Effective Opinion Spam Detection: A Study on Review Metadata Versus Content

Ajay Rastogi[†], Monica Mehrotra, Syed Shafat Ali

Department of Computer Science, Jamia Millia Islamia, New Delhi, India

Citation: Rastogi, Ajay, Monica Mehrotra, and Syed Shafat Ali. "Effective opinion spam detection: A study on review metadata versus content." *Journal of Data and Information Science*, vol. 5, no. 2, 2020, pp. 76–110. <https://doi.org/10.2478/jdis-2020-0013>

Received: Jan. 21, 2020

Revised: Apr. 12, 2020

Accepted: Apr. 23, 2020

Abstract

Purpose: This paper aims to analyze the effectiveness of two major types of features—metadata-based (behavioral) and content-based (textual)—in opinion spam detection.

Design/methodology/approach: Based on spam-detection perspectives, our approach works in three settings: review-centric (spam detection), reviewer-centric (spammer detection) and product-centric (spam-targeted product detection). Besides this, to negate any kind of classifier-bias, we employ four classifiers to get a better and unbiased reflection of the obtained results. In addition, we have proposed a new set of features which are compared against some well-known related works. The experiments performed on two real-world datasets show the effectiveness of different features in opinion spam detection.

Findings: Our findings indicate that behavioral features are more efficient as well as effective than the textual to detect opinion spam across all three settings. In addition, models trained on hybrid features produce results quite similar to those trained on behavioral features than on the textual, further establishing the superiority of behavioral features as dominating indicators of opinion spam. The features used in this work provide improvement over existing features utilized in other related works. Furthermore, the computation time analysis for feature extraction phase shows the better cost efficiency of behavioral features over the textual.

Research limitations: The analyses conducted in this paper are solely limited to two well-known datasets, viz., YelpZip and YelpNYC of Yelp.com.

Practical implications: The results obtained in this paper can be used to improve the detection of opinion spam, wherein the researchers may work on improving and developing feature engineering and selection techniques focused more on metadata information.

Originality/value: To the best of our knowledge, this study is the first of its kind which considers three perspectives (review, reviewer and product-centric) and four classifiers to analyze the effectiveness of opinion spam detection using two major types of features. This study also introduces some novel features, which help to improve the performance of opinion spam detection methods.

Keywords Opinion spam; Behavioral features; Textual features; Review spammers; Spam-targeted products

1 Introduction

With the overwhelming advancements in technological landscape, particularly World Wide Web, the e-commerce industry has experienced a colossal growth in terms of online business marketing. This in turn has served as an essential practice to push the sales of business products among the common masses. One of the main factors for such a successful trend, besides advertisement, are the public opinions. These opinions can be expressed in the form of product reviews describing purchase experiences, which tend to drive how consumers think about a product or service. For example, on Amazon, whenever a consumer aims at purchasing an item, the corresponding reviews are generally taken into consideration. Therefore, these reviews have the potential to influence the decision making of a consumer as to whether to make a purchase or not. Consequently, reviews can be considered as a basic unit for a business and an eye-opener for consumers.

However, given this significance of product reviews, if used with malicious intentions, could lead to dire consequences. For example, a business organization may mislead its consumers by inducing forged reviews for monetary gains. Therefore, determining the authenticity of online reviews emerges as a challenging task. People write malicious reviews, traditionally known as opinion spam[®], to promote or demote a target product or service with the aim to mislead consumers for financial gains. Jindal and Liu (2007a) were the first who highlighted opinion spam in Amazon reviews and provided novel solution to detect them. Literature on opinion spam have suggested two spamming perspectives: review spam and review spammer. In review-centric approaches, the focus is on detecting review-spam (Jindal & Liu, 2007a; 2007b; 2008; Li et al., 2011; Li et al., 2013; Ott et al., 2011), while reviewer-centric approaches center around identifying review spammers (Fei et al., 2013; Lim et al., 2010; Mukherjee et al., 2013a; Wang et al., 2012).

