！我就偏要買 GALAXY R！#5

剛看了一下
Galaxy R 好像沒有消除雜訊的這個功能選項？

幫女王慶生最好的方式, 就是送他一隻 SII 再推他下山谷! #42

你們的對話也太有趣
連爸爸都要請出來了
| 大笑 |

準備衝 Galaxy Nexus 的來簽到吧 #24

拖到月底也太久了吧
店家不清楚可能真的要打電話問一下了
我才不想白跑一趟買不到手機咧
等等就來問問

一起透過 i9100 的鏡頭，分享生活中的小確幸吧！#24

一張大溪的全景照
一位 100% 的向日葵人

白色 SII+ 粉紅色 SGP protector glass 好看嗎? #7

樓主貼的保護貼好美啊！
不過價格……好可怕啊！價位真的是有點高，可以直接買保護殼了！

熱騰騰的 Galaxy Nexus 終於到手啦 #41

newlu 大
我傾向不包膜耶
弧形螢幕另一個好處就是這個
而且包膜的話鏡面質感好像會下降
好好保護他就好拉

The above posts are all replies to threads initiated by spam posts. Again, these posts only serve the purpose of heating up the discussion or just keep it alive. Since the intended messages may already be delivered by the first post or some other replies in the thread, the spammers avoid stating any strong opinion about the brands directly to make these posts even less suspicious.

Such carefully written spam posts may make the automatic detection very challenging, because the content-centric features could be ineffective. Nevertheless, in the later experiments, we find the contents still encompass some clues that help spam detection greatly.

4.2 Low Spam Post Ratio of (Some) Spammers

Spammers are the posters who had submitted any spam post recorded in the leaked spreadsheets, as defined in section 3.1. Even though we inspect the posts from the training set, which only contains posts from the board where the 'spam density' is the highest, still, only about **33%** of the posts **by spammers** are recorded as spam. The distribution of the **#spams / #posts** ratio of each spammer in the training set is plotted in the following figure.

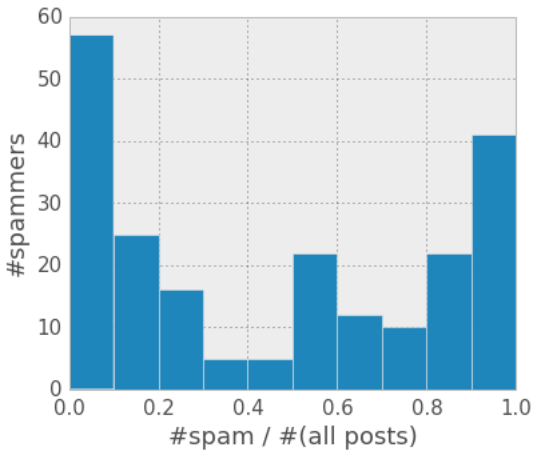


Figure 1: distribution of the spam post ratios among spammers

The figure demonstrated that a large amount of the 'spammers' actually rarely spammed. The majority non-spam posts of these spammers could neutralize the spam signal extracted from posts if we try to average them on a per-user basis.

4.3 Different Types of Spammer Accounts

In fact, there seems to be mainly two types of spammer accounts in this dataset: accounts of **reputable** posters who are paid one or few times to write quality long post to promote the brand, and **throwaway** accounts shared internally among the spammers to synthesize public opinions. The figure below is the scatter plot of the **spam post ratio** vs **total number of threads** made by the **spammers**.

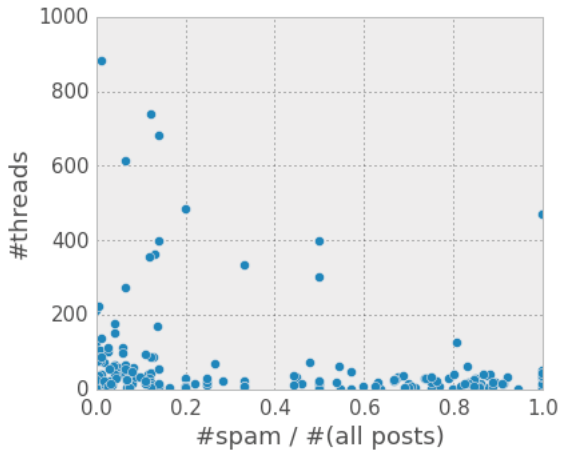


Figure 2: spam posts ratio vs number of threads

We can see that most of the spammers with a high *#spams/#posts* have initialized very few threads, which could be a clue that these are the throwaway accounts created for the sole purpose of spamming. On the other hand, accounts with a lower ratio show a bigger variance in #threads. Some of them are likely to be reputable posters, who are usually the ones that makes lots of threads.

Usually, throwaway accounts are often **created in mass** within a short period of time, as it takes much more effort and patience to spread out the daunting task of registering throwaway accounts, especially if a large amount of them are needed. To test if this applies in our dataset, we adopt a simple heuristic that categorize account initiating **less than 35 threads** as throwaway account, and reputable account otherwise. In the figure below, the number of spammer accounts registered within each two weeks after 2009 in the training set is plotted.

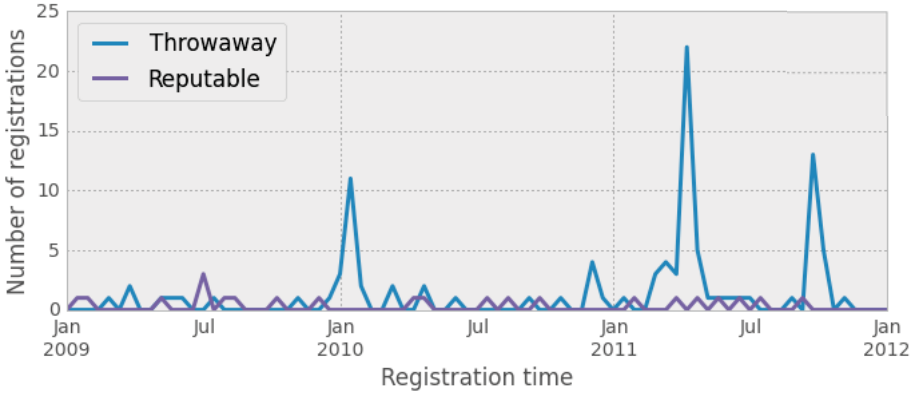


Figure 3: number of spammer accounts created within each 2 weeks

Indeed, there are three short periods when particularly high numbers of throwaway spammer accounts were registered: January 2010, April 2011 and October 2011. On the other hand, speaking of reputable accounts, the times of registration are spread quite evenly. This observation could later help us identify throwaway spammer accounts.

4.4 First Post vs Replies in Threads

As in most online web forums, the **first post**, also known as **original post** in a thread is written by the user submitting the thread. First posts are relatively **richer in content** as they serve the critical role in initializing a discussion on a specific topic. On the other hand, **replies** are often quite **concise**, and sometimes don't really carry any opinion, as manifested in some of the spam reply examples listed in section 4.1.

First posts and the **replies** in threads display different characteristics in many aspects. In the following figure, we can see that the first posts tend to contain more characters. Moreover, at least one image is embedded in **19.2%** of the first posts, but only in **4.1%** of the replies.

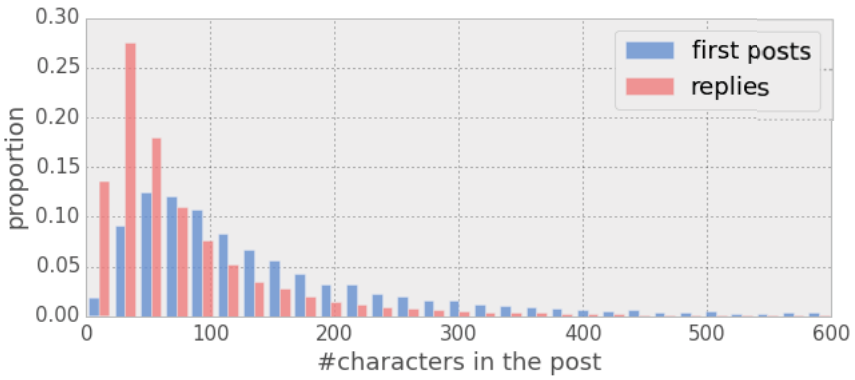


Figure 4: #characters in first posts and replies

The table below shows the spam counts and proportions for first posts and replies in training set. It's a bit surprising to see the ratio of spams in first posts is as high as **5%**. In other words, for every 20 threads in the training set, one of them is created for covert marketing! In contrast, %spam is much lower for replies.

type	#posts	#spams	%spams
first posts	10951	546	4.99%
replies	148481	1337	0.90%
all posts	159432	1883	1.18%

Table 9: #spams and %spams in first posts vs replies in training set

Considering all these differences between first posts and replies, we decide to separately train a detection model for each later. The performance is expected to look much nicer for first posts, since they carry more information (richer content) and have a significantly higher spam density, in contrast with replies. The high performance will be very helpful when we leverage prediction results on first posts to assist the spammer detection.

4.5 Pattern in Submission Time of Posts

Because making spam posts on *Mobile01* is a **job** rather than a leisure activity for the spammers, we postulate that a higher percentage of spam posts would be submitted during work time, compared to non-spam posts.

