


Who are the spoilers in social media marketing? Incremental learning of latent semantics for social spam detection

Long Song¹  · Raymond Yiu Keung Lau¹ ·
Ron Chi-Wai Kwok¹ · Kristijan Mirkovski² ·
Wenyu Dou³

© Springer Science+Business Media New York 2016

Abstract With the rise of social web, there has also been a great concern about the quality of user-generated content on social media sites (SMSs). Deceptive comments harm users' trust in online social media and cause financial loss to firms. Previous studies use various features and classification algorithms to detect and filter social spam on several social media platforms. However, to the best of our knowledge, previous studies have not exploited both probabilistic topic modeling and incremental learning to detect social spam on SMSs. Thus, the main contribution of this paper is design of a novel detection methodology that combines topic- and user-based features to improve the effectiveness of social spam detection. The proposed methodology exploits a probabilistic generative model, namely the labeled latent Dirichlet allocation (L-LDA), for mining the latent semantics from user-generated comments, and an incremental learning approach for tackling the

✉ Long Song
song.long@my.cityu.edu.hk

Raymond Yiu Keung Lau
raylau@cityu.edu.hk

Ron Chi-Wai Kwok
isron@cityu.edu.hk

Kristijan Mirkovski
kmirkovsk2@gmail.com

Wenyu Dou
wenyu.dou@cityu.edu.hk

¹ Department of Information Systems, College of Business, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, People's Republic of China

² School of Information Management, Victoria Business School, Victoria University of Wellington, 23 Lambton Quay, Wellington, New Zealand

³ Department of Marketing, College of Business, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, People's Republic of China

changing feature space. An experiment based on a large dataset extracted from YouTube demonstrates the effectiveness of our proposed methodology, which achieves an average accuracy of 91.17 % in social spam detection. Our statistical analysis reveals that topic-based features significantly improve social spam detection, which has significant implications for business practice.

Keywords Social spam · Spam detection · Topic modeling · Incremental learning · Machine learning · Big data

1 Introduction

Spam became prevalent in the late 1990s and early 2000s when email was considered to be the primary tool for information exchange among individuals and firms. With the introduction of email spam filters, spammers have started looking at other platforms for better payoffs. One of these “money-making” platforms for spammers are social media sites (SMSs) that play an increasingly important role in our daily lives [1]. Nowadays, online social media data exhibits the 4Vs characteristics that are often used to describe Big Data, namely volume, velocity, variety, and veracity [2]. In terms of volume, the number of active users on Facebook and Twitter has reached, respectively, 1.55 and 0.32 billion in November 2015.¹ There are over 500 million tweets generated on a daily basis.² Besides signifying its importance in our daily lives, the features of online social media (i.e., creation and exchange of user-generated content, support for collective actions, and facilitation of diverse social interactions) denote its indispensable function of being a business tool for promoting e-commerce products and services. Thus, increasingly more e-commerce retailers choose online social media as a main marketing platform or a new “social CRM” tool that fosters instant interactions with potential consumers.

SMSs likewise provide spammers with unprecedented opportunities to launch various attacks. Spammers perform deceptive acts [3, 4], conduct unfair trading activities [5, 6], and even make illegal profits [7] by posting social spam on SMSs. Social spam refers to low-quality information for which users do not ask or specifically subscribe to [8]. Social spam is used to launch phishing attacks [9], promote adverse websites [10], distribute malwares [11], and spread adverse messages [12, 13]. Embedded URLs in social spam direct users to adverts, malware, or pornographic websites (see Fig. 1). According to Nexgate’s state of social media spam report, there has been a 355 % growth of social spam in the first half of 2013 [14]. As well it has been revealed that more spammers were found on Facebook and YouTube than any other SMS. Grier et al. [15] reported that 8 % of the 25 million URLs posted on Twitter were phishes, malware, and scams.

The embedded URLs or deceptive contents in social spam gradually compromise consumers’ trust, patience and satisfaction, and even worse, lead to leakage of personal information or monetary loss [16]. A survey conducted by Maritz Research

¹ <http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>.

² <http://www.internetlivestats.com/twitter-statistics/>.

3dragondude 3 years ago

This has been flagged as spam [hide](#) • [Not Spam](#)

if anyone is interested in getting free ps3 games, xbox 360 games, wii games or any other game free then visit
vgforfree . blogspot . com

Reply ·   in reply to [Hailey Alana](#) ([Show the comment](#))

mrb4477 3 years ago

This has been flagged as spam [hide](#) • [Not Spam](#)

go up to your address bar, put the letter "Q" between the words you and tube and then press enter or click go



Reply ·  

Fig. 1 Snapshot of social spam about Starbucks found on YouTube

revealed that out of 3400 individuals who referred to review sites, such as Yelp and Trip Advisor, only half trusted the reviews they had read [17]. Moreover, the Federal Bureau of Investigation and the American National White Collar Crime Center report that the monetary losses from scam websites reached \$240 million in 2008 [18]. Consequently, consumers' trust in user-generated content on SMSs is decreasing.

Businesses leveraging social media to promote their products or services to consumers lose potential sales due to the fact that the competitors may take advantage of social spam to inflate their brand popularity. According to a research conducted by Harvard Business School, one star increase of restaurant's rating on Yelp leads to around 9 % increase in the revenues of the corresponding restaurant [19]. In fact, generating social spam has already been proven as an effective revenue stream for search engine optimization and public relations firms. For example, they intentionally post positive reviews or comments on SMSs to improve the reputation of their clients and in the same manner act adversely to deteriorate the reputation of their clients' competitors.

SMSs, such as Yelp, Facebook, and YouTube, carry the burden to filter and prevent social spam to enable consumers and firms to find accurate information or extract accurate market intelligence. Recently, CNN reported that TripAdvisor (Italy) was levied a fine of €500,000 by the Italian Competition Authority (ICA) for unfair trade activity and misleading consumers, despite the fact that certain review filters had been implemented.³ In 2013, Twitter spent approximately \$700,000 to prevent social spam. However, the existing spam filtering methods seem ineffective given the large number of spamming cases reported in press in recent years; they are far from perfect.⁴ Furthermore, social spam leads to a huge waste of system resources, such as the bandwidth and disk space, which is a severe problem in this era of Big Data.

Previous studies use different features (e.g., user-, text, graph-, and social network-related attributes) and classification algorithms (e.g., Naïve Bayesian and

³ See "\$611,000 fine as TripAdvisor gets bad review in Italy" by Barry Neild, Dec' 14, available at <http://edition.cnn.com/2014/12/23/travel/tripadvisor-fine/>.

⁴ See "Fake online reviews: 4 ways companies can deceive you" by Megan Griffith-Greene, Nov' 14, available at <http://www.cbc.ca/news/business/fake-online-reviews-4-ways-companies-can-deceive-you-1.2825080>.

Bayesian Network) to design frameworks for detecting and reducing social spam on many social media platforms (e.g., Facebook, Twitter, Sina Weibo, Myspace, YouTube, and Flickr). However, to the best of our knowledge, previous studies have not exploited both probabilistic topic modeling and incremental learning for detecting social spam on SMSs. Our main contribution is design of a novel methodology that integrates word-, topic- and user-based features, and applies labeled latent Dirichlet allocation (L-LDA) and incremental learning to improve the accuracy of social spam detection on SMSs. More specifically, the proposed computational method is underpinned by incremental learning and L-LDA [20] to mine latent topics describing the inherent semantics of social spam. The L-LDA is a supervised variant of latent Dirichlet allocation (LDA)—a topic modeling method originally developed by Blei et al. [21]. This probabilistic topic modeling assumes that each document is a mixture of various topics and each topic is characterized by a set of words with a high probability of co-occurrence. The latent topics can be transformed to topic-based features for classification and combined with word-based features.

Our proposed methodology applies the most discriminative words identified by a χ^2 test to label topics. We perform a comparison among the three types of features, namely, word-, topic-, and user-based features through rigorous empirical experiments on a dataset of YouTube comments. Furthermore, we empirically verify the interpretability and discriminatory power of each spam topic extracted through the L-LDA model, which has not been reported in literature before.

The rest of the paper is organized as follows. In section two, we review previous studies on features and classifiers for social spam detection, topic modelling for spam detection, and incremental learning to identify the research gap. Section three and four illustrate the proposed methodology for social spam detection and present the research hypotheses. Section five describes our experiments that were conducted based on a dataset of YouTube comments. In section six, we discuss the results. Finally, in section seven, we conclude this paper by elaborating on our contributions and proposing directions for future research.

2 Related work

2.1 Features and classifiers for social spam detection

State-of-the-art machine learning algorithms, especially those incorporating supervised learning techniques, are the most common practices for detecting social spam. The general procedures of these practices include: (1) extracting features from spam messages; (2) applying the extracted features to train a classifier; and (3) performing a classification through the trained classifier. Previous studies have adopted a variety of features and classification algorithms for social spam detection on SMSs. Markines et al. [22] developed six kinds of features for detecting social spam: (1) *TagSpam*—measures the tags' use and combinations that are statistically unlikely to appear in legitimate posts; (2) *TagBlur*—measures the semantic blur such as number of high frequency tags; (3) *DomFp*—estimates the likelihood that the content of a

tagged page is generated automatically through structural similarity with other pages and their body of annotations manually labeled as spam; (4) *NumAds*—measures the number of times an ad server appears in a Web page tagged by a user; (5) *Plagiarism*—measures the number of results returned by a search engine, excluding the originating resource’s URL; and (6) *ValidLinks*—measures the number of user profiles created for spam purposes. These features were integrated with AdaBoost classifier to capture the properties of social spam in a public dataset from BibSonomy.org.