One of the important aspects of opinion spam detection is to analyze the effectiveness of features, i.e. which features are the most prominent indicators of opinion spam. Some studies have focused on textual characteristics such as n-gram models (Mukherjee et al., 2013b; Ott et al., 2011), part-of-speech (POS) tagging information (Feng et al., 2012; Ott et al., 2011), positive or negative inclinations (Li et al., 2011; Mukherjee et al., 2013c; Rayana & Akoglu, 2015) etc., while others

[®] Opinion spam and fake reviews have been used interchangeably in this paper. Moreover, we use opinion spam in a generic way to refer to review spam, review spammers and spam-targeted products.

Research Paper

have utilized behavioral signatures such as rating deviation (Lim et al., 2010; Savage et al., 2015), extremity of ratings (Mukherjee et al., 2013a; Rayana & Akoglu, 2015), activity time window (Mukherjee et al., 2012; Mukherjee et al., 2013a), early time frame (Mukherjee et al., 2013a; Rayana & Akoglu, 2015) etc. Being the seminal work of its kind, Rastogi and Mehrotra (2018) analyzed the effectiveness of behavioral and textual features individually in opinion spam detection under reviewer and review-centric settings. This study was confined to train SVM classifier on YelpZip dataset. In the present study, besides reviewer and review-centric, one more novel perspective i.e. product-centric setting has been introduced where we aspire to identify spam-targeted products. In addition to this, three more classifiers, logistic regression (LR), multi-layer perceptron (MLP), and naïve Bayes (NB) have been employed to measure the impact of different classification mechanisms on opinion spam detection. To get a broader understanding of the effectiveness of features, our experiments are conducted on two real-world datasets YelpZip and YelpNYC. In addition, we have also proposed some novel features in this work which, together with some previously used features, are compared with the features used in three well-known related works: i) Mukherjee et al. (2013a), ii) Mukherjee et al. (2013c), and iii) Rayana and Akoglu (2015). Our motivation for this comparison is to analyse the effectiveness of proposed sets of features over existing literature in detecting opinion spam, spammers and spam-targeted products. The following are the main contributions of this paper:

- This study is the first of its kind which considers two common perspectives i.e. review, and reviewer-centric, one more novel perspective i.e. product-centric, and four classifiers to analyze the effectiveness of opinion spam detection using two major types of features.
- We compare the resulting hybrid features (combination of behavioral and textual) against behavioral and textual features independently, and find that behavioral features are as good as hybrid, and textual remain the worst in detecting opinion spam.
- Some of the new behavioral and textual features have been introduced in this work and the resulting set of features (either adopted from previous works or devised in this work) are compared with some previous related works. The results show that the existing features when supplemented with our novel features can outperform the features used in several related works.

Our results indicate that, in comparison to the textual, behavioral features contain the prominent spam signatures resulting in a better identification of opinion spam. Not only are they better able to classify spam, but they portray their superiority in identifying the spammer as well as the products targeted by spammers, thereby,

illustrating their effectiveness across all three spamming perspectives. Furthermore, this significant characteristic of behavioral features encompasses both the datasets and is found to be classifier independent. The statistical significance analysis of the results is further validated by performing two-tailed Z-test. Interestingly, when we mix behavioral and textual features, the performance of the resulting hybrid features is negligibly marginal when contrasted against the behavioral. Notably, we encounter certain cases, where behavioral features either perform better or as good as hybrid features. This further goes on to strengthen the case of choosing behavioral features as the prime candidates for opinion spam detection. In addition, the features used in this work provide improvement over existing features used in some well-known related works in the literature. We also perform computation time analysis for the feature extraction phase. Our analysis shows that the time involved in engineering behavioral features is significantly lesser in comparison to the time involved in the textual. This added advantage of the resulting cost efficiency of extracting behavioral features again points towards choosing behavioral features for opinion spam detection.