To check our postulation, we plot the distribution of submission time of spam and non-spam. In figure 5, the submission time of non-spam posts distributes

pretty evenly over each day of week, whereas the amount of spam posts drops drastically on Saturday, and has a moderate decrease on Sunday. In figure 6, we can observe that in each hour of day, more spam posts are submitted during work hour, especially between 10 a.m. and 11 a.m., while non-spam posts are more often made during the spare hours. Hence, we see there is more or less a trend that spam posts are more often made during work time than leisure time, in comparison with non-spam posts.

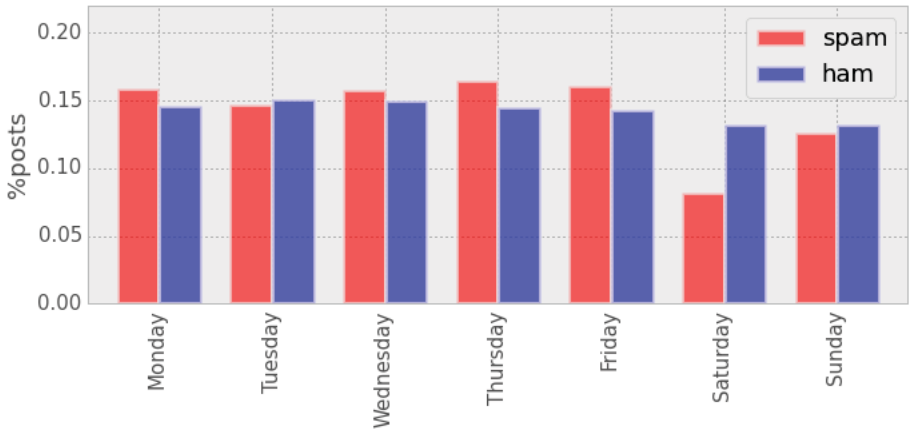


Figure 5: proportion of spam submitted throughout a week

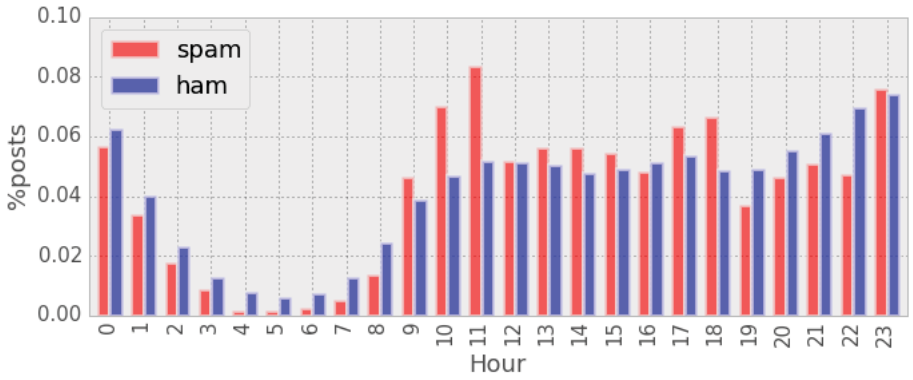


Figure 6: proportion of posts submitted throughout a day

4.6 Activeness of Threads

The threads started by spam first posts are expected to be more active, since those are written to **draw attention and exposure**, while non-spam threads may or may not be created with such intent in mind.

One intuitive way to measure the activeness of a thread would be counting the total number of posts in the thread, which is equal to *1 (first post) + #replies*. In the figure below, the numbers of posts in spam and non-spam threads are plotted. Clearly, spam threads tend to attract more replies, which could be either spam replies or non-spam replies.

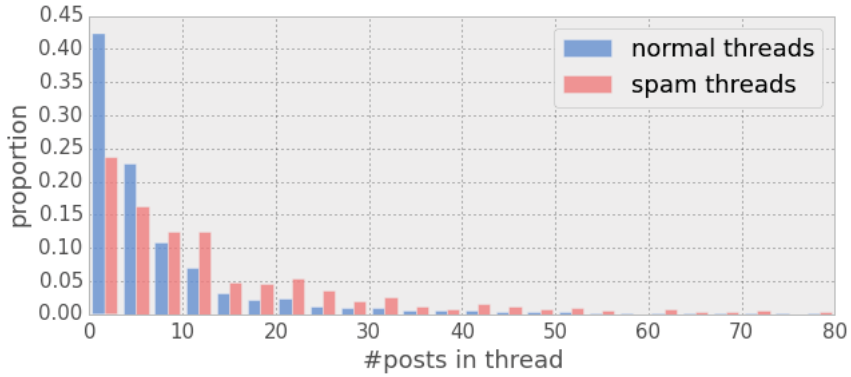


Figure 7: #posts in spam threads vs normal threads

Another way to measure the activeness of a thread is with the number of clicks. This is one of the primitive attributes of thread we scraped, as described in table 4.[11] As shown below, spam threads seem to get more clicks in comparison with normal threads.

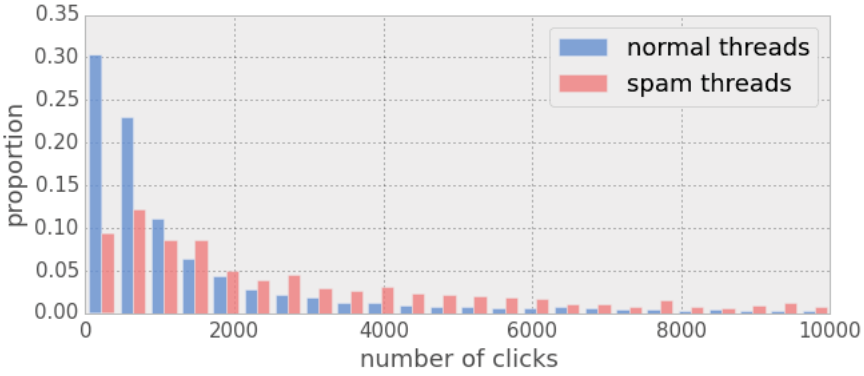


Figure 8: #clicks in spam threads vs normal threads

[11] Although not visible on the current web interface, the number of clicks is still available somewhere in the *HTML* of the threads.

4.7 Collusion between Spammers

Looking into the leaked speadsheets, we notice that a few threads contain multiple spam posts submitted by different accounts, which is an indication of **collusion** going on between **multiple spammers**. Usually, these spammer would express similar opinion in the same thread to reinforce the credibility, or it could be just a result of multiple spammers bumping the same thread[12] in an attempt to attract more attention to it.

Sometimes, it could be just the same person submitting posts with different spammer accounts in a thread, but still, it can be seemed as collusion between multiple spammer accounts on the surface.

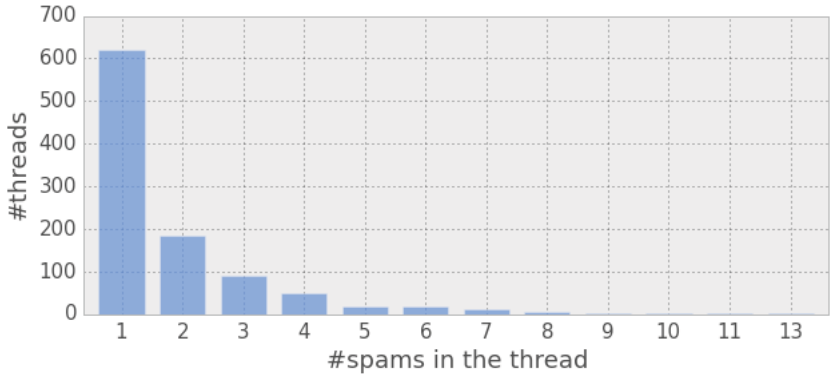


Figure 9: number of threads containing specific number of spams

In the figure above, we could observe that there are a few threads containing 2 or more spam posts. In fact, as much as **67%** of the spam posts are in a thread with at least 2 spam posts.

For a concrete example, we excerpt the first 10 posts of an example thread, in which 7 of them are actually spam posts, which were post by 5 unique spammer accounts.

準備衝 Galaxy Nexus 的來簽到吧

#1 woosawowo spam

這禮拜 Galaxy Nexus 就要上市了

[12] which is often started by a spam first post

等好久了早就準備好要趕快入手可惜他沒在資訊展開賣阿
不然應該可以有很多優惠大放送
看到好多體驗文超級生火的
有沒有已經準備好銀彈要衝 Galaxy Nexus 的一起來簽到吧
話說除了台哥大有預購之外
有沒有其他地方可以直接買空機阿
比如說三星旗艦店之類的
好像沒什麼消息

#2 甘草仔 spam

去過 Galaxy Nexus 高峰會之後，
我認眞的考慮入手 ICS，我眞的覺得好棒喔
但這隻手機好像是台哥大獨賣 哭哭

#3 square.chen ham

可以單機購買啊 在配合信用卡 12 期免利息.....