Lee et al. [23] proposed a honeypot-based approach for detecting social spammers on SMSs. Initially, spammer behaviors on Myspace and Twitter were studied to develop a set of features (i.e., tweets similarity, material status, and number of friends), which were then empirically tested using 60 different classifiers with a bag-of-words model to represent the text-based features. Lastly, the developed classifiers were applied to datasets in-the-wild, which provided support for the effectiveness of social honeypots as social spam detectors. Wang et al. [8] developed a social spam detection framework for multiple SMSs such as Facebook, Myspace, and Twitter. The framework includes: (1) mapping techniques for converting network specific objects to framework-defined standard model of an object; (2) fast-path techniques, such as blacklists, hashing, and similarity matching, for pre-filtering by checking incoming objects against spam; and (3) classification technique, such as Bayesian, for classifying spam or non-spam objects. Associative classification was adopted to strengthen the cross social-corpora classification. Jin et al. [24] designed a social spam detection framework for Facebook, which uses GAD clustering algorithm for large scale clustering and integrated active learning algorithm for scalability and real-time spam detection. This framework has three types of features: (1) image content such as color histogram, color correlogram, CEDD, Gabor features, edge histogram, and SIFT; (2) text such as caption, description, comments, and URLs; and (3) social network such as user characteristics and behaviors.

Lin and Jia [25] adopted three types of features to detect social span on Sina Weibo⁵: (1) *lexical*—measures the difference in behaviors of spammers and legitimate users; (2) *status*—measures the outlink URLs, length of login, nature of topics, use of emotions, and reposting patterns; and (3) *user*—measures the number of user’s followers and users. The developed classifiers incorporated Naive Bayesian algorithm, logistic regression, and support vector machine (SVM). Dae-Ha et al. [26] adopted social network feature, such as request reject ratio, request acceptance ratio, personality commonness, same community, and friend’s friend, to train a Bayesian Network classifier for detecting social spam on SMSs. Po-Ching and Po-Min [27] applied a J48 decision tree algorithm to analyze features, such as URL rate and interaction rate, for detecting spam accounts on Twitter. Sureka [28] proposed an effective method for detecting social spam in YouTube comments, which mines activity logs of users to extract patterns such as average time difference between comments, percentage of comments, comment repeatability across videos, and comment repetition and redundancy.

⁵ A Chinese microblogging website (www.weibo.com).

2.2 Topic modelling and spam detection

Previous studies adopted various types of features—user-, text-, graph-, and social network-related attributes—to detect social spam. These attributes are low-level features in comparison to the high-level features such as topic-based features. Topic-based features generated from topic models serve as a more abstract representation of documents. However, the application of topic-based features to detect social spam has received relatively little attention from researchers. Indeed, spam detection performance might be improved by leveraging a high-level representation. Topic models, especially variants of the LDA model, are becoming increasingly popular in different research areas such as information retrieval [29], bioinformatics [30], and image classification [31]. Therefore, some researchers attempted to apply LDA for spam detection and extraction of user-generated opinions from SMSs. Bíró et al. [32] adopted the novel multi-corpus LDA, an extension of the classical LDA model, for web spam detection. LDA was also used to track the trend of online opinions. Cui et al. [33] proposed an incremental Gibbs sampling algorithm to train the LDA model, and hence to track and observe the trends of topics being discussed online. Sizov [34] developed a framework for Web 2.0 content characterization with spatial awareness, which integrates Bayesian statistical models to explicitly describe spatial coordinates jointly with tag co-occurrence patterns. The proposed model is an extension of the classical LDA model, which besides the LDA-like tag generation process also integrates topic-specific normal distributions to describe the location (i.e., latitude and longitude). The content categorization, clustering, and tag recommendation capabilities of the proposed model were tested on dataset from Flickr.

2.3 Incremental learning as a new learning paradigm for social spam detection

Social spam detection is an evolving phenomenon, which implies the constantly-changing nature of the underlying dataset and the extracted features. As new types of social spam constantly emerge, classifiers should be retrained using existing and new training examples. Obviously, classifier retraining is time-consuming as well as keeping all the available training data wastes a large amount of storage space [35]. Consequently, traditional machine learning techniques without retraining may not be effective for detecting social spam on SMSs. However, a new learning paradigm, such as incremental learning, might be a better solution. Incremental learning is a new machine learning paradigm in which a classification model can be refined based on new training examples rather than retraining the model using the entire dataset. In fact, there are several popular incremental learning algorithms such as candidate elimination [36, 37], Cobweb [38], ID5 [39], and ILA [40]. Furthermore, there are some traditional algorithms, such as SVM [41] and logistic regression [42], that can be extended as incremental classification models to meet the social spam detection requirements.

2.4 Research gap

Previous studies use various features (e.g., user-, text-, graph-, and social network-related attributes) and classification algorithms (e.g., Naïve Bayesian and Bayesian Network) to design frameworks for detecting social spam on SMSs (e.g., Facebook, Twitter, Sina Weibo, Myspace, YouTube, and Flickr). However, to the best of our knowledge, none of the previous work reported in literature has exploited both probabilistic topic modeling and incremental learning to detect social spam on SMSs. For our work, thus, we integrate word-, topic-, and user-based features, and apply L-LDA and incremental learning to enhance the performance of social spam detection. We also conducted a rigorous experiment on dataset from YouTube comments to test the effectiveness of our proposed framework for detecting social spam.

3 The proposed methodology for social spam detection

In this section, we illustrate the proposed methodology for social spam detection on SMSs (see Fig. 2). The proposed methodology leverages three types of features, namely, word-, topic-, and user-based features to detect social spam. χ^2 test and *tf-idf* scheme are adopted to conduct feature extraction and selection for the word-based feature. Discriminative words with high χ^2 scores are regarded as the topic

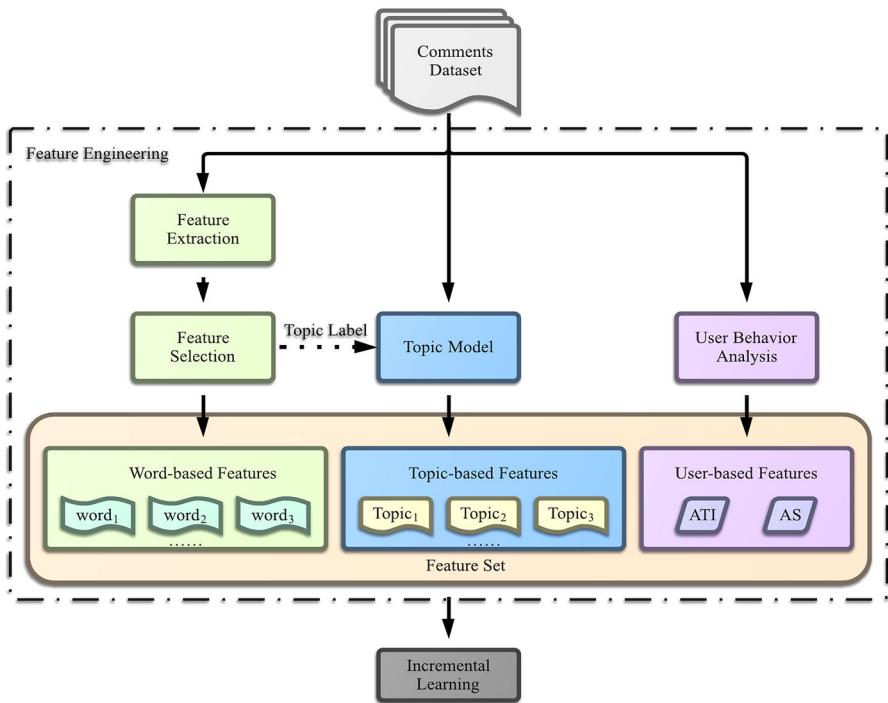


Fig. 2 The proposed methodology for social spam detection

labels which are taken as part of the inputs to the L-LDA model. L-LDA is then applied to extract latent topics from user-contributed comments such as documents. The normalized topic frequencies are subsequently extracted as the proposed topic-based features. User behavior analysis is conducted to compute each user's average time interval of posting (ATI) comments and the average similarity (AS) between two adjacent comments. These are taken as the proposed user-based features. Finally, a series of incremental classifiers are built to classify spam and ham such as legitimate user comments.

3.1 Feature engineering

Feature engineering is crucial to the performance of a classification task, and therefore we elaborately construct our feature set (see Table 1).

3.1.1 Word-based features

In the field of information retrieval, researchers attempt to identify effective means of representing documents from a collection and efficiently processing large collections. Accordingly, we adopt the famous vector space model as the foundation for several text processing tasks such as document clustering and classification. Meanwhile, we also exploit the inner statistical information for user comments as much as possible because this information is crucial for several tasks such as clustering, classification, information retrieval, and summarization (see Fig. 2).

To meet the aforementioned requirements, the popular *tf-idf* scheme [43] is adopted for document representation. The *tf-idf* scheme has some appealing properties. It offers a simple representation of documents by computing the weights of words appearing in a document. As a typical feature extraction method in text categorization [44, 45], features extracted based on this weighting scheme are discriminative and powerful for many classifiers such as KNN, SVM, and Rocchio [46, 47]. Therefore, we apply the *tf-idf* weighting scheme as modified by Singhal et al. [48] as the basis to represent word-level features:

$$w_{i,d} = \frac{1 + \ln[1 + \ln(tf_{i,d})]}{1 - b + b \times \frac{|d|}{avdl}} \times \log \frac{N + 1}{df_i} \quad (1)$$

where b is set to a default of 0.20, and $tf_{i,d}$ is the term frequency of term i in document d . $|d|$ is the document length; $avdl$ is the average document length across

Table 1 Proposed feature set

Type	Description	Quantity
Word-based	Weighted tf-idf: $W_1, W_2, \dots, W_{7921}$	7921
Topic-based	Topic frequency: T_1, T_2, \dots, T_k	$k = 6, 10, 20, 50, 80, 100$
User-based	ATI, AS	2
Total		7923 + k

Bold values indicate the best value in the table

the corpus. df_i is the number of documents that contains term i , and N is the size of the corpus.