The remainder of the paper is organized as follows: Section 2 discusses the relevant literature to this work. Section 3 provides our research methodology framework with detailed description of the individual modules in its subsections. Dataset description, pre-processing, feature extraction and experimental setup are thoroughly described in those subsections. Then, Section 4 summarizes and discusses the results, and finally, Section 5 concludes the paper with possible future research directions.

2 Literature review

Detection of fake reviews has been a topic of interest among researchers from diverse disciplines, especially since the seminal work of Jindal and Liu (2007b). To date, several studies have been conducted to mitigate the same. Different studies approached opinion spam detection problem from different perspectives viz., review spam (Jindal & Liu, 2008), review spammer (Mukherjee et al., 2013a) and group spamming (Mukherjee et al., 2012) as stated by Rastogi and Mehrotra (2017). In the present literature, two wider categories of features have been utilized in the studies so far, by either working on text information or by utilizing meta-data associated with reviews. Treading along the same idea, we divide the literature into two categories: content-based features and metadata-based features.

2.1 Content-based features

To assess the credibility of online reviews, a vast variety of content-specific features have been suggested in the literature. As fraudsters have very little or no

experience with the product or service, they tend to use different words during deceptive writing, creating a distributional difference between the languages of deceptive and truthful opinions (Li et al., 2013; Li et al., 2014). Unigram and bigram word features have shown good accuracies in detecting spam reviews (Li et al., 2013; Ott et al., 2011). However, n-gram features experience difficulty in cross-domain deception detection i.e. when these features are tested on multiple domains (like product, hotel, restaurant etc.), then their performance degrades (Li et al., 2014). Furthermore, Ott et al. (2011) checked for POS distributions in deceptive and truthful opinions and found a plausible distinction between the two. Their finding indicates that deceptive reviews appear to be more imaginative i.e. consist of more verbs and adverbs, while truthful reviews show an informative writing style i.e. consist of more nouns and adjectives. Instead of simple POS tags distribution in deceptive and truthful reviews, Feng et al. (2012) mined deep syntactic patterns captured from context free grammar (CFG) parse trees. For the same, they used POS tag information in the sentences of fake and honest reviews and generate certain discriminative production rules by using CFG parse trees. Along with all these features, by gaining clues about deceptive writing from psychology and computational linguistics, some works utilized Linguistic Inquiry and Word Count (LIWC) features to spot fake opinions (Li et al., 2014; Ott et al., 2011).

2.2 Metadata-based features

Metadata linked with reviews, viz., review date, time and rating, provides useful information about the behavior of individual reviewers. In order to capture the abnormal behavior of review spammers, much of the previous works have utilized these metadata features and build more efficient review spamming models to predict review spam. Based on the metadata type, behavioral features have further been grouped into two groups: rating-related features and temporal features as discussed below.

2.2.1 Rating-related features

These features are based on the ratings given to some target products by a reviewer or by a group of reviewers. As suggested by Savage et al. (2015), spammers often deviate from the majority of reviewers in terms of product ratings. For the same, they took the difference between the average rating, as an estimate of majority rating, and the rating given by a reviewer on the corresponding product. The same feature, rating deviation, has been used by most of the previous works (Fei et al., 2013; Jindal & Liu, 2007a; Lim et al., 2010; Mukherjee et al., 2013a; Mukherjee et al., 2013c; Rayana & Akoglu, 2015), making it a strong indicator of review spam. In addition, Lim et al. (2010) used rating deviation along with one more factor

indicating how early the review was written after product launch. The authors combined these two, rating deviation and earliness of a review, into one feature by taking weighted average over all the reviews of a reviewer, to yield a spam score for each reviewer. Moreover, literature suggests that review spammers are more prone to give extreme ratings (1 or 5 on a 5-star scale) as their sole aim is to either raise or lower the average rating of a targeted product (Mukherjee et al., 2013a; Rayana & Akoglu, 2015).