#4 小籠包 spam

甘草仔別急，我覺得你應該會贏得手機
所以就不用買啦
對了我要衝啦
趕快讓 nexus one 退休了
今天問台哥大的人
居然一問三不知
只叫我留資料等候聯絡
說她要再問問看
眞是不合格
我已經打算 NP 去台哥大了

#5 TTW2010 ham

我本來有在台哥大預購 6 號
但是打電話去問出貨時間
台哥大表示 15 出貨 16 號之後才會到

所以我就退掉了
然後跑到台哥大門市去問能不能預購
門市說不能而且不一定每個門市都會到貨

所以.....
我星期四打算先跑離家最近的台哥大
沒有就跑第二近的三星生活館
再沒有就跑去內湖家樂福的三星
應該就能買到吧
因爲老實說好像不是那麼熱門
不討厭 HTC 也買了不少 HTC, 但是對 H 粉超級厭惡,
那容不下”異”見的心態眞的很可怕

#6 woosawowo spam

也許通訊行會有貨吧我猜
不過我跟小籠包意見一樣
我覺得甘草大很有可能得獎阿
我應該也是會去三星旗艦店問問看吧
如果眞的沒有的話就去台哥大看看
希望 15 號就能到手

#7 jiantz spam

我想我還是先觀望一下
等 woo 大開箱了
15 號 Galaxy Nexus
16 號 iPhone4S
到底最後會選哪一支
現在也還是個謎阿

#8 甘草仔 spam

哈摟 小籠包 原來妳坐我旁邊壓 妳好妳好 希望承您貴言搂
WOO 大也感謝您的讚美
不用觀望了啦 兩支一起買，先買 NEXUS XD

我覺得預購已經來不及了，我想直接去沒市把玩實機，
有現貨可以考慮直接殺了 XD
我可以廣告我的文章嗎 XD

#9 danadanad **spam**

我眞的覺得這支蠻屌的耶
以前看 NEXUS S 都沒什麼感覺
但是這次 GALAXY NEXUS 可以很明顯的感受到 GOOGLE 的誠意

#10 noisycat **ham**

Galaxy Nexus 確定只有兩個地方有賣，
一個是三星旗艦店 (空機)，一個是台哥大!!

These 7 posts basically all took the same stand and more or less conveyed some positive opinion about *Samsung*. Leveraging such collusive activities between spammers improves the performance of our spammer detection model by a great margin, as demonstrated in section 5.6.5.

5 Detection

In this section, we will discuss some aspects of devising the detection models, which includes selecting a evaluation metric (section 5.1), how we split the dataset into training set and test set for the posts and the user accounts (section 5.2.1), and the machine learning procedure (section 5.3). Three detection models will be constructed:

1. Spam detection for **first posts** (section 5.4)
2. Spam detection for **replies** (section 5.5)
3. **Spammer** detection (section 5.6)

where we would add each type of features iteratively to see how the performance of the models will be progressively improved.

5.1 Evaluation Metric

In our dataset, both spam posts and spammers' ratio are quite low. Therefore, **accuracy** shouldn't be the main metric to look at since it's dominated by the majority non-spam/non-spammer class, about which we don't really care.

Let's just walk through it with spammer detection, while the same arguments can also be applied to spam detection:

High **precision** on the spammer class is desired because we don't want to falsely incriminate a innocent forum user as a spammer; high **recall** on the spammer class is also desired because we'd like to find as many opinion spammer out there as possible. Depending on the application, precision could be more important than recall, or vice versa. For instance, in an application where the detection system is used in an initial filtering stage narrowing down the set of suspicious users for a later stage of manual classification, high recall might be preferred over high precision since misclassifying a spammer as normal user completely rules out the possibility of identifying the instance right, while identifying a normal user as spammer could still be corrected in the later stage of the pipeline.

Because no particular application is aimed at, we don't have a prior preference on either precision or recall[13]. Therefore, our evaluation metric of choice would the harmonic mean of precision and recall, also known as **F-measure**, on the **spam/spammer class**.

5.2 Data Splitting

We split our data instances into the **training set** and the **test set**. Data exploration discussed in section 4 was conducted only on the training set; moreover, model selection and parameter tuning are performed based on the result of 5-fold cross validation on the training set. Avoiding ever touching the test set until the final evaluations makes the evaluation result on test set better reflects the real world expected performance.

[13]To leverage the model for first posts in spammer detection, we actually prefer it to have higher precision for 'internal use', but we choose to not take it into consideration when selecting the evaluation metric.

5.2.1 Posts (for Spam Detection)

For spam detection, each instance in our dataset[14] is a post. These posts are assigned to either training set or test set according to their **temporal orders**. Posts submitted between **Jan 2011** and **Dec 2011** are selected into training set, and the ones submitted between **Jan 2012** and **May 2012** are put into the test set. The reason we didn't utilize all the posts from 2012 is the ratio of spam posts drops drastically after May 2012 (only 30 spams in total), so we simply excluded those to keep the ratio of spam posts in the test set close to the training set.

However, there is a problem that, for many of the posts in test set, there also exists posts by the same user in training set. Under this circumstance, even if the final trained model performs well on **test set**, it doesn't necessarily imply that the model is a good opinion spam detector. The model might just capture the **writing habit** of the spammers that may or may not have intrinsic connection to the spam activity. For example, a spammer might use some words all the time in the spam posts purely out of personal preference, which would cause the model to learn to recognize such words as 'spam keywords' and thus possibly gain performance on test set, without really capturing the essence of opinion spam. Although this issue can't be completely eliminated considering that some spammer accounts are shared by the spammers[15], still we try to mitigate it by removing all the posts by user accounts who have posts included in training set from the test set, and call the resulting set '**test set***'. As a result, there won't be any posts submitted by the same user account between training set and test set*

	#spam posts	**#all posts**	**spam ratio**
training set	1883	159432	1.12%
test set	1233	92552	1.33%
test set*	414	32932	1.26%

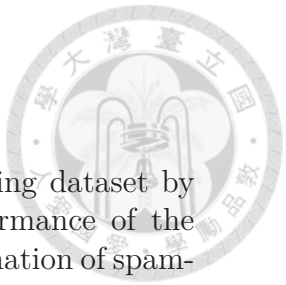
Table 10: training and test set of posts

5.2.2 Users Accounts (for Spammer Detection)

Similar to the previous section, we want to assign user accounts to the training set and the test set according to temporal order. Time of the account's

[14]Here, and in this whole detection section, 'dataset' refers to the set of instances for our particular detection task, rather than the whole dataset we collected in section 3

[15]Here, 'spammers' refer to the actual human posters, rather than the spammer accounts as in most uses of 'spammer(s)' in this paper



registration comes to mind. However, the rationale of splitting dataset by temporal order should be that we could evaluate the performance of the model detecting spammers appearing in future given the information of spammers caught in the past[16], which is a likely scenario for real world applications. Registration time does not always signify the **active periods** of the users. In this regard, **submission times of the posts by the users** is a more sensible choice.

Following the thinking in the previous section, apparently, users that have submitted a post during the first period (Jan 2011 to Dec 2011) but not the second (Jan 2012 to May 2012) should be put into the training set, and users having submitted a post in the second but not the first period should be assigned to the test set. Which set should the users who have submitted posts in **both periods** be assigned to though? Since we are going to use the detection model for first posts to assist spammer detection in section 5.6.3, this set of users shouldn't be in the test set as some of the spam first posts by these users might have been 'peeked' by the model for spam post detection. Hence, these users are assigned to the training set.

	#spammers	**#all users**	**spammer ratio**
training set	215	17216	1.25%
test set	84	8603	0.98%

Table 11: training and test set of users

5.3 Machine Learning

The machine learning procedure is conducted mainly with the help of the *Scikit-Learn* library (Pedregosa et al., 2011). We've tried various learning algorithms such as *Logistic Regression*, *SVM with linear kernel*, *SVM with RBF kernel* from *Sickit-Learn*, *SVMperf*, etc. Most of the time, *SVM with RBF kernel* seems to win out by a non-negligible margin. *SVMperf* claims to somehow **directly optimize** the F-measure (Joachims, 2005), but the result F-measure from our experiments is not better than *SVM with RBF kernel* from *Scikit-Learn*, while *SVMperf* taking a much longer time to train a model. Therefore, we decide to stick with *SVM with RBF kernel* from *Scikit-Learn*, which is actually a *Python* wrapper for the widely-used **LibSVM** (Chang and Lin, 2011), to conduct the of our experiments. As suggested in Hsu

[16]By replacing 'spammers' with 'spam posts', it becomes the rationale for splitting posts by temporal order in the section 5.2.1

et al. (2003), we scale each feature to **zero mean** and **unit variance** before feeding it to *SVM*.

There are two primary hyperparameters C and γ to be tuned in **SVM with RBF kernel**. For this purpose, whenever a model is to be learned, we first run **5-fold cross-validation** multiple times on the training set to facilitate a **grid search** on C and γ with **F-measure** as the metric to optimize. The grid to search is represented below.

$$(C, \gamma) \in \{10^x | -3 \leq x \leq 3, x \in \mathbb{Z}\} \times \{10^y | -5 \leq y \leq 2, y \in \mathbb{Z}\}$$

5.4 Spam Detection for First Posts

As discussed in section 4.4, we'd like to train a detection model specifically for first posts in thread, so we only use the first posts from the training set and test set of posts introduced in section 5.2. The counts and ratios of spam for the first posts are listed below for future reference.

	#spam posts	**#all posts**	**spam ratio**
training set	546	10951	4.99%
test set	208	5870	3.54%
test set*	70	3035	2.30%

Table 12: only considering first posts in thread

Due to the low ratio of spam posts in training set, in the following experiments on spam detection for first posts, we randomly (but deterministically) select 60% of the non-spam posts to remove from the **training set**[17] beforehand, so as to speed up the learning procedure.