Although the *tf-idf* scheme has been successfully applied to document representation, a huge corpus vocabulary tends to produce many sparse vectors under the environment of Big Data. These sparse vectors incur unnecessary computational costs and may hamper classification accuracy due to the large number of missing values along the high dimensional feature space. If we utilize an effective feature selection method to eliminate the uninformative words (features), we can improve the computational efficiency and potentially improve the classification performance. Accordingly, we apply the χ^2 test to conduct feature selection in the word set.

χ^2 test is one of the most widely used metrics for feature selection in text classification [49–51]. χ^2 test is used as “(1) a goodness-of-fit test between a group of data and a specific probability distribution, or (2) a test for the degree of dependence or association between two factors or variables” [52]. χ^2 test is grounded in the information theory, which “tries to capture the intuition that the best terms for the class c are the ones distributed most differently in the sets of positive and negative examples of c ” [53, 54]. After running a χ^2 test, the χ^2 score χ^2 is computed for each word. The value of χ^2 represents the association or dependency between the word and the spam class. The higher χ^2 score a word has, the more discriminative the word is. Since the size of the corpus vocabulary is very large, we select only the most distinctive words according to the χ^2 score to reduce the dimensionality of the feature space. This is one of the ways to alleviate the computational costs of Big Data.

3.1.2 Topic-based features

The proposed feature selection method can overcome some of the shortcomings, such as very sparse document vectors, of the *tf-idf* document representation approach. However, this method fails to reveal the intra-document statistical structure. Adopting the “bag-of-words” assumption allows documents to be viewed and decomposed from a micro-level, which implies that the semantic information among words may be lost. Therefore, a macro-level view of the documents is also considered to deal with such a problem.

The systemic functional linguistic theory (SFLT) is a mechanism for representing texts, and its language has three meta-functions such as ideational, interpersonal, and textual [55]. The ideational meta-function represents a theory of human experience that pertains to the aspects of “mental world,” including attitudes and desires [56], and can be applied to several types of information such as topics, emotions, and opinions [57]. When the representation of texts reaches this level, the semantic information among words can be extensively retained. Blei [21] made a noteworthy improvement in this area of research by proposing one of the earliest probabilistic topic models, namely LDA. This model can identify how certain topic patterns are mixed in a document. LDA assumes that the entire corpus has k number of topics, and the content of each document focuses on these k topics. A document is regarded as a mixture of topics with different probabilities, and a topic represents a

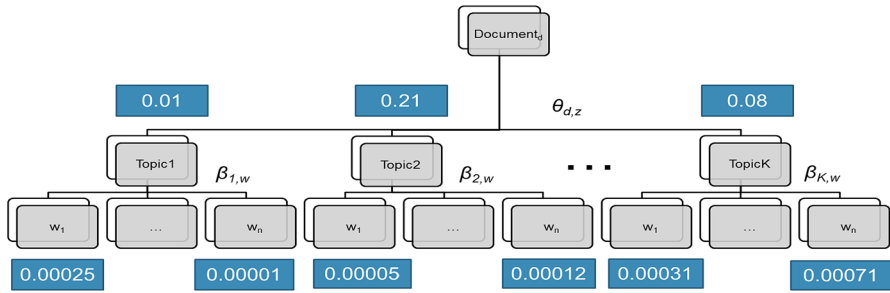


Fig. 3 An example of the hierarchical organization of a document

unique sequence of words based on their probabilities to occur (see Fig. 3). $\theta_{d,z}$ denotes the probability that topic z occurs in document d and $\beta_{z,w}$ denotes the probability that a word w occurs in a topic z .

L-LDA is a supervised variant of the LDA probabilistic graphical model [20]. It differs from LDA in the sense that it can automatically learn the latent topics in documents of a training set based on the given topic labels, and predict the occurrences of the defined topics in previously unseen documents of the test set. Assuming that we approximate a total of k topics of a corpus; for each document d , topic label $A_d = (l_1, l_2, \dots, l_k)$ will be generated in a certain manner. Each $l_k \in \{0, 1\}$ represents whether the document is related to the k th topic or not. Changing the value of the topic label for each document allows us to alter the relevance between the document and the topics of interest. Therefore, the topics obtained from running L-LDA are definitely relevant to the documents. To help interpret the meaning of topic labels, the topic label can be compared to a few centroids of topic clusters to help interpret the meaning of the topic model. Meanwhile, the L-LDA automatically clusters words around some centroids to form interesting topics (see Fig. 4). LDA and L-LDA differ from each other in terms of the topic mixture distribution— θ —that is decided by both binary topic presence indicators A_d and the topic prior α . In L-LDA, A_d can help filter topics sampled from a Dirichlet distribution over α and ensure the topics learned at the end could be mapped to the topic labels.

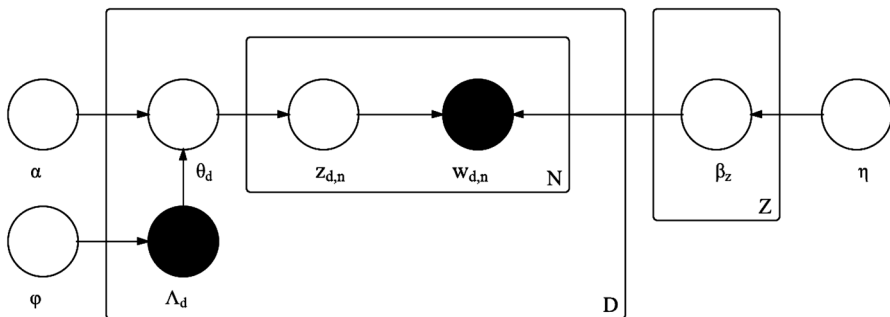


Fig. 4 Graphical model of L-LDA

The L-LDA model characterizes a document by a mixture of topics in which each topic has different probabilities and word distributions. Hence, two distributions exist: document-topic and topic-word distributions. It is assumed that both distributions to have a Dirichlet prior. For every document $d \in D$, the distribution θ_d on topic set Z is sampled from $\text{Dir}(\alpha)$. For each topic $z \in Z$, distribution β_z on vocabulary set V is sampled from $\text{Dir}(\eta)$. Hence, for the n th word in document d represented as $w_{d,n}$, topic assignment $z_{d,n}$ can be iteratively calculated with Gibbs sampling [58]. After obtaining topic assignment z , document-topic distribution $\theta_{d,z}$ and topic-word distribution $\beta_{z,w}$ are estimated as follows:

$$\theta_{d,z} = \frac{N_{d,z} + \alpha}{\sum_{z=1}^{|Z|} N_{d,z} + |Z|\alpha} \tag{2}$$

$$\beta_{z,w} = \frac{N_{z,w} + \beta}{\sum_{w=1}^{|V|} N_{z,w} + |V|\beta} \tag{3}$$

where $N_{d,z}$ represents the number of words assigned to topic z in document d , and $N_{z,w}$ represents the frequency of word w assigned to topic z in the corpus.

However, generating topic label A_d for each document and defining the topics of interest remain a challenging issue. This is the reason why L-LDA has not been widely used in prior research. We believe some topics should differentiate spam from ham in our corpus. Topics should have a certain level of discriminative power. Thus, χ^2 score χ^2 can be treated as a measure of the discriminative power of a word. A topic label for a document is generated based on the occurrence of the most discriminative words in that document. For document d , each l_n in its topic label $A_d = (l_1, l_2, \dots, l_k)$ can be

$$l_n = \begin{cases} 1, & \forall i \in d, n = \text{Rank}(\chi_i^2) \leq k \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\text{Rank}(\chi_i^2)$ indicates the rank of the χ^2 score of word i in document d . We can assume that the latent topics generated by L-LDA tend to be as discriminative as their centroids for spam detection.

Considering the discriminative words as topic labels for each document, a multi-labeled corpus can be successfully generated as an input to the L-LDA model. Then the L-LDA model will generate a topic assignment $z_{d,n}$ for each word i in document d and further arrive at $N_{d,z}$. For topic z , its frequency in document d can be represented by $N_{d,z}$. Then, we selected normalized topic frequency as the feature for spam detection. For document d , the normalized frequency of topic z can be computed as follows:

$$t_{d,z} = \tau + (1 - \tau) \frac{N_{d,z}}{\max_{z \in Z} N_{d,z}} \tag{5}$$

where τ is a value between 0 and 1. For document d , its topic-based feature is $T_d = [t_{d,1}, t_{d,2}, \dots, t_{d,k}]$.

3.1.3 User-based features

Identifying spammers can also contribute to detecting social spam since the majority of spammers share some behavior patterns. In some recent studies, researchers found that spammers tend to produce deceptive contents in a relatively short time interval, and their contents tend to be similar [28, 59–61]. It is suggested that spam accounts have a “bursty” property, indicating that spam comments can be densely populated in a short period of time [60]. One explanation for this phenomenon might be that spam messages are automatically created via bot accounts that reposting similar contents. Based on the latter, Gao et al. [61] extracted the absolute time interval between consecutive wall events to detect social spam on Facebook. Chen et al. [62] found that average interval of posts could achieve a high precision in detecting sock puppets on Chinese news websites. Besides, content similarity is also found to be effective in capturing fake reviewer groups on Amazon [63] and spammers on Twitter [23, 64, 65].

Accordingly, we take into account ATI and AS of two adjacent comments posted by the same user as our user-based features. For user u who post N_u comments, the ATI of user u is computed as follows:

$$\text{ATI}(u) = \frac{\sum_{k=2}^{N_u} (t_{u,k} - t_{u,k-1})}{N_u} \quad (6)$$

where $t_{u,k}$ represents the posting time of the k th comment of user u . Similarly, the AS of user u is defined as follows:

$$\text{AS}(u) = \frac{\sum_{k=1}^{N_u} \text{sim}(c_{u,k}, c_{u,k-1})}{N_u} \quad (7)$$

where $c_{u,k}$ is the k th comment of user u and function $\text{sim}()$ is a similarity function that computes the similarity between two comments. We use the approximate string matching algorithm that works by identifying the smallest number of edits required to change one string into another one [66, 67].