2.2.2 Temporal features

Temporal features describe the bursty nature of review spammers by concentrating on dimension date provided with meta-data of reviews. Review spamming often indicates some bursty patterns occurring in small time windows on target products (Fei et al., 2013), meaning that spam attacks often contain reviews in bursts. For the same, Fei et al. (2013) introduced a burst detection mechanism to identify the suspicious candidates for review bursts. Then they modeled them using markov random field and used loopy belief propagation method to ultimately spot opinion spammers in candidate bursts. A related but different approach is employed by Xie et al. (2012) to detect singleton review (SR) spam. The authors hypothesized that if there is a sudden increase in the number of singleton reviews with either very low or high ratings in a small-time window, then it might indicate an SR spam. Furthermore, Heydari et al. (2016) also used time as a crucial factor in detecting review spam by examining the reviews captured in suspicious bursty intervals. They used rating deviation, content similarity and activeness of accounts as three key factors inspected across suspicious intervals to spot review spam. Another important temporal feature highlights the activity behavior of reviewers showing for how long time a reviewer remains active (Mukherjee et al., 2013b). After a close and careful inspection, Mukherjee et al. (2013b) showed that spammers usually don't remain active for long time, whereas normal reviewers have comparatively larger activity windows.

3 Research methodology

Our proposed research methodology framework for opinion spam detection is shown in Fig. 1. Firstly, we take the review data as input and pre-process it to get the suitable instances of reviews, reviewers and products. Secondly, in feature extraction module, two categories of features, i.e. behavioral features (metadata-based), derived from meta-data information associated with reviews and textual features (content-based), derived from the review text itself, have been engineered with respect to three different settings, viz., reviewer, review and product-centric.

Research Paper

Then under these settings, by using both types of features, classification models are independently trained to predict the reviewer as spammer or not-spammer, review as spam or not-spam and product as targeted or not-targeted. It is important to note that, in Fig. 1, we have shown only two classes, i.e. spam and not-spam for the sake of generality. However, these classes will be different for different settings. At last, in each setting, we evaluate the performance of our classification models with the aim to find out efficient features responsible to detect opinion spam. In the upcoming subsections, we shall shed light on different modules building up the proposed framework.

Figure 1. Framework of proposed research methodology.

3.1 Dataset description

The experiments are conducted on two restaurant review datasets, viz., YelpZip[®] and YelpNYC[®] (near benchmarks, previously used by Fontanarava et al. (2017), Rayana & Akoglu (2015), and Rayana & Akoglu (2016)), which were collected from yelp.com by Rayana and Akoglu (2015). These datasets contain review IDs along with the reviewer and product ID information, ratings, timestamps (dates) and the labels for reviews (spam or not-spam). Dataset statistics for both YelpZip and YelpNYC are given in Table 1.

Table 1. Dataset statistics (for YelpZip and YelpNYC).

Dataset	# Reviews (spam %)	# Reviewers (spammer %)	# Products (restaurants)
YelpZip	608,598 (13.22%)	260,277 (23.91%)	5,044
YelpNYC	359,052 (10.27%)	160,225 (17.79%)	923

3.1.1 Dataset pre-processing

Before starting our feature extraction phase, we perform preprocessing on both the collected datasets. The reason for the same is the presence of large number of singleton reviews for both reviewers and products. As explained by Mukherjee et al. (2013a), products and reviewers with fewer reviews provide few behavioral information. Therefore, we filter out all the reviews of the products and reviewers having less than three reviews. After the initial filtering, if it results in less than three reviews for any product or reviewer, we again filter out those reviews. This process is repeated until all the reviewers and products are left with at least three reviews. After preprocessing, we left with preprocessed datasets whose statistics have been given in Table 2.

Table 2. Dataset statistics after preprocessing (for YelpZip and YelpNYC).