5.4.1 Random Baseline

As an absolute baseline, the model predicts whether a first post is spam based on the result of flipping a fair coin. As expected, the precision is about equal to **ratio of the spams**, which is 3.54% and 2.30% on test set and test set*, respectively. The recall is around 50%, which reflects that fact that there is a half chance we correctly identify a spammer as such by flipping a fair coin.

[17]Notice we're not downsampling the non-spam instance from the test set.

	precision	recall	F-measure
test set	3.43%	49.04%	6.42%
test set*	2.52%	55.71%	4.82%

Table 13: random baseline for spam detection for first posts

5.4.2 Bag-of-words

After performing **Chinese word segmentation** on the HTML-stripped cleansed content from each post with *Jieba*[18], we count the occurrence of each word in training set, and construct a 'vocabulary' with these words. Next, rare words with less than 5 occurrences are removed from the vocabulary, since these would be the **sparse** bag-of-words features and might cause overfitting. On the other hand, words appeared in over 30% of the posts are also removed, as these are likely to be **stop words** or the like. After the vocabulary is set up, we represent each post as a vector of occurrence of each word in the vocabulary, where the occurrences are normalized by the length of the post.

In bag-of-words, each word in the vocabulary corresponds to a feature. Since the high number of features (words) could slow down the training process significantly and may cause overfitting, we apply **randomized PCA** (Halko et al., 2011) on the $\#posts \times \#words$ bag-of-words matrix to reduce the word dimension. The desired number of dimension to reduced to with PCA is tuned by looking at the average F-measure from 5-fold cross validation on the training set, as plotted in the following figure.

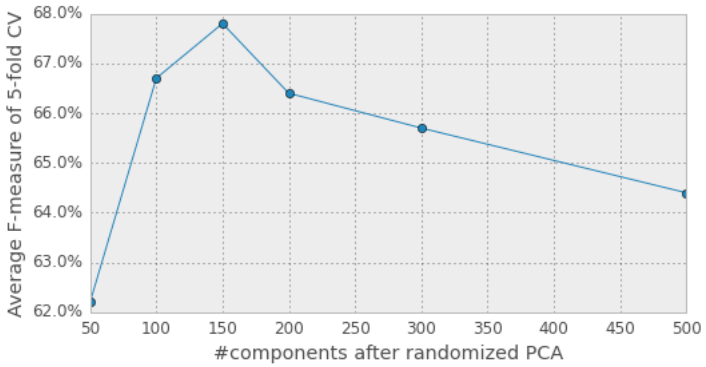


Figure 10: values of F-measure as #component in PCA changes

[18] https://github.com/fxsjy/jieba

The absolute performance shown in the plot might look unusually high. However, it's partially due to the fact that we downsampled the non-spam posts in training set, so the validation sets in 5-fold CV all have much higher ratios of spam posts than test set. What we really care about is the **relative** performance. As shown in the plot, reducing to 50 components may cause too much information loss and thus deteriorating the average F-measure. On the other hand, too many components may cause some degree of overfitting which also worsens the performance. The average F-measure is the highest when the bag-of-words is reduced to **150 components**, so we adopt it to train our model on the whole training set and see how it performs on test set.

	precision	**recall**	**F-measure**
test set	62.89%	48.08%	54.50%
test set*	50.00%	51.43%	50.70%

Table 14: content bag-of-words features only (150 components)

The performance is actually decent, whereas our observation on subtlety of the spam posts in section 4.1[19] that first post gives us a hunch that the contents of the posts might not give big clues about whether a post is spam, since the contents of spam posts are well-disguised.

Such result makes us curious about what's happening under the hood. To dive deeper into it, we'd like to get the importance of each feature in order to observe what types of words are the decisive factors in the model's predictions. However, for a **non-linear model** like *SVM with RBF kernel*, there's no simple way of computing importance of each feature. Nevertheless, by 'falling back' to linear kernel, the model suffers around 10% performance loss in F-measure on test set, but we're able to see the relative importance of each word by looking at the coefficients after inverse-transformed with PCA. The following figure is a word cloud containing the words with the highest coefficients (weights), that is, words that are the strongest spam indicators, where the font size of each word is positively correlated with its weight.

[19]Notice most of the examples listed in section 4.1 are replies, though.

Figure 11: words with the highest weights

The next word cloud is for words with the lowest weights, that is, words that are the strongest non-spam posts indicators.

Figure 12: words with the lowest weights

We can already observe the distinctive difference between the two word clouds at the first glance. The first one is mainly about **Samsung's top products** (*galaxy, nexus, note, sii*) and the **user experiences** (體驗, 看到, 覺得), while focusing on the **multimedia aspect** (照片, 拍照, 影片). On the other hand, the second word cloud is more about **seeking help** (問題, 解決, 無法), and involves more **polite words** (謝謝, 大大, 小弟) and **technicalities** (*rom,* 開機, 設定).

The previous bags-of-word features were based on only **contents** of the posts, but there is also much information lying in the **titles** of the threads, so we create another 50 dimension-reduced bags-of-word features based on the titles, and combine these with the contents ones to yield 200 features. We prefer not to have them mixed together because title and content may have distinct groups of 'spam keywords'.

	precision	**recall**	**F-measure**
test set	59.12%	51.44%	55.01%
test set*	56.16%	58.57%	57.34%

Table 15: content and **title** bag-of-words features

With the addition of **title bag-of-words**, a further improvement in F-measure can be seen.

The dimension-reduced bags-of-word features turned out to be surprisingly helpful. The model is able to accomplish **over 55%** in F-measure while the ratio of spam is only around 3% on the test sets for first posts. Compared to the random baseline, it boosts the F-measure by as much as **45%**, which implies that the contents of posts actually give some strong clues about whether a first post is spam. Although on the surface, each spam post looks rather unsuspicious on its own, collectively, spam posts put more **emphasis on certain topics**, in comparison with non-spam posts, and our model trained with bag-of-words features was able to exploit this distinction.

5.4.3 Content Characteristics

A set of features derived from basic characteristics of the contents of the post is introduced.

feature	description
n_all	number of characters used in the post
n_words	number of words in the post (segmented by *Jieba*)
n_lines	number of lines in the post
n_hyperlinks	number of hyperlinks in the post
n_img	number of images added to the post
n_emoticon	number of emoticons used in the post
n_quote	number of quotations from previous posts
p_digit	proportion of digits
p_english	proportion of English characters
p_punct	proportion of punctuation characters
p_special	proportion of non-alphanumeric characters
p_wspace	proportion of white space characters
p_immediacy	proportion of first person pronouns (e.g., 我, 咱)
p_ntusd_pos	proportion of positive words in *NTUSD*
p_ntusd_neg	proportion of negative words in *NTUSD*
p_emoticon_pos	proportion of positive emoticons
p_emoticon_neg	proportion of negative emoticons

Table 16: description of not-so-obvious feature names

In regard to the naming of these features, the **n_** prefix means **number of**, while the **p_** prefix means **proportion of** (divided by the number of characters in the post). Most features should be self-explanatory then.

We compute **symmetric KL divergence** to find out which features exhibit the most different distributions between spams and hams. The formula of symmetric KL divergence is:

$$D_{KL}(P_{spam}(f)\|Q_{ham}(f)) + D_{KL}(Q_{ham}(f)\|P_{spam}(f))$$

where P_{spam} and Q_{ham} are the distributions of the feature f under all spam posts and all non-spam first posts, respectively. The higher the symmetric KL divergence is, the more different the two distributions are, which makes the feature more useful in discriminating between spam and non-spam first posts.

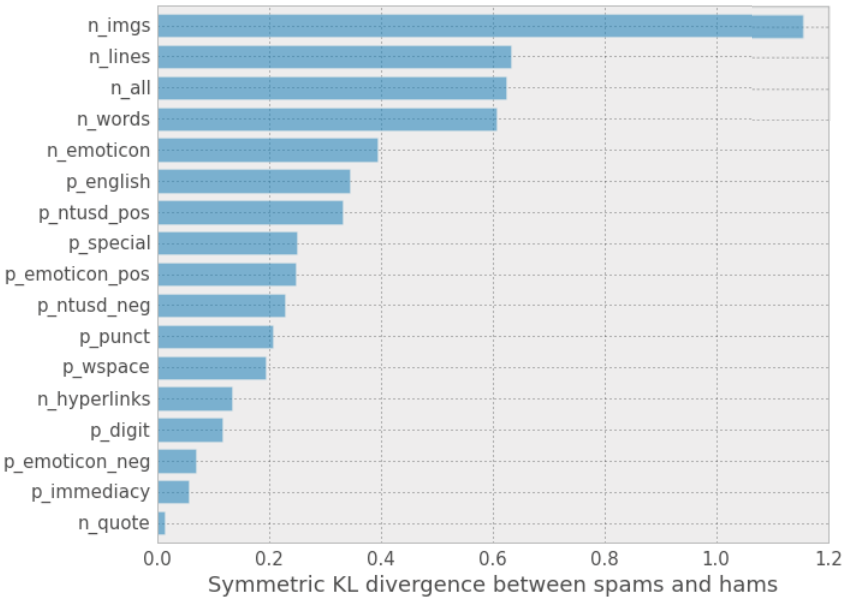


Figure 13: content characteristics features

The top four features that distinguish spam and non-spam first posts best are *n_all*, *n_imgs*, *n_words* and *n_lines*, which are all related to the **quantity** of content. This is not surprising because many of the spam first posts are essentially **advertisements in disguise** (e.g., unboxing posts and 'positive experience with a *Samsung* product' posts) and would generally use lots of words and pictures to showcase *Samsung* products in an attempt to impress people.