3.2 Incremental learning

Incremental learning, also called online learning, is a popular machine learning method. It allows classifiers to learn newly emerging training instances without going through the whole training set. It does not require a large number of training examples at the beginning and the performance of a classifier can continuously improve by learning some new training instances. Moreover, when the learning target keeps evolving in a dynamic environment, an incremental learning approach can better adapt to the changes and capture the new trend. Therefore, it is particularly suitable for complex or evolving tasks such as fraud detection [68] and spam detection [69, 70].

Incremental learning is performed in a sequential manner; at round t , a learner is given an instance \mathbf{x}_t from the dataset and then predicts an outcome p_t . Here, p_t refers to a machine-generated spam indicator for the instance of \mathbf{x}_t . Once the machine

Fig. 5 An incremental learning paradigm

```

Incremental Learning
INPUT:  $(x_t, y_t), t = 1, 2, \dots, T$ 
Initialize : Set  $w_t = 0$ 
for  $t = 1, 2, \dots, T$ 
    receive  $x_t$  //new instance
    predict  $p_t$ 
    compute loss  $l(p_t, y_t)$  //loss function
    update  $w_t$  as  $w_{t+1}$  based on the loss
end
OUTPUT:  $w_{T+1}$ 

```

predicts the outcome p_t , it will obtain a feedback by computing the loss function $l(p_t, y_t)$ to compare the outcome p_t and the manual spam label y_t . The loss function $l(p_t, y_t)$ is a measurement of the discrepancy between the machine-generated spam indicator p_t and the manual spam label y_t . Then based on the loss function $l(p_t, y_t)$, the hyperplane w_t for classification will be updated (see Fig. 5). It is worth emphasizing that when the classifier makes a prediction for t th trial, it will only use the instance x_t to adjust the current classifier in the previous trials order to get p_t . This makes it possible for the classifier to evolve together with the new batches of data instances. Also, for different algorithms, the updating rules for the hyperplane w_t are different.

In our work, we adopt four incremental algorithms to accomplish the learning task. The incremental algorithms include SVM [41], classical perceptron [71], relaxed online maximum margin algorithm (ROMMA) [72], and logistic regression [42]. For perceptron, SVM and logistic regression, the updating rule is mainly based on the stochastic gradient descent (SGD) method [73, 74]. The SGD method has several advantages when it is compared to the gradient descent (GD) method. For the SGD method, only one training example is used to estimate the gradient of the target function in each iteration. Thus, it will converge to the global minimum faster. This characteristic is very crucial for a large training dataset. For ROMMA or passive-aggressive perceptron algorithms [75], their updating rule is to obtain a hyperplane that can correctly classify the previously seen examples with a maximum margin. It only uses one example at a time to update the hyperplane. From the two aforementioned updating rules, we can see that the practice of using one example at a time to converge to the optimal classifier makes the incremental learning method outperform the traditional machine learning algorithms. The incremental learning method can easily capture spammers' new spamming patterns and tendency in the setting of evolving social media. As a result, it makes the proposed detection method continuously adapt to spammers' possibly changing behavior. Moreover, it alleviates the problem of classifier training from an extremely large dataset, particularly under the environment of Big Data.

For the extraction of topic-based features for the new data, we adopted the "folding-in" heuristic proposed by Hoffmann [76], which is also in line with the core idea of incremental learning. For each new comment, we first assigned a random topic for each word and kept the topic assignments for the old corpus unchanged. Then we ran Gibbs Sampling on this new comment and update the

document-topic distribution $\theta_{d,z}$ and topic-word distribution $\beta_{z,w}$ accordingly. In such a way, we were able to generate topic-based features for new data promptly and update our L-LDA model without retraining the whole model. Thus, our proposed framework operates under the core logic of incremental learning.

4 Research hypotheses

It should be noted that little research has been done to investigate how the aforementioned features affect the performance of classification. The feature set adopted in our framework includes three types: word-, topic-, and user-based features. Therefore, we present our hypotheses to evaluate these features' influence on the classification performance measured in terms of accuracy, precision, and recall. The overall accuracy measures the total percentage of correctly classified spam and ham. Precision is a measure for the percentage of correctly classified spam out of the spam set identified by the classifier. Recall assesses the detection rate of spam class, that is, the percentage of correctly classified spam out of the true spam set.

The topic-based features, which are obtained via the L-LDA model, can reveal the latent semantic structure of each social spam. We believe that adding this type of feature can increase the capability of the classifiers to detect spam based on the inherent semantics of spam contents. Therefore, we construct the following hypotheses:

H1a Incorporating topic-based features into the original feature set improves the *accuracy* of classifiers.

H1b Incorporating topic-based features into the original feature set improves the *precision* of classifiers.

H1c Incorporating topic-based features into the original feature set improves the *recall* of classifiers.

Here, the original feature set refers to the set without the aforementioned features. The user-based features reflect the behavior of spammers to some extent and this type of features is objective. Thus, user-based features may enhance the performance of the classifiers. Hence, we hypothesize:

H2a Incorporating user-based features into the original feature set improves the *accuracy* of classifiers.

H2b Incorporating user-based features into the original feature set improves the *precision* of classifiers.

H2c Incorporating user-based features into the original feature set improves the *recall* of classifiers.

The information provided by user-based features is typically external to the user comments, whereas topic-based features are internal of the user comments.

Nevertheless, internal features might be more effective in social spam detection due to the definition of social spam, which refers to a low-quality information that users do not ask for or specifically subscribe to on SMSs. The latter definition implies that social spam is better defined based on the content level. Hence, the topic-based features tend to be better than the user-based features. That is:

H3a Compared with user-based features, the topic-based features will help the classifier achieve better *accuracy*.

H3b Compared with user-based features, the topic-based features will help the classifier achieve better *precision*.

H3c Compared with user-based features, the topic-based features will help the classifier achieve better *recall*.

5 Experimental evaluation

5.1 Dataset

The dataset for experimental evaluation was made up of millions of YouTube comments about the most popular video clips. The comments were labeled as true spam or not, using either an official spam filter by YouTube or manually with the “Flag for spam” button available above each comment posted at a video page. We elaborately extracted the preceding features to classify labeled comments into spam or ham using incremental classifiers. Data collection began on October 31, 2011 [77, 78] (see Table 2).

The most appealing feature of the dataset was that the labels were created by the audience on YouTube when they were browsing the comments. Then, the spam filter on YouTube or the administrative staff of YouTube would verify the reliability of these labels. Although some spam comments were not tagged as spam because of the infeasibility of the manual verification of a large volume of comments, these errors were still acceptable for an experiment in terms of the large amount of traffic on YouTube every day. Moreover, except the label and comment, posting time, video number, and user ID were also included in each row of the dataset (see Table 3). Spam is regarded as a low quality information that is not subscribed by the user such as adversarial contents for online shopping sites or phishing sites.

Table 2 Properties of the dataset

Properties	Values
Videos	6407
Total comments	6,431,471
Comments marked as spam	481,334
Total users	2,860,264

Table 3 Spam examples in the dataset

Number	Spam
1	Three most cool things in the World for me before 1)))) Jordan—the super star 2)))) 66cheap.com—the cheapest shopping site 3)))) the iphone – best connector NOW THERE’S ONE MORE, IT’S THE VIDEO ABOVE!!!!!!!!!!!!
2	Three Best things in the World for me now:):):):) :) 1. Lucas—My boyfriend! 2. 55cheap. com—the cheapest shopping site 3. the video above— the most ironical and interesting video I think:]:]:]:] :]:] :]:]
3	I just earned 38\$ dollars, by using AppRedeem. You can use it by using your iPod, iPhone or iPad. You download apps and gets instant money to your paypal account!(You can delete them after). Go to the website: “m.AppRedeem.com”, must be on your iPod, iPhone or iPad. Replace the “,” with dots, (.) Also Works with android. To get your 38\$ dollars first time of use, use bonus code: “MyiPad”!

5.2 Experimental design

5.2.1 Pre-processing

YouTube is a popular video streaming website on which people from different countries share video clips and post comments. According to the New York Times, YouTube is the second most searched website in the world [79], and approximately 100 h worth of video clips are uploaded every minute [80]. Basically, user comments are written in multiple languages. Our main focus is to detect spam contents written in English. Hence, the first step in the pre-processing stage was to filter non-English comments that left us with a total number of 3,492,590 English comments, including 304,092 labeled spam and 3,188,498 ham.

Then, some users’ comments were removed from the dataset because their total number of posted comments was smaller than a threshold. If a user posted less than the minimum number of comments, the user was excluded from the dataset. The features that were extracted include comparison of two adjacent comments by the same user, which to some degree reflect the long-term activity of a user. If the user posted too few comments, the discriminative power of the corresponding feature tends to be low. Besides, data preprocessing is a way to identify task-relevant data as well as reduce noisy and low-quality data [81]. For user group U_n (group of users each of whom only post n comments), if n is too small those groups’ comments will provide less information than those with a larger value of n . Thus, we regard those groups of data as low-quality ones which should be eliminated in preprocessing. Here, we exhibit the dataset statistics before and after preprocessing (see Table 4). Thus, to make use of high-quality data for computation, we only reserve those data generated by user group U_{6+} in which each user at least has posted 6 comments. In the end, we have 78,965 users and 6240 videos in our cleansed dataset.