Dataset	# Reviews (spam%)	# Reviewers (spammer%)	# Products (restaurants)
YelpZip (Preprocessed)	356,766 (4.66%)	49,841 (9.21%)	3,975
YelpNYC (Preprocessed)	90,906 (7.58%)	15,351 (10.67%)	873

Moreover, in Rayana and Akoglu (2015), authors have considered the Yelp filtered and recommended reviews as “spam” and “not-spam”, respectively. The same was verified by Luca and Zervas (2016). Since our approach works in three settings as discussed in Section 1, it is important to assign the labels to both reviewers and products as well. Similar to Rayana and Akoglu (2015), labels to reviewers are

[®] <http://odds.cs.stonybrook.edu/yelpzip-dataset/>

[®] <http://odds.cs.stonybrook.edu/yelpNYC-dataset/>

assigned based on the reviews written by them. If a reviewer has filtered reviews, we assign a label as “spammer”, while reviewers having no filtered reviews are labelled as “not-spammer”. In the similar fashion, we consider a product as “targeted” if it has filtered reviews, otherwise “not-targeted”.

3.2 Feature extraction

One of the important steps in dealing with the opinion spam detection problem is to extract the relevant set of features. Different studies have utilized different features, viz., content-specific review information (Li et al., 2013; Ott et al., 2011), behavioral information (Lim et al., 2010; Mukherjee et al., 2013a), network structural information (Wang et al., 2012), product characteristics (Sun et al., 2016) etc. to identify fake reviews. Researchers have exploited these features, sometimes separately and sometimes by taking any combination of these, as improvement over existing techniques. As suggested by the existing literature (Rastogi & Mehrotra, 2017), review spam and review spammers are two major detection targets to tackle the problem of opinion spam. Review spam are detected in review-centric perspective, whereas spammers are detected by considering reviewer-centric perspective. This paper takes both the approaches into account and refer to them as review-centric and reviewer-centric setting, respectively. In addition to this, one novel perspective i.e. product-centric setting, has been introduced by taking the products, potentially targeted for spamming, into consideration. The key motivation to consider this product-centric view is as follows: Since a review is basically viewed from three main perspectives, viz., reviewer, review and product, this enables us to imagine a review in the form of a tripartite network as shown in Fig. 2. Therefore, while analyzing review and reviewer-centric approaches, it is equally important to examine the products targeted by the spammers.

This Section explains the features employed in this study for each of the above mentioned three settings. Two common types of features, viz., behavioral (metadata-based) and textual (content-based) have been extracted in each setting. To briefly summarize our feature extraction part, Table 3 lists all the features (both behavioral and textual) with brief description, for each of the three settings. The features for each category are either adopted from the literature (referenced in Table 3) or devised in this study. Our main motivation to devise new textual and behavioral features centers around the following: The main aim of this work is to analyze the impact of different features related to metadata and review text in detecting opinion spam, spammers and spam-targeted products. So, it is both imperative and intuitive to collect as much behavioral or textual information as possible. For example: motivated by a feature “*review rank*” used in Rayana and Akoglu (2015), we devise

Figure 2. Tripartite network of reviewers (U), reviews (R) and products (P).

two new binary features “*top ranked review (TRR)*” and “*bottom ranked review (BRR)*”. The rationale behind construction of these two features is to capture the notion of spammers trying to write the reviews as early as possible to get maximum impact and to influence the sentiment on the product. So top ranked reviews (*TRR*) are highly suspicious to spam as compared to those fall in bottom ranked (*BRR*). Moreover, most of the review-centric features can be easily utilized in reviewer-centric setting, by taking the average of all the feature values of reviews of a reviewer (Table 3). The idea behind constructing the reviewer-centric features in such a way is that the features which have been proven to discriminate between spam and not-spam reviews (refer to Section 2), can in turn discriminate spammer from a non-spammer. Most importantly, it is interesting to know that whether the features introduced in this work can provide some improvements over existing features used in state-of-the-art. With the same intention, we measure the impact of the features used (adopted and devised) in this study against previous well