Figure 14: number of images in spam and ham

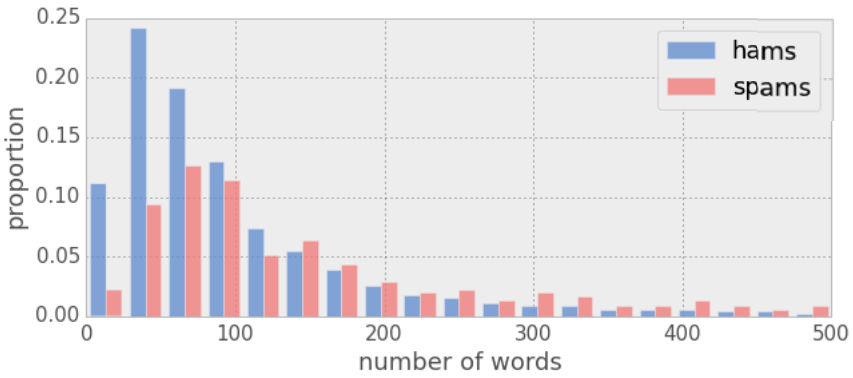


Figure 15: number of words in spam and ham

On top of the bag-of-words features, we add these 17 numerical features that characterize the contents of the first posts. The resulting performance is shown below. F-measure increases by about 3% on both test set and test set* respectively, so these features do seem to provide extra information that help detect spam first posts.

	precision	**recall**	**F-measure**
test set	73.05%	49.52%	58.79%
test set*	64.91%	52.86%	60.32%

Table 17: bag-of-words and **content characteristics**

5.4.4 Submission Time and Thread Activeness

We are done adding the content-centric features, so it's time to incorporate some **non-content-centric** features.

As discussed in section 4.5, spam posts have a tendency of being submitted more often during work time. To make use of this observation, we add a binary indicator feature for **each hour** in a day and **each day** in a week, in total $24 + 7 = 31$ new features. When the post was submitted during the hour or the day a feature corresponds to, then its value is 1; otherwise it's 0.

Moreover, we use **number of posts in the thread** started by the first post as another feature, which can serve as a measure of the activeness of the thread, as discussed in section 4.6.

	precision	**recall**	**F-measure**
test set	72.37%	52.88%	61.11%
test set*	66.67%	57.14%	61.54%

Table 18: bag-of-words, content characteristics, **submission time** and **thread activeness**

By incorporating these non-content-centric features, we see further improvement in F-measure on both test set and test set*.

5.4.5 Sentiment Scores Toward the Brands

The main objective of the covert marketing campaign is to promote a certain brand and sometimes denounce its competitor's brands in order to give it an unfair edge. Hence, we expect spam posts to show a **positive attitude** when it comes to *Samsung*, and possibly a **negative attitude** toward the competitors.

We devise a simple method to capture the sentiment toward brands in posts. Basically, we just add up the polarity of **sentiment words** in *NTU sentiment dictionary (NTUSD)* (Ku et al., 2006) and **emoticons** near mention of a brand or a product. For preciseness, the pseudocode producing the sentiment scores is presented in Algorithm 1

The following table shows number of spam posts in training set by the polarity of our estimated sentiment scores toward the brands. The result is not what we desired, since there are many posts with negative polarity toward *Samsung*, and many with positive polarity toward *HTC*. Even worse, the #positive/#negative ratio of *Samsung* is actually lower than *HTC*.

brand	**positive**	**negative**	**neutral**	**no mention**
Samsung	504	312	379	688
HTC	110	62	111	1600

Table 19: number of spam post with different polarities

With the sentiment scores toward *Samsung* and *HTC*, instead of showing any improvement, the F-measure dropped a little on both test set and test set*.

Algorithm 1 Compute Sentiment Score Toward the Brands

```
1  function AllBrandsSentimentScores(content)
2      for β ← [Samsung, HTC, ...] do
3          scores[β] ← BrandSentimentScore(content, β)
4      return scores

5  function BrandSentimentScore(content, β)
6      B ← list of aliases of β                          ▷ manually collected
7      P ← list of aliases of β's products               ▷ described in section 3.3
8      score ← 0
9      for α ← B ⨁ P do                                  ▷ longest aliases first
10         if α is in content then
11             S ← the sentence containing α plus the next one
12             score ← score + SegmentSentimentScore(S)
13     return score

14 function SegmentSentimentScore(S)
15     pw ← #(NTUSD positive words in S)                 ▷ longest matches first
16     nw ← #(NTUSD negative words in S)                 ▷ longest matches first
17     pe ← #(positive emoticons in S)
18     ne ← #(negative emoticons in S)
19     score ← pw − nw + pe − ne
20     return score
```

	precision	recall	F-measure
test set	70.97%	52.88%	60.61%
test set*	65.57%	57.14%	61.07%

Table 20: bag-of-words, content characteristics, submission time, thread activeness, and **sentiment scores toward the brands**

We postulate that spammers might put more effort into the promoting **the latest products**, because those are also the ones that are being promoted through proper ways of advertising. Hence, we make a variation of the algorithm to only account for the mention of products whose release date is **within one month** from the submission time of the post. More precisely speaking, the *line 6* from Algorithm 1 should be skipped, and the right hand side of *line 7* should be modified to be 'list of aliases of $\beta\acute{s}$ products which are released within one month from the submission time of *content*'.

Still, it shows no sign of improvement on top of the existing features.

	precision	recall	F-measure
test set	72.03%	49.52%	58.69%
test set*	64.41%	54.29%	58.91%

Table 21: bag-of-words + content characteristics submission time, thread activeness, and **sentiment scores toward the hot products**

There are some viable explanations of why the polarity of our estimated sentiment score fails to reflect the true opinion polarity. First, as discussed in section 2.1.4 and 4.1, the spam posts are carefully written to subtly deliver the messages, so they might to some degree avoid using sentiment words. Second, **sarcasm** is heavily used on *Mobile01*, which even some human readers often can't fully grasp. Third, *NTUSD* is not specifically designed for *Mobile01*, and has been there for some years, while the community on *Mobile01* many have given some words new meanings, and even invented new words in their subculture.

To further investigate it, we list concrete examples[20] of which our algorithm failed to grasp the true sentiment, where lime green background is used to

[20]Since we're not going to repeat this discussion on spam detection for replies, the examples includes both first posts and replies.

to indicate *HTC* brand/product mentions, and blue background for *Samsung* brand/product mentions; positive words or emoticons near a mention are signified by red background, while light blue background signifies the negative ones. Segments surrounded by the '|' symbols represent emoticons on *Mobile*.

***Samsung* → −2 *HTC* → 0**

|orz| 只能說 S2 眞的是 怪物 阿!!

⋮

This posts used negative emoticons and words to compliment a *Samsung* product in a dramatic manner.

***Samsung* → +3 *HTC* → 0**

我比較 推薦 Note 耶
因爲 htc 好像把 XL 當精品賣.. 哈
現在單核心的手機還敢賣那麼貴的.. 而且還很多人讚賞
眞的只有 HTC
XL 眞的不錯啦
不過考量到一支手機可能要使用個一兩年的時間
我還是比較 推薦 NOTE
除了筆的功能很 方便
到時要是出現 5.0 的系統.. 單核心不夠跑怎麼辦

The algorithm successfully detect the positivity toward *Samsung* based on the positive words near the two mentions of a *Samsung* product. However, to recognize sarcastic mockery of *HTC* is out of reach for this simple algorithm.

***Samsung* → −3 *HTC* → 0**

我也覺得吵這些要適可而止了
現在全世界有幾個國家像台灣這樣送一堆東西的

到時所有人把 三星 嚇到 不敢送 東西
吃虧的還 不是 我們自己???
眞的把事情鬧大了.. 只有爽到現在.. 卻苦到以後買手機的人阿

Samsung $\rightarrow -2$ ***HTC*** $\rightarrow 0$

好久 沒有 攻擊 三星 了
今天又來一篇
該領錢下班囉!

Samsung $\rightarrow -1$ ***HTC*** $\rightarrow 0$

印象中是之前 三星 手機就有的功能！ 不是 新功能！
不過還蠻方便的，隨時掌握朋友的生日，也能增加彼此的話題嘛！

In these examples, some negative words are around '三星 *(Samsung)*', but **no actual negativity** toward *Samsung* was there.

Samsung $\rightarrow +1$ ***HTC*** $\rightarrow +2$

手中的 感動機 用了也快一年,
之前是因爲很 喜歡 HTC 的 sence
感覺很質感
不過這次的 Galaxy nexus 的介面
整個很 吸引 到我
而且我眞的覺得 4.65 吋才是最適合的大小吧
我玩過 XL 也還 OK, 只是不愛那白色帶點紅的設計
快點上市吧!! 等不及啦

In this example, the spammer actually praised a *HTC* product at the start of the post, but then claimed that a *Samsung* product is even better. Sentiment polarity toward both brands are accurately identified (both positive), but recognizing the **comparison** is the critical here.

Sentiment/attitude toward the relevant brands is definitely an aspect that can be exploited to help the detection of spam posts. However, as demonstrated by these examples, a more advanced algorithm is needed.