The training and test sets should be separated because we adopted a machine learning method for classifying spam. The two sets were separated using a time sequence of each user posting the comments. Further, we extracted the first two-

Table 4 Dataset Statistics before and after preprocessing

Dataset	Original	English	Filtered	Training set	Test set	
Total comments	6,431,471	3,492,590	1,055,375	724,569	330,806	
Comments marked as spam	481,334	304,092	210,283	142,965	67,318	
Comments marked as ham	5,951,037	3,188,498	845,092	581,604	263,488	
User Groups	U ₁	U ₂	U ₃	U ₄	U ₅	U ₆₊
Comments (%) in English comments	32.14	16.44	9.82	6.48	4.90	30.22

thirds of the comments of each user as the training set and the remaining one-third as the test set based on the time stamp on each comment. In other words, the test set incorporated a time lag behind the training set. Under this setting, it is convenient for us to measure the performance of our proposed framework to predict the extent of spam comments in a following period.

Tokenization and stemming were also performed before generating the feature set. We relied on stemming which was one of the most common techniques in information retrieval to eliminate the basic variations of some words. In linguistic morphology, stemming is defined as the process of reducing inflected (or sometimes derived) words to their stem, base, or root form. Generally, a written word form and the efficiency of content-based spam filter can be significantly improved by stemming [82]. This process effectively eliminates the influence of different forms of the same word while word statistics are produced.

5.2.2 Feature extraction and selection

As previously mentioned, we have three types of features in our feature set: word-, topic-, and user-based features. We adopt *tf-idf* scheme for word-based features to calculate the weight for each word in each comment. We compute the ATI and AS of each user for user-based features. Topic-based features are slightly more complicated because topic labels are generated via χ^2 test.

However, the cardinality of the word set is 211,278 after performing the χ^2 test. The set has many misspelled words that appear less than 10 times among the one million comments. If every word is selected as a feature, the feature vector of each comment tends to be very sparse, which will affect both the efficiency and the detection performance. Thus, we set a minimum frequency of 50 for each word, and a word is removed from the feature set if its frequency is below the minimum. Once we set the minimum frequency limit, the size of the vocabulary decreased to 7921.

If the number of latent topics to be learned via L-LDA is set to k , then the top k discriminative words generated by χ^2 test will be selected as the topic labels. After applying L-LDA to analyze the latent topic structure in each comment, the normalized topic frequency in each comment can be obtained. By including the user-based features ATI and AS, we obtain a total of $k + 7923$ features in our feature set.

5.2.3 Classification

We utilized a toolbox from Google named *sofia-ml* to implement incremental SVM, perceptron, ROMMA and logistic regression algorithms [83]. The following four different feature combinations were created to improve the performance for each feature type: (1) only word feature (W); (2) word + user features (WU); (3) word + topic features (WT); and (4) word + topic + user features (WTU). We attempted to measure and see how performance of our methodology would be improved by adding certain features. We assumed that the detection results were not only affected by the combinations of feature sets, but also the number of latent topics to be applied. Thus, the number of latent topics was set as $k = 6, 10, 20, 50, 80, 100$. Finally, except the W and WU groups, 2×6 experiment groups were established.

5.2.4 Evaluation metrics

The standard metrics used in evaluating the performance of the experimental models involved precision (PRE), accuracy (ACC), recall (REC), F_1 -measure (F1), and receiver operating characteristic curve (ROC) together with the area under the ROC curve (AUC). We introduced the statistical significance test to compare the performance of classifiers to avoid stochastic fluctuation problems in evaluation. Paired t-tests were adopted to test the statistical differences among the performance scores achieved via the cross validation procedure.

6 Experimental results and analysis

6.1 Overall performance

We performed the training and testing of 14 groups of classifiers using incremental learning methods, and measured the overall performance of each group (see Tables 5, 6, 7, and 8). The five basic performance metrics mentioned above were applied in each case. For WT and WTU groups, when topic quantity $k = 10$ or 20 , detection performance seemed to be better. In terms of accuracy, precision, and recall, better performance was achieved by a group containing topic-based features rather than using only word- or user-based features.

Table 5 Performance (%) of the W group

Classifier	Metrics				
	ACC	PRE	REC	F1	AUC
SVM	87.45	65.42	81.33	72.51	86.64
LogitReg	83.95	57.97	76.82	66.08	87.82
ROMMA	74.85	43.59	80.17	56.47	85.15
Perceptron	76.29	45.13	76.50	56.77	83.71

Bold values indicate the best value in the table

Table 6 Performance (%) of the WU group

Classifier	Metrics				
	ACC	PRE	REC	F1	AUC
SVM	88.05	66.23	84.20	74.14	87.19
LogitReg	87.58	65.66	81.65	72.79	91.10
ROMMA	84.47	58.78	79.26	67.50	89.18
Perceptron	78.09	47.70	79.30	59.57	85.79

Bold values indicate the best value in the table

Table 7 Performance (%) of the WT group

Topic quantity (k)	Classifier	Metrics				
		ACC	PRE	REC	F1	AUC
6	SVM	90.54	77.37	75.63	76.49	86.80
	LogitReg	80.77	52.22	64.97	57.90	81.50
	ROMMA	84.08	59.69	67.04	63.15	85.00
	Perceptron	76.66	45.44	73.27	56.09	81.62
10	SVM	90.64	78.36	74.60	76.43	86.57
	LogitReg	87.67	69.43	70.44	69.93	87.48
	ROMMA	72.35	41.11	82.85	54.95	84.73
	Perceptron	76.24	45.00	75.29	56.33	83.68
20	SVM	90.69	78.59	74.57	76.53	86.66
	LogitReg	82.34	55.25	69.52	61.57	83.50
	ROMMA	56.66	30.87	91.16	46.12	85.82
	Perceptron	71.98	39.81	73.68	51.69	78.45
50	SVM	89.83	73.69	77.83	75.70	87.05
	LogitReg	71.81	40.48	81.93	54.19	84.10
	ROMMA	80.21	50.95	74.30	60.45	84.33
	Perceptron	65.78	34.94	79.13	48.48	76.12
80	SVM	88.00	67.36	79.63	72.98	86.46
	LogitReg	71.10	39.85	82.42	53.72	84.34
	ROMMA	77.02	46.12	76.71	57.61	83.85
	Perceptron	60.43	32.17	85.20	46.71	77.86
100	SVM	86.65	63.58	80.56	71.07	85.91
	LogitReg	72.48	41.14	81.75	54.74	84.70
	ROMMA	61.63	33.31	88.40	48.39	83.77
	Perceptron	64.11	34.19	82.56	48.36	74.97

Bold values indicate the best value in the table

Groups with topic-based features often achieved a higher precision as topic-based features could reveal a comment's latent semantics that were utilized by a classifier to distinguish between spam and ham. Under certain situations, user-based features are external and sometimes unreliable because spammers could control and alter

Table 8 Performance of (%) the WTU group

Topic quantity (k)	Classifier	Metrics				
		ACC	PRE	REC	F1	AUC
6	SVM	90.71	76.44	78.60	77.50	87.34
	LogitReg	87.41	67.09	74.82	70.74	88.62
	ROMMA	85.20	61.29	74.05	67.07	87.25
	Perceptron	85.30	59.97	83.41	69.77	88.96
10	SVM	91.17	77.93	78.94	78.43	87.75
	LogitReg	89.87	75.19	74.97	75.08	89.98
	ROMMA	79.37	49.61	85.64	62.83	89.68
	Perceptron	87.84	71.87	66.12	68.88	85.98
20	SVM	91.03	77.52	78.80	78.15	87.67
	LogitReg	88.51	69.29	78.20	73.48	89.38
	ROMMA	86.38	63.88	76.06	69.44	88.25
	Perceptron	74.42	43.47	84.20	57.34	83.52
50	SVM	90.41	74.49	80.43	77.35	87.73
	LogitReg	81.42	52.71	84.70	64.98	90.00
	ROMMA	78.86	48.86	83.86	61.75	88.34
	Perceptron	72.09	41.18	86.69	55.84	85.49
80	SVM	89.82	72.31	80.95	76.39	87.42
	LogitReg	81.80	53.30	85.23	65.59	90.28
	ROMMA	77.79	47.36	81.79	59.99	86.53
	Perceptron	59.16	32.01	89.59	47.17	81.25
100	SVM	89.11	69.71	82.24	75.46	87.37
	LogitReg	82.59	54.62	85.29	66.59	90.50
	ROMMA	77.76	47.26	79.95	59.40	85.82
	Perceptron	64.52	35.06	87.24	50.02	76.66

Bold values indicate the best value in the table

their behavior deliberately (e.g., writing spam in a different way to reduce AS). However, although the way to generate a spam may change, the inherent connections between spam topics and spam comments remains unchanged. Topic model could easily identify a new spam topic by taking advantage of the inherent semantics of spam contents and the previously trained topic-word distribution.