5.5 Spam Detection for Replies

Following the discussion for first posts, now we consider spam detection for replies. For 'neatness' and a consideration in section 5.5.5, we remove all replies in the same thread as any reply in the training set from the test set. The spam counts and ratios for replies are listed in the table below.

	#spam posts	**#all posts**	**spam ratio**
training set	1337	148481	0.90%
test set	1020	67025	1.52%
test set*	343	25165	1.36%

Table 22: only considering replies in thread

The ratio of spam posts for replies is even lower than for first posts, so this time for downsampling, we randomly pick as high as 90% of the non-spam posts to remove from the training set.

5.5.1 Random Baseline

The performance of random baseline for replies is worse in comparison with the random baseline for first posts, which reflects the fact that spam ratio is much lower for replies, as observed in section 4.4.

	precision	**recall**	**F-measure**
test set	1.47%	48.33%	2.85%
test set*	1.38%	50.15%	2.68%

Table 23: random baseline for spam detection for replies

5.5.2 Bag-of-words

Here, we repeat the procedure in section 5.4.2. The optimal number of dimensions to reduce the bag-of-words features to is 250, according to the result F-measure of 5-fold CV.

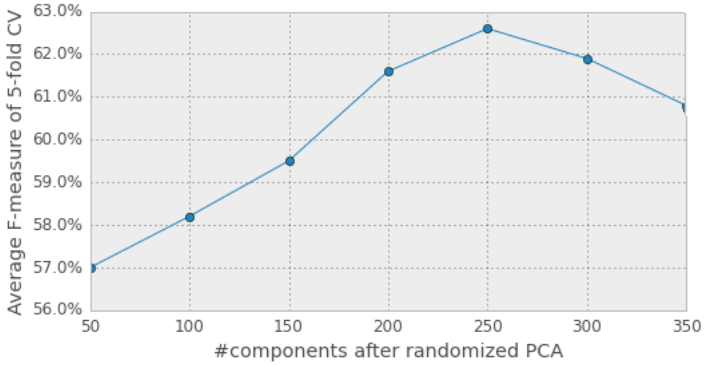


Figure 16: F-measure values as #component in PCA changes

The evaluation result does not looks nearly as nice as bag-of-words for first posts. Also, the performance on test set is significantly better than on test set*, so pattern of writing habits of the users might partially contribute to the 20.29% F-measure on test set.

	precision	**recall**	**F-measure**
test set	17.31%	24.51%	20.29%
test set*	11.85%	19.83%	14.83%

Table 24: content bag-of-words features only (250 components)

To explain such discrepancy between the performance with bag-of-words features for first posts and for replies, in addition to the fact that spam ratio is lower and content is less for replies, as observed in section 4.4, many of the spam replies are the **vacuous** ones with the sole intention of keeping the discussion in the thread alive to attract more attention to the thread, which is probably started by a spam first post, as mentioned in section 4.1. They are **concise** and contain **little to no opinion** on the brands, so it would be very hard to distinguish them from non-spam posts. On the other hand, because first posts are the very posts that **initiate the threads**, obviously it cannot be used for such 'keeping a thread alive' purpose.

This time, adding title bag-of-words features doesn't help, which is probably due to the fact that title is per thread (also per first post), rather than per reply. A title is shared by all the replies to the thread.

	precision	recall	F-measure
test set	15.60%	26.47%	19.63%
test set*	10.28%	22.16%	14.05%

Table 25: content and title bag-of-words

In the subsequent experiments, title bag-of-words features won't be incorporated.

5.5.3 Content Characteristics

As in section 5.4.3, we compute the *symmetric KL divergence* of the content characteristics features for replies to find out which of them are the most useful ones.

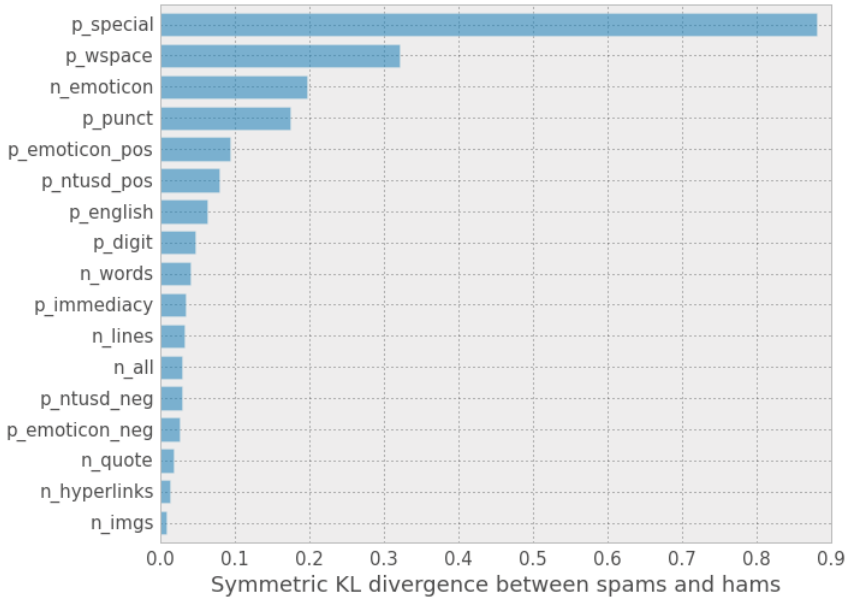


Figure 17: content characteristics features

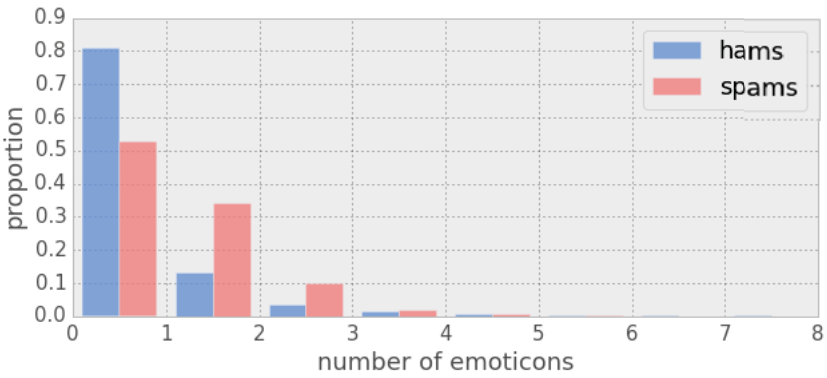


Figure 18: number of emoticons in spam and ham

Interestingly, replies seem to use more emoticons in general. For what it's worth, our explanation is that in spam replies it's more often to have either positive or negative attitude explicitly expressed to side with the previous 'pro-Samsung' posts[21], compared to non-spam replies.

With the addition of content characteristics features, the performance is slightly improved.

	precision	**recall**	**F-measure**
test set	17.70%	25.20%	20.79%
test set*	12.32%	19.83%	15.20%

Table 26: bag-of-words and **content characteristics**

5.5.4 Submission Time, Thread Activeness and Position in Thread

As in section 5.4.4, non-content-centric features indicating the hour/day of the submission time, and the number of posts in the thread in which the reply is, are added. Moreover, the two attributes about the position of a post (reply) in the thread, *nfloor* and *pnum*, as described in table 2, are also incorporated as non-content-centric features here.

According to the evaluation result shown below, these non-content-centric features seem to be quite helpful in spam detection for replies.

[21] which is often the first post in the thread, but could also be some previous replies.

	precision	recall	F-measure
test set	19.66%	30.98%	24.06%
test set*	14.65%	25.07%	18.49%

Table 27: bag-of-words, content characteristics, **submission time**, **thread activeness** and **position in thread**

5.5.5 Spamicity of the First Post in the Thread

When a thread is started by a spam first post, we could envision more spam activities to follow (as spam replies to the thread), due to the collusion between spammers discussed in section 4.7.

To measure the **spamicity**[22] of the first post in the thread in which the reply is, here we leverage our model for spam detection for first posts by using its probabilistic prediction on the first post in the thread as an additional feature.

For obvious reasons, the model we leverage here is the best model we have for first posts spam detection, that is, *SVM with RBF kernel* trained with dimension-reduced bag-of-words (contents + titles), content characteristics, submission time, and thread activeness features, whose performance is shown in table 18.

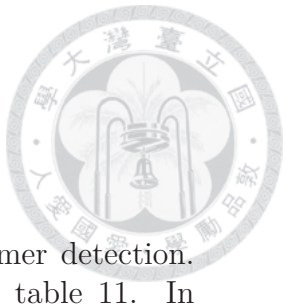
It would be problematic if a thread has its first posts in training set, and a reply in test set. Fortunately, we already removed the replies that could cause such problem from the test set, as mentioned in section 5.5.

By leveraging our best model for spam detection for first post, we see obvious improvement in performance.

	precision	recall	F-measure
test set	25.59%	29.61%	27.45%
test set*	21.10%	26.82%	23.62%

Table 28: bag-of-words, content characteristics, submission time, thread activeness, position in thread, and **spamicity of the first post in the thread**

[22]The word **spamicity** refers to concept of 'degree of spam'. In our paper, we use our trained SVM model's probabilistic output to measure spamicity. The closer it is to 1, the more likely the input instance is spam.



5.6 Spammer Detection

In this section, we set out to conduct experiments on spammer detection. Basic counts of the training set and test set were listed in table 11. In the later experiments, we downsample the non-spammers in training set by randomly removing 60% of them.

5.6.1 Random Baseline

As in spam detection, we create an absolute baseline with random guessing.

	precision	**recall**	**F-measure**
test set	0.91%	45.88%	1.78%

Table 29: random baseline for spammer detection

5.6.2 Profile Information

There are six numerical attributes[23] in user profiles on *Mobile01*, as introduced in table 6. Similar to section 5.4.3, we measure their usefulness in distinguishing the spammers from non-spammers by computing symmetric KL divergence:

$$D_{KL}(P_{spammer} \| Q_{hammer}) + D_{KL}(Q_{hammer} \| P_{spammer})$$

where $P_{spammer}$ and Q_{hammer} are the distributions of an attribute under all spammers and non-spammers, respectively.

[23]Here we don't deem the registration time and the last login time as numerical attributes.

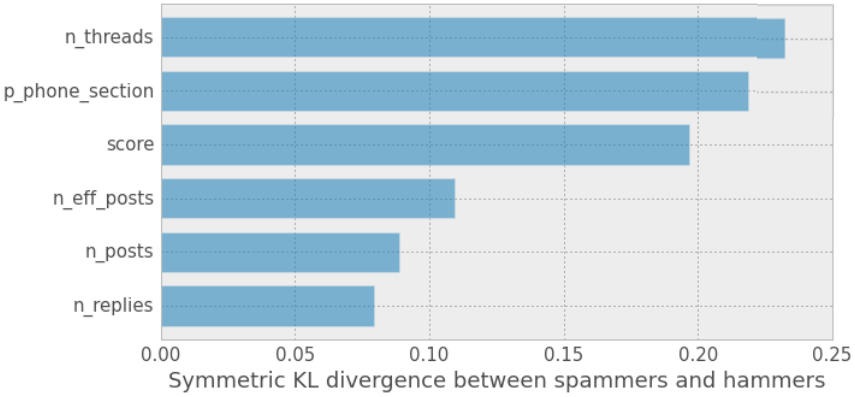


Figure 19: measuring usefulness of the features from profiles

Spammers are the more productive/reputable posters according to the number of threads they made, the *scores*, which was described in table 6, they have, and etc. It echoes the observation in section 4.3 that some spammers are reputable writers hired to make a handful of spam posts. Furthermore, on most web forums in general, there would be lots of 'lurkers'[24] who regularly login and read posts, but barely participate in discussions. It might be another factor that contributes to such difference.

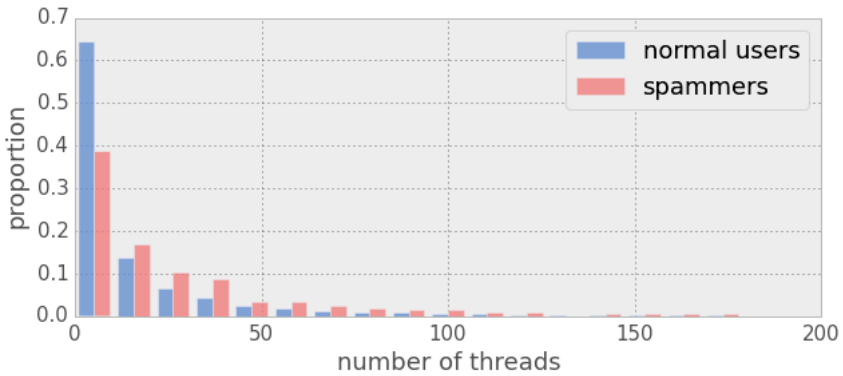


Figure 20: number of threads of spammers and non-spammers

[24]'潛水者' in Chinese

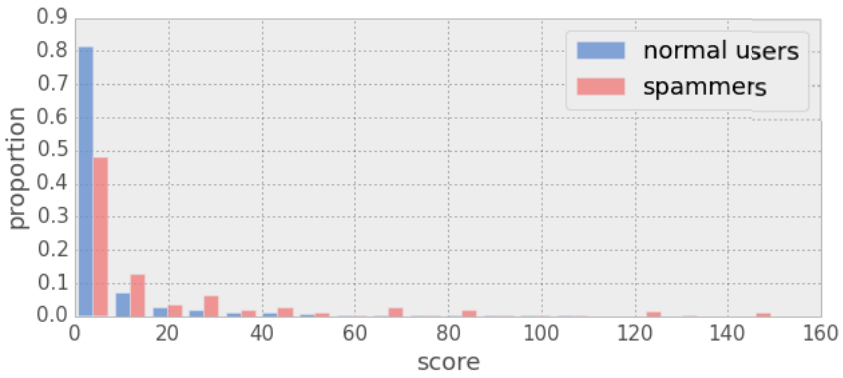


Figure 21: 'score' of spammers and non-spammers

Using these attributes in user profiles as features, the F-measure increased, compared with the random baseline. Still, profile information alone seems to be far from sufficient for our model to distinguish spammers from non-spammers.

precision	recall	F-measure
2.75%	22.62%	4.91%

Table 30: **profile attributes as features**

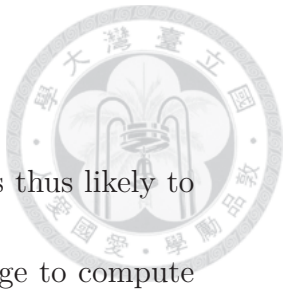
5.6.3 Maximum Spamicity of the First Posts of the User

Similar to section 5.5.5, we leverage the model from the spam detection for first posts to build a feature from its quality spamicity estimates for first posts.

A new feature ***max_spamicity_fps***[25] is computed by taking the **maximum** of the spamicity estimates of all first posts submitted by the user. Because the adopted definition of spammer is 'whoever makes one or more spam posts', taking the maximum is a more sensible choice, compared to taking the mean or the median.

The fact that our best model for first posts spam detection happens to have a **higher precision** than recall is a plus here. If the model misses a spam first post of one spammer (high precision, low recall), there's still a chance some other spam first post by the spammer could be detected. On the contrary, if the model misidentifies a non-spam first post by a normal user as spam (low precision, high recall), and gives it a high spamicity estimate, then the

[25] '**max**imum **spamicity** of the **f**irst **p**ost**s** made by the user'



value of *max_spamicity_fps* will be high for that user, who is thus likely to be misclassified as a spammer.

One thing we should be careful about is the model we leverage to compute the spamicity features shouldn't be trained with any spam first post by any user in test set, since this would almost guarantee a high spamicity estimate for that post, which results in a high *max_spamicity_fps* for that user in test set, without really knowing anything. Fortunately, with the data splitting method described in section 5.2, the training set of *posts* won't contain **any post** by any user in test set of *users*, let alone a spam first post.

By adding the *max_spamicity_fps* feature, the performance is significantly improved by almost 50% in F-measure. Leveraging the model for first posts really pays off here.

precision	**recall**	**F-measure**
61.02%	42.86%	50.35%

Table 31: profiles and ***max_spamicity_fps***

For what it's worth, 54 out of the 84 (64.3%) of the spammers in test set has submitted a spam post that is a first post in thread. In principle, it should be the maximum number of spammers that could be identified with this feature.

5.6.4 Burstiness of Registration of Throwaway Accounts

In section 4.3, we observed most of the throwaway spammer accounts were registered in bursts. To make use of this observation, we devise the feature **burstiness_throwaway_reg**[26] that counts the number of throwaway spammers accounts which were **registered 20 days within** the registration of the respective account. When the value of this feature is high, it's an indication of being in a burst depicted in figure 3, so the user account in discussion is likely to be a (throwaway) spammer account.

Adding **burstiness_throwaway_reg** on top of the existing profile and *max_ff_preds* features, the F-measure increased by a little.

[26]'**burstiness** of **throwaway** accounts' **reg**istrations

precision	recall	F-measure
67.31%	41.67%	51.47%

Table 32: profiles, *max_spamicity_fps*, and ***burstiness_throwaway_reg***

5.6.5 Frequently Appeared Groups of Posters

We discussed the collusion between spammers, where they would make posts in the same thread as discussed in section 4.7. To detect and make use of the collusion between spammers, we apply **frequent itemset mining**, which is a widely applied technique in the field of *data mining*. In the popular 'shopping in supermarket' example, it finds the set of items which are frequently put into the same *basket* and bought together. Moreover, we could set a *support threshold* to specify how frequent is frequent enough for the itemsets to be selected.

Applying the shopping analogy to our scenario, user id of each post is an 'item', and every 30 posts in a thread forms a 'basket', so each frequent itemset would be a group users that frequently 'appeared together' in threads. In our experiments, frequent itemset mining is conducted on all threads in both training set and test set, with the help of the *Orange* (Demšar et al., 2013) library, and doesn't involve the use of ground truth.

Rather than being somehow incorporated as a feature in our model, the mined frequent itemsets are used to fuel a **smoothing process** on the prediction outputs of the **base model**[27]. In this smoothing process, for each 3-element frequent itemset[28], we add up the spamicity[29] of the users in it. If the sum is bigger than a **threshold**, then we predict **all users** in this 3-element itemset to be spammers. The pseudocode of the whole procedure is in algorithm 5.7.

With the described smoothing process, the F-measure reaches 64.60%, which doesn't look bad at all considering the ratio of spammers in test set is only 0.98%.

[27]The current model for spammer detection we're improving upon. Its performance was shown in table 32

[28]If there are bigger frequent itemsets, we simply take all 3-element subsets of them.