We found that SVM classifiers outperformed other classifiers in most cases, whereas the performance achieved by logistic regression classifiers was the closest to that of SVM when compared with perceptron and ROMMA (see Fig. 6). SVM performed the best in the WTU group with topic quantity $k = 10$ that achieved an accuracy of 91.17 % and a F1-measure of 78.43 %. By comparing the performance of all SVM classifiers in the WTU group, it revealed that topic quantity $k = 10$ was the optimal value (see Fig. 7). Detection performance tended to improves when topic quantity increased from the minimum, and slightly decreased when the topic quantity reached the maximum. By fixing $k = 10$, the ROC curves are depicted in

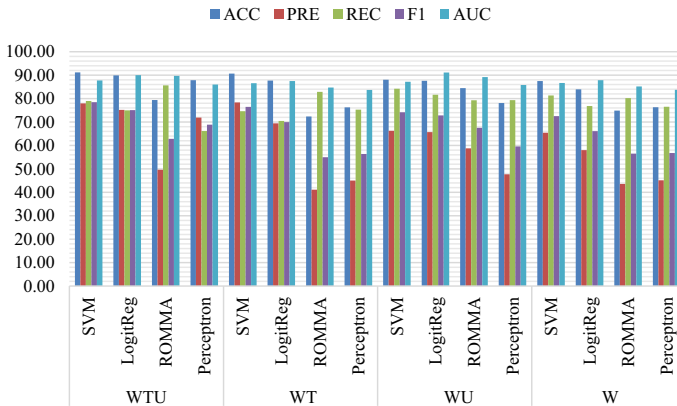


Fig. 6 A comparison of classification performance with topic quantity $k = 10$

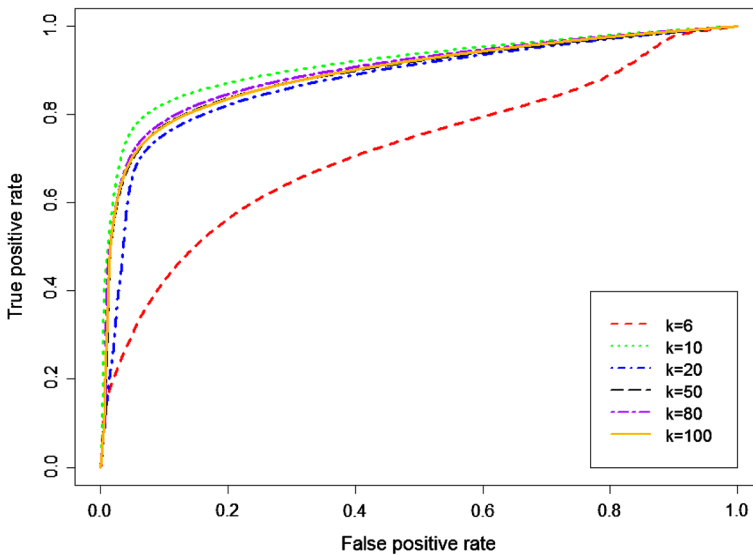


Fig. 7 ROC curves for SVM classifiers with the WTU feature set

Fig. 8. It reveals that the WTU group outperforms other groups. Moreover, classifiers with and without topic-based features lead to very different performance given that both the W and the WU groups, and the WT and the WTU groups are similar to the neighboring group, but dissimilar to each other with remote groups.

6.2 Hypothesis testing

For the hypothesis testing part, we used stratified sampling method [84] to divide our test dataset into 20 subsets and perform a classification of these subsets. The stratified sampling can ensure that spam and ham ratio is consistent in these subsets

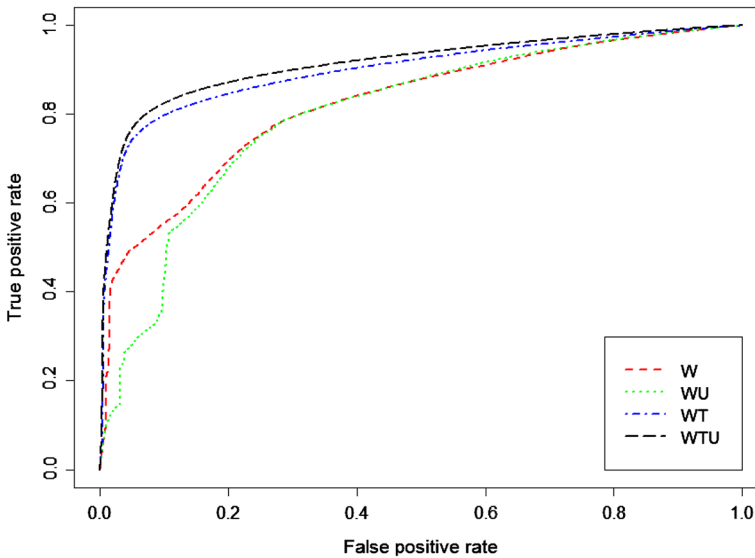


Fig. 8 ROC curves for SVM classifiers with topic quantity $k = 10$

Table 9 P-values for paired t-tests against ACC, PRE, and REC

Feature set	W	WU	WT	WTU
H1a, H2a, H3a—Accuracy				
WU	<0.001			
WT	<0.001	<0.001		
WTU	<0.001	<0.001		<0.10
H1b, H2b, H3b—Precision				
WU	<0.001			
WT	<0.001	<0.001		
WTU	<0.001	<0.001		<0.001 ^a
H1c, H2c, H3c—Recall				
WU	<0.001			
WT	<0.001 ^a	<0.001 ^a		
WTU	<0.001	<0.001 ^a		<0.001

^a Opposite to hypothesis

with those in the training set. Then paired t-tests were performed on the results from the 20 subsets to evaluate the performance of classifiers with various combinations of feature sets and hence to verify the hypotheses set out in section four (see Table 9).

In terms of accuracy, the results of all of the pairs are significant. Hence, H1a and H3a are supported at $p < 0.001$, whereas H2a is marginally supported because $ACC(WTU) > ACC(WT)$ is only significant at $p < 0.10$. By comparing the results of the two groups related to H3a, we conclude that after introducing topic-based features, the effect of user-based features is weakened in terms of accuracy. Topic-

based features are more effective than user-based features in improving detection performance as measured by accuracy, as stated in H3a.

In the following table about precision, only the result of $PRE(WTU) > PRE(WT)$ is non-significant, whereas those of the others are all significant. Thus, H2b is partly supported, whereas H1b and H3b are supported at $p < 0.001$. Although $PRE(WTU) > PRE(WT)$ is non-significant, it is oppositely significant, which means $PRE(WTU) < PRE(WT)$ is significant at $p < 0.001$. This could be regarded a piece of evidence that user-based features are sometimes unreliable because spammers may deliberate alter their behavior, which might further undermine the contributions of topic-based features when they these two feature sets are combined.

For the recall-related table, $REC(WU) > REC(W)$ and $REC(WTU) > REC(WT)$, which support H2c at $p < 0.001$ that user-based features will improve the recall of classifiers. H1c and H3c are rejected but the opposite are supported. The results signal that user-based features can improve recall to some extent, at the cost of precision. It seems that classifiers with user-based features are more inclined to judge a comment as spam, which would decrease the precision and increase the recall. However, misclassification is possible to occur partly because spam labels in the dataset are based on the content-level and provided by users on YouTube. Another reason might be that spammers use their accounts to generate spam as well as legitimate comments. This could lead to some legitimate comments to be judged as spam as a result of the suspicious users and their characteristics. Besides, the reason why H1c and H3c are supported in another direction is that the unbalanced proportion between spam and ham might lead to the occurrence of large numbers of previously unseen legitimate words. These words are more likely to be involved in a ham topic assignment by the topic model because it is very possible that they are surrounded by legitimate words. The topic model has the ability to identify a new example of spam based on the previous topic-word distribution and the connection between spam topics.

In sum, the groups containing topic-based features outperform those only with word or user-based features in terms of accuracy and precision. The groups with user-based features have the tendency to detect spam and achieve a higher recall than the other groups. Our experimental results offer considerable insights to social marketers who want to identify a robust strategy to filter spam on SMSs. If the social marketers want their SMSs to be absolutely clean from spam and prefer a more efficient detection process, they should utilize more user-based features to filter spam. In contrast, if the marketers want to reduce the chance of misclassifying legitimate users as spammers, they should utilize more topic-based features to filter spam.

6.3 Feature analysis

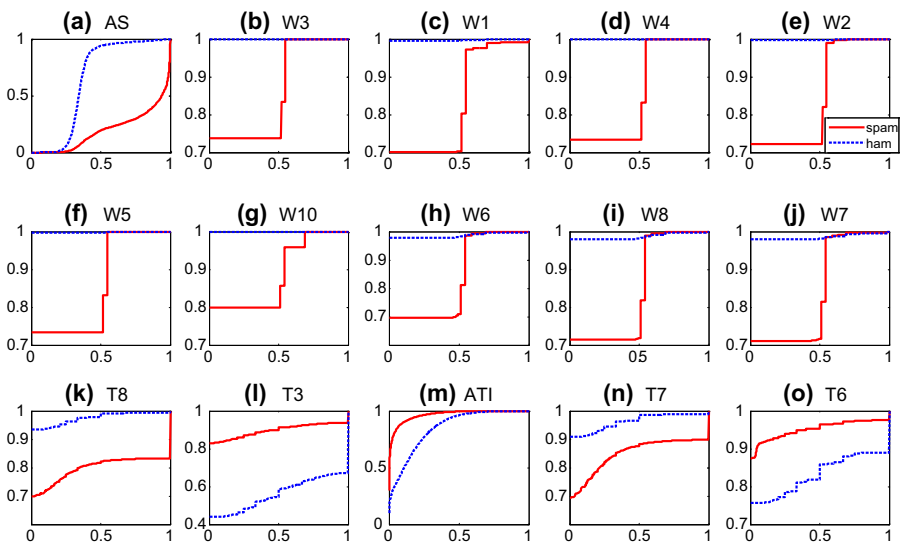
To examine the discriminative power of the proposed feature sets, Fisher score [85] is applied. Fisher score is a measure of the degree of how an independent feature could distinguish two classes although it ignores the mutual information between the features. The features with high Fisher scores are AS as well as the discriminative words with high χ^2 scores (the feature description is based on the

Table 10 F-scores of discriminative features

Rank	Description	F-Score
1	AS	1.464436
2	W3: cheapest	0.283030
3	W1: com	0.276846
4	W4: above	0.273212
5	W2: site	0.271944
6	W5: three	0.260267
7	W10: shop	0.187820
8	W6: world	0.182876
9	W8: most	0.166200
10	W7: best	0.152408
15	T8	0.132072
22	T3	0.103344
23	ATI	0.100820
33	T7	0.059718
44	T6	0.023775

χ^2 score rank). In addition, ATI together with other topic-based features are likewise in the top tier (see Table 10). The extreme high value of the feature AS suggests that it is very common for spammers to keep posting near duplicated comments on YouTube.

To further analyze these features, we plot the discriminative power of these features based on the cumulative distribution function (CDF) for each selected feature (see Fig. 9). We found that all the selected word-based features (W1–W8,

**Fig. 9** CDFs of features with high Fisher scores

W10) are spam-related, which means that the larger their values are, the more likely the corresponding comments are spam. This is understandable because these are typical spam-related words according to the spam examples highlighted before (see Table 3). Besides, AS, T7, and T8 are likewise spam-related, whereas ATI, T3, and T6 are ham-related. For topic-based features, we list words with relatively high probabilities under the selected topics to verify our claim. It shows that the words under T3 are quite neutral and those under T6 seem to be commonly used in chatting or gossiping which also tend to be irrelevant with respect to spam (see Table 11). For the words under T7 and T8, they are likely to appear in some adversarial messages of online shops; the names of some celebrities, such as Jordan and Kobe, are captured by T7 and T8 because many promoters leverage the celebrities to attract people's attention. By performing a POS tag analysis on those high probability words under the selected topics shown in Table 11, we found T6 has around 16 % interjections, which could support that T6 is about online chatting or gossiping. Besides, we also observed that T7 and T8 contained around 20 %

Table 11 Highly probable words of discriminative topics

Topic	T3	T6	T7	T8
Topic words with high probabilities	user	wa	NUMBER	video
	like	fuck	jersey	shop
	just	lol	best	interest
	know	shit	thing	boyfriend
	people	xd	more	cheapest
	right	so	cool	now
	more	look	most	best
	game	awesome	before	most
	comment	guy	world	world
	stupid	kid	super	three
	way	hate	star	site
	hate	funny	iphone	com
	wrong	oh	connector	cheap
	play	omg	cheapest	kobe
	idiot	dislike	site	lucky
	stop	sound	jordan	wade
	kid	wow	com	web
	god	amazing	shop	nice
	troll	suck	cheap	apple
	white	bitch	favourite	smart
	person	epic	justin	spam
	american	cute	watch	phone
	retard	hell	make	youtube
	Gay	wtf	love	hottest
	ignore	pretty	bieber	ipad

more common nouns (NN), 12 % more adjective comparatives (JJR) and superlatives (JJS), and 6 % more proper nouns (NNP) on average when compared with T3 and T6. These findings are largely in line with Ott et al.'s work [86] which asserts that deceptive writing is usually done in an exaggerated language. Furthermore, we did get other findings by running the POS tag analysis; however, they were not relevant for our study.

According to our analysis, T7 and T8 are spam-related; such a finding is consistent with the aforementioned explanations (see Fig. 9). Finally, by directly inspecting the words with high probabilities for a topic, it is straightforward to infer the spam class of that topic. It suggests that the topics extracted via L-LDA are highly interpretable.

7 Conclusions

With the ubiquitous of the social web, there has been an explosive growth of user-contributed comments. Meanwhile, there has also been a growing concern about the wide spread of social spam embedded in user-contributed comments. Given the big volume of user-contributed comments on SMSs, there is a pressing need to develop novel methodologies and techniques to tackle social spam.

Previous studies use various features (e.g., user-, text, graph-, and social network-related attributes) and classification algorithms (e.g., Naïve Bayesian and Bayesian Network) to design frameworks for detecting social spam on SMSs (e.g., Facebook, Twitter, Sina Weibo, Myspace, YouTube, and Flickr). However, to the best of our knowledge, previous studies have not exploited both probabilistic topic modeling and incremental learning for detecting social spam on SMSs. Thus, the main contributions of our research are the design and evaluation of a novel social spam methodology which is underpinned by the L-LDA model and incremental learning. More specifically, we exploit word-, topic-, and user-based features to better represent social spam and leverage incremental classifiers, such as SVM, logistic regression, perceptron, ROMMA, to enhance spam detection performance. Based on several millions of user comments posted to YouTube, our experimental results show that the proposed methodology can achieve an average accuracy of 91.17 % and an average F1-measure of 78.43 %, respectively. According to our paired t-tests, topic-based features improve the overall accuracy and precision. However, they may hurt the recall of spam detection. In contrast, user-based features enhance the recall of spam detection, but it may hurt precision.

The managerial implication of our research is that e-commerce managers can apply the proposed methodology to alleviate the interferences of social spam, and hence to discover more accurate business intelligence from the big volume of user-contributed comments posted to SMSs. Moreover, our methodology can improve the daily administration of SMSs, and enhance the hygiene and efficiency of these sites in the era of Big Data. For social media marketers, the proposed methodology is a powerful tool to ensure that their marketing campaigns are free from the interferences of social spam. Finally, the proposed methodology helps users identify relevant information from SMSs and improve their loyalty to these sites.

For our future research, we will refine the existing topic-based features because some latent topics mined using L-LDA may be similar to each other. Further clustering of the latent topics mined via L-LDA should improve the quality of the topic-based features. We will also try to take the videos into consideration to come up with more effective features to detect spammers with small comment accounts so as to make up for the limitation in our work. Besides, additional features, such as spam diffusion patterns in a social network, will be explored to enhance social spam detection performance. Moreover, crowd sourcing will be explored to improve the quality of our evaluation dataset because we found some occasional mistakes of the “Spam Hint” judgments provided by YouTube. Furthermore, deep learning methods will be examined to learn other high-level features apart from topic-based features.

Acknowledgments This work was supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Projects: CityU 11502115), and the Shenzhen Municipal Science and Technology R&D Funding - Basic Research Program (Project No. JCYJ201404191156 14350).

References

1. van Marle, D. (2011) IP telephony shifts from unified communications to social media. In *Proceedings of the 50th FITCE Congress, 2011* (pp. 1–4). Piscataway: IEEE
2. Gupta, R., Gupta, H., & Mohania, M. (2012). Cloud computing and big data analytics: What is new from databases perspective? In *Big Data Analytics* (pp. 42–61). Berlin: Springer.
3. Chandramouli, R. (2011). Emerging social media threats: Technology and policy perspectives. In *Proceedings of the 2nd Worldwide Cybersecurity Summit (WCS), London* (pp. 1–4). Piscataway: IEEE
4. Zhou, L., Wu, J., & Zhang, D. (2014). Discourse cues to deception in the case of multiple receivers. *Information & Management*, 51(6), 726–737. doi:10.1016/j.im.2014.05.011.
5. Wu, G., Greene, D., Smyth, B., & Cunningham, P. A. (2010) Distortion as a validation criterion in the identification of suspicious reviews. In *Proceedings of the 1st Workshop on Social Media Analytics, New York* (pp. 10–13, SOMA '10): Association of Computing Machinery (ACM). doi:10.1145/1964858.1964860.
6. Yoo, K.-H., & Gretzel, U. (2009). Comparison of deceptive and truthful travel reviews. In W. Höpken, U. Gretzel, & R. Law (Eds.), *Information and Communication Technologies in Tourism 2009* (pp. 37–47). Vienna: Springer.
7. Theft, fraud cost retailers \$8 million a day: study. (2007), *The Ottawa Citizen*, pp. E.3-E3.
8. Wang, D., Irani, D., & Pu, C. (2014). SPADE: A social-spam analytics and detection framework. *Social Network Analysis and Mining*, 4(1), 1–18. doi:10.1007/s13278-014-0189-1.
9. Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of ACM*, 50(10), 94–100.
10. Lin, Y.-R., Sundaram, H., Chi, Y., Tatemura, J. I., & Tseng, B. L. (2008). Detecting splogs via temporal dynamics using self-similarity analysis. *ACM Transactions on the Web*, 2(1), 4. doi:10.1145/1326561.1326565.
11. Boyd, D., & Heer, J. (2006) Profiles as conversation: Networked identity performance on friendster. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences, Koloa, Hawaii* (Vol. 3, pp. 59c-59c). Piscataway: IEEE Computer Society
12. Brown, G., Howe, T., Ihbe, M., Prakash, A., & Borders, K. (2008). Social networks and context-aware spam. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work, New York* (pp. 403–412, CSCW '08): Association of Computing Machinery (ACM). doi:10.1145/1460563.1460628.
13. Zinman, A., & Donath, J. (2007). *Is Britney Spears spam?* In Paper presented at the 4th Conference on Email and Anti-Spam, Mountain View, California.

14. Harold, & Nguyen (2014). *2013 State of Social Media Spam Report* (2013 Research Report ed., pp. 21). Burlingame, California: Nexgate.
15. Grier, C., Thomas, K., Paxson, V., & Zhang, M. (2010) @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, New York* (Vol. Chicago, Illinois, pp. 27–37): Association of Computing Machinery (ACM). doi:<http://doi.acm.org/10.1145/1866307.1866311>.
16. Zhang, D., Yan, Z., Jiang, H., & Kim, T. (2014). A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites. *Information & Management*, *51*(7), 845–853.
17. Ensing, & David (2013). Money talks and listens: Characteristics of rating and review site users. *Maritz Research's White Papers*, *4*
18. IC3 (2008). 2008 Internet Crime Report (p. 25): Internet Crime Complaint Center.
19. Reviews, reputation, and revenue: The case of Yelp.com (2011). Harvard Business School, Boston College. http://www.hbs.edu/faculty/Publication%20Files/12-016_0464f20e-35b2-492e-a328-fb14a325f718.pdf.
20. Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings on the Conference on Empirical Methods in Natural Language Processing, Singapore* (pp. 248–256): Association for Computational Linguistics
21. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.
22. Markines, B., Cattuto, C., & Menczer, F. (2009). Social spam detection. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web, New York* (pp. 41–48, AIRWeb '09): Association of Computing Machinery (ACM). doi:<http://doi.acm.org/10.1145/1531914.1531924>.
23. Lee, K., Caverlee, J., & Webb, S. (2010). Uncovering social spammers: social honeypots + machine learning. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York* (pp. 435–442): Association of Computing Machinery (ACM). doi:[10.1145/1835449.1835522](http://doi.acm.org/10.1145/1835449.1835522).
24. Jin, X., Lin, C., Luo, J., & Han, J. (2011). A data mining-based spam detection system for social media networks. *Proceedings of the VLDB Endowment*, *4*(12), 1458–1461.
25. Lin, L., & Kun, J. (2012). Detecting spam in Chinese microblogs: A study on Sina Weibo. In *Proceedings of the 8th International Conference on Computational Intelligence and Security, Guangzhou, Guangdong Province* (pp. 578–581): China Printing Solutions. doi:[10.1109/cis.2012.135](http://doi.org/10.1109/cis.2012.135).
26. Dae-Ha, P., Eun-Ae, C., & Byung-Won, O. (2013). Social spam discovery using bayesian network classifiers based on feature extractions. In *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, Australia, July 2013* (pp. 1808–1811) Piscataway: IEEE
27. Po-Ching, L., & Po-Min, H. (2013). A study of effective features for detecting long-surviving Twitter spam accounts. In *Proceedings of the 15th International Conference on Advanced Communication Technology, PyeongChang, South Korea, Jan 2013* (pp. 841–846). Piscataway: IEEE
28. Sureka, A. (2011). *Mining user comment activity for detecting forum spammers in Youtube*. Paper presented at the 1st International Workshop on Usage Analysis and the Web of Data, Hyderabad, India
29. Brody, S., & Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Los Angeles, California* (pp. 804–812): Association for Computational Linguistics
30. Liu, B., Liu, L., Tsykin, A., Goodall, G. J., Green, J. E., Zhu, M., et al. (2010). Identifying functional miRNA–mRNA regulatory modules with correspondence latent dirichlet allocation. *Bioinformatics*, *26*(24), 3105–3111.
31. Wang, C., Blei, D., & Li, F.-F. (2009). Simultaneous image classification and annotation. In *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL* (pp. 1903–1910). Piscataway: IEEE
32. Bíró, I., Szabó, J., & Benczúr, A. A. (2008). Latent dirichlet allocation in web spam filtering. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web, Beijing* (pp. 29–32). New York: Association of Computing Machinery (ACM)

33. Cui, K., Zhou, B., Jia, Y., & Liang, Z. (2010). LDA-based model for online topic evolution mining. *Computer Science*, 37(11), 156–193.
34. Sizov, S. (2010). Geofolk: Latent spatial semantics in web 2.0 social media. In *Proceedings of the third ACM international conference on Web search and data mining* (pp. 281–290). New York: ACM
35. Geng, X., & Smith-Miles, K. (2009). Incremental learning. In S. Li & A. Jain (Eds.), *Encyclopedia of biometrics* (pp. 731–735). Berlin: Springer.
36. Mitchell, T. M. (1997). *Machine learning*. Boston: McGraw-Hill.
37. Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18(2), 203–226.
38. Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2), 139–172.
39. Utgoff, P. E. (1988). Id5: An incremental id3. In *Proceedings of 5th International Workshop on Machine Learning, Ann Arbor, Michigan* (pp. 107–120). Burlington, MA: Morgan Kaufmann
40. Martinez, C., & Tony, G.-C. (1995). ILA: Combining inductive learning with prior knowledge and reasoning. 17
41. Tsai, C. H., Lin, C. Y., & Lin, C. J. (2014). Incremental and decremental training for linear classification. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York* (pp. 343–352). New York: Association of Computing Machinery (ACM)
42. Mairal, J. (2015). Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2), 829–855.
43. Salton, G., & McGill, M. J. (1986). *Introduction to modern information retrieval*. New York: McGraw-Hill.
44. Aphinyanaphongs, Y., Fu, L. D., Li, Z., Peskin, E. R., Efstathiadis, E., Aliferis, C. F., et al. (2014). A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization. *Journal of the Association for Information Science and Technology*, 65(10), 1964–1987.
45. Sood, S. O., Churchill, E. F., & Antin, J. (2012). Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology*, 63(2), 270–285.
46. Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. *Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, USA, 1997* (pp. 143–151). San Francisco: Morgan Kaufmann Publishers Inc.
47. Soucy, P., & Mineau, G. W. (2005) Beyond TFIDF weighting for text categorization in the vector space model. In *Proceedings of the International Joint Conferences on Artificial Intelligence, Edinburgh, Scotland* (Vol. 5, pp. 1130–1135): IJCAI Organization
48. Singhal, A., Choi, J., Hindle, D., Lewis, D. D., & Pereira, F. (1999). AT&T at TREC-7. In *Proceedings of the 7th Text Retrieval Conference, Gaithersburg, MD* (pp. 239–252): National Institute of Standards and Technology (NIST)
49. Alexandrov, M., Gelbukh, A. F., & Lozovoi, G. (2001) Chi square classifier for document categorization. In *Proceedings of the 2nd International Conference on Computational Linguistics and Intelligent Text Processing, Mexico City* (Vol. 2004, pp. 457–459). Berlin: Springer
50. Dunham, M. H., & Ming, D. (2003). *Introductory and advanced topics*. Upper Saddle River, NJ: Prentice Hall/Pearson Education.
51. Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3(7–8), 1289–1305.
52. Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M. S., & Al-Rajeh, A. (2008). Automatic arabic text classification. In *Paper presented at the 9th International Conference on the Statistical Analysis of Textual Data, Lyon*.
53. Mesleh, A Md. (2007). Chi square feature extraction based svms arabic text categorization system. *Journal of Computer Science*, 3(6), 430–435.
54. Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1–47.
55. Halliday, M. A., & Matthiessen, C. M. (2004). *An introduction to functional grammar*. New York: Routledge.
56. Fairclough, N. (2003). *Analysing discourse: Textual analysis for social research*. London: Routledge.
57. Abbasi, A., & Chen, H. (2008). CyberGate: a design framework and system for text analysis of computer-mediated communication. *MIS Quarterly*, 32(4), 811–837.
58. Casella, G., & George, E. I. (1992). Explaining the Gibbs sampler. *The American Statistician*, 46(3), 167–174.

59. Duan, Z., Gopalan, K., & Yuan, X. (2011). An empirical study of behavioral characteristics of spammers: Findings and implications. *Computer Communications*, 34(14), 1764–1776. doi:10.1016/j.comcom.2011.03.015.
60. Gao, H., Chen, Y., Lee, K., Palsetia, D., & Choudhary, A. N. (2012). Towards online spam filtering in social networks. In *NDSS*
61. Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., & Zhao, B. Y. (2010). Detecting and characterizing social spam campaigns. In *Paper presented at the Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, Melbourne*.
62. Chen, C., Wu, K., Srinivasan, V., & Zhang, X. (2013). Battling the internet water army: detection of hidden paid posters. In *Paper presented at the Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Niagara*.
63. Mukherjee, A., Liu, B., & Glance, N. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web, 2012* (pp. 191–200). New York: ACM
64. Song, J., Lee, S., & Kim, J. (2011). Spam filtering in twitter using sender-receiver relationship. In R. Sommer, D. Balzarotti, & G. Maier (Eds.), *Recent advances in intrusion detection* (Vol. 6961, pp. 301–317), Lecture Notes in Computer Science Berlin, Heidelberg: Springer.
65. Wang, A. H. (2010). Don't follow me: Spam detection in Twitter. In *Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT) 2010* (pp. 1–10)
66. Myers, E. W. (1986). An O(ND) difference algorithm and its variations. *Algorithmica*, 1(1–4), 251–266.
67. Ukkonen, E. (1985). Algorithms for approximate string matching. *Information and Control*, 64(1), 100–118.
68. Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3), 291–316.
69. Manaskasemsak, B., Jiarpakdee, J., & Rungsawang, A. (2014). Adaptive Learning Ant Colony Optimization for Web Spam Detection. In *Computational Science and Its Applications—ICCSA 2014* (Vol. 8584, pp. 642–653, Lecture Notes in Computer Science). Berlin: Springer.
70. Congfu, X., Baojun, S., Yunbiao, C., & Weike, P. (2014). An adaptive fusion algorithm for spam detection. *IEEE Intelligent Systems*, 29(4), 2–8.
71. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.
72. Li, Y., & Long, P. (2002). The relaxed online maximum margin algorithm. *Machine Learning*, 46(1–3), 361–387.
73. Zhang, T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21th International Conference on Machine Learning, Banff, Alberta, Canada, 2004* (p. 116). New York: Association of Computing Machinery (ACM). doi:10.1145/1015330.1015332.
74. Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1), 3–30.
75. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(3), 551–585.
76. Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Paper presented at the Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, CA*.
77. O'Callaghan, D., Harrigan, M., Carthy, J., & Cunningham, P. A. (2012) Identifying discriminating network motifs in YouTube spam. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media, Dublin* (pp. 521–529): Association for the Advancement of Artificial Intelligence
78. O'Callaghan, D., Harrigan, M., Carthy, J., & Cunningham, P. A. (2012) Network analysis of recurring YouTube spam campaigns. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media, Dublin* (pp. 531–534)
79. Helft, M. (2008). Search ads come to YouTube. <http://bits.blogs.nytimes.com/2008/10/13/search-ads-come-to-youtube/>.
80. YouTube (2013). Youtube: Statistics.
81. Sivaselvan, B., & Gopalan, N. P. (2009). *Data mining: Techniques and trends*. New Delhi: Prentice-Hall.

82. Ahmed, S., & Mithun, F. (2004). Word stemming to enhance spam filtering. In *Paper presented at the 1st Conference on Email and Anti-Spam, Mountain View, CA*.
83. Sculley, D. (2010) Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington DC* (pp. 979–988). New York: Association of Computing Machinery (ACM)
84. Neyman, J. (1934). On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4), 558–625.
85. Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. New York: Wiley.
86. Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). 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, Stroudsburg* (Vol. 1, pp. 309–319, HLT'11): Association for Computational Linguistics