[29]probabilistic prediction output by our base model if the user is in test set, or just 0 or 1 according to the ground truth if the user is in training set

Algorithm 2 The whole process combined with the smoothing step

```
1  function FinalModel(users_train, labels_train, users_test, posts)
2      features_train ← ExtractFeatures(users_train)
3      features_test ← ExtractFeatures(users_test)
4      model ← MLprocedure(features_train, labels_train)
5      preds, probs ← MakePredictions(model, features_test)
6      freq_groups ← FindFrequentGroups(posts)            ▷ all posts
7      preds ← SmoothPredsFreqGrps(preds, probs, freq_groups)
8      return preds

9  function FindFrequentGroups(posts)
10     threads ← index the posts with their thread ids
11     baskets ← empty list        ▷ initialize the list of 'baskets' or itemsets
12     for t ← threads do
13         for β ← each 30 consecutive posters' ids in thread t do
14             append β to baskets
15     freq_groups ← FrequentItemsetMining(baskets)    ▷ 'Orange'
16     remove the groups with support < 3 from freq_groups
17     return freq_groups

18 function SmoothPredsFreqGrps(preds, probs, freq_groups)
19     for freq_group ← freq_groups do
20         for freq_triple ← each 3-item subset of freq_group do
21             sum ← 0                ▷ initialize the 'spamicity' of the triple
22             for uid ← freq_triple do
23                 if uid ∈ users_train then
24                     sum ← sum + labels_train[uid]          ▷ 1 if spammer
25                 else
26                     sum ← sum + probs[uid]      ▷ probablity of spammer
27             if sum > 1.2 then       ▷ 'spamicity' higher than a threshold
28                 for uid ← freq_triple do
29                     preds[uid] ← 1              ▷ predict all 3 as spammers
30     return preds
```

precision	recall	F-measure
67.53%	61.90%	64.60%

Table 33: profiles, *max_spamicity_fps*, *burstiness_throwaway_reg*, with the **smoothing process**

Again, the fact that our best model for spam detection for first posts happens to have higher precision than recall helps. It somehow causes the base model trained with *max_spamicity_fps* feature to have a higher precision as well. Since this smoothing process intuitively gears toward improving the recall, the lower recall of the base model leaves a big room for this process to boost the performance.

5.7 Caveat

In section 5.6.4, the burstiness feature checked (sort of) if the account was registered during the same period as some throwaway spammer accounts in the training set. The usefulness of such feature depends on the fact that our test set is **temporally contiguous** with training set. On the contrary, if the training set and test set were collected from 2010 and 2012, respectively, the one year gap may make such feature less useful, because there is a slimmer chance there is a throwaway spammer accounts in test set which was registered in the same batch as the ones in training set.

In section 5.6.5, when an account in a mined frequent itemset is in the training set, the algorithm directly look up the ground truth for its spamicity (line 24 in algorithm). Again, the process relies on the fact the accounts in training set and test set could be in the same group that frequently appeared together.

The thing is this whole dataset is very **tightly coupled** together, which makes some undesirable interpretations possible on our experimentation results. For machine learning purpose, we split the data mainly according to the temporal order as described in section 5.2, but there always seems to be some sort of irrelevant connection between training set and test set that could attribute the evaluation result to. For example, in spam detection, some posts in training and test set could share the same author, so we created test set* to eradicate an alternative theory that our model is capturing some user-specific but non-spam related behaviors, as discussed in section 5.2.1. Even then, different accounts might still be shared by the same actual posters. It's very difficult, if not impossible, to completely exclude all kind of irrelevant connections between training set and test set, and

100% attribute the performance of the model to it successfully capturing the essential pattern of this covert marketing campaign (or opinion spamming activities).

The temporal contiguousness mentioned in the first two paragraphs in this section is another manifestation of tightly coupled nature of the dataset. Instead of trying to exclude such factor, which might be impossible to do, we leverage it to help the detection of spammers. We just need to be aware of the fact that for our results to be meaningful, the temporal contiguousness assumption between the training set and the test set has to be there.

6 Future Work

Admittedly, most of the techniques we adopt in spam and spammer detection were rather basic, which might leave some room for improvement. In this section, we specifically point out some potentially fruitful directions that could be worked on in the future.

6.1 Sentiment/Attitude Shown in Posts

In our study, to make use of the prior knowledge that spam posts might show certain attitudes toward certain brands, we estimated the sentiment scores toward brands with a naive algorithm described in Algorithm 1. Although the result we got wasn't so positive, if a more advanced algorithm could be devised to successfully grasp deeper understandings of the posts, and thus output better sentiment score estimates toward the brands, most likely the detection performance can be further improved.

6.2 Interaction between Forum Posters

In comparison to product reviews sites like *Amazon* or *TripAdvisor*, the activities of users on a web forum, from which our dataset is extracted, involves more interactions between each other. It may be in an explicit form such as quoting a previous post or directly mention a poster's username, or in an implicit form that requires the algorithm to have a higher degree of understanding in natural language in order to recognize it. There might be certain patterns in the ways spammers interact with each other or with other posters that could be leveraged in spam detection for replies, which still has plenty room for improvement.

6.3 Integration of Spam and Spammer Detection

In our method for spammer detection, we utilize the maximum spamicity of first posts by the poster as a feature, where the spamicity is estimated by the model we devised for spam first post detection. Conversely, we could have also used the spamicity of the posters to help in detection of spam posts, but decided not to do this because it adds more complexity to the way the dataset should be split. Taking it a step further, having a united model instead of separately constructing one for spam posts and spammer may be ideal, because in essence, spam and spammer detection are just two different perspectives on the same problem.

7 Conclusions

In this work, we forayed into this case study of opinion spam, from which we unprecedentedly collected a dataset along with the 'true' ground truth. The dataset was organized in a *SQLite* database and could easily be reused (section 3). Mainly with features derived from the contents of the first posts, we were able to obtain decent results on opinion spam detection for first posts. Seemingly contradictory to the observation that spam posts are carefully written to avoid getting caught (section 4.1), our investigation demonstrated that spam first posts collectively put more focus on certain topics that are not that suspicious per se (section 5.4.2), and we also saw the unusually rich contents of spam first posts could also be a giveaway (section 5.4.3). On the other hand, performance has much more room for improvement for spam detection for replies. On the basis of a decent detection model for first posts, a quality model for spammer detection was constructed (section 5.6.3). In addition, leveraging the collusion between spammers significantly boosted its performance (section 5.6.5).

References

Fabrıcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgılio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

Enrico Blanzieri and Anton Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1):63–92, 2008.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2: 27:1–27:27, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, Miha Štajdohar, Lan Umek, Lan Žagar, Jure Žbontar, Marinka Žitnik, and Blaž Zupan. Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, 14:2349–2353, 2013. URL `http://jmlr.org/papers/v14/demsar13a.html`.

Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 171–175. Association for Computational Linguistics, 2012.

Stephanie Gokhman, Jeff Hancock, Poornima Prabhu, Myle Ott, and Claire Cardie. In search of a gold standard in studies of deception. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 23–30. Association for Computational Linguistics, 2012.

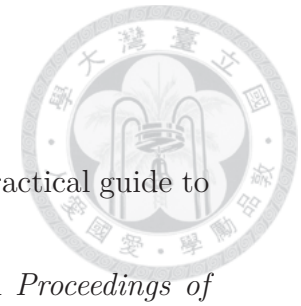
Zoltan Gyongyi and Hector Garcia-Molina. Web spam taxonomy. In *First international workshop on adversarial information retrieval on the web (AIRWeb 2005)*, 2005.

Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587. VLDB Endowment, 2004.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

C Harris. Detecting deceptive opinion spam using human computation. In *Workshops at AAAI on AI*, 2012.

Pedram Hayati, Vidyasagar Potdar, Alex Talevski, Nazanin Firoozeh, Saeed Sarenche, and Elham A Yeganeh. Definition of spam 2.0: New spamming boom. In *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, pages 580–584. IEEE, 2010.



Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

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. ACM, 2008.

Nitin Jindal, Bing Liu, and Ee-Peng Lim. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1549–1552. ACM, 2010.

Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384. ACM, 2005.

Andrew M Kaikati and Jack G Kaikati. Stealth marketing: how to reach consumers surreptitiously. *California Management Review*, 2004.

Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. Tagging heterogeneous evaluation corpora for opinionated tasks. In *Conference on Language Resources and Evaluation (LREC)*, 2006.

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. ACM, 2010.

M McCord and M Chuah. Spam detection on twitter using traditional classifiers. In *Autonomic and Trusted Computing*, pages 175–186. Springer, 2011.

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. ACM, 2011.

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. Association for Computational Linguistics, 2011.

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

James W Pennebaker, Cindy K Chung, Molly Ireland, Amy Gonzales, and Roger J Booth. The development and psychometric properties of liwc2007. *Austin, TX, LIWC. Net*, 2007.

Aldert Vrij, Ronald Fisher, Samantha Mann, and Sharon Leal. A cognitive load approach to lie detection. *Journal of Investigative Psychology and Offender Profiling*, 5(1-2):39–43, 2008.

Guan Wang, Sihong Xie, Bing Liu, and Philip S Yu. Review graph based online store review spammer detection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1242–1247. IEEE, 2011.

Sihong Xie, Guan Wang, Shuyang Lin, and Philip S Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 823–831. ACM, 2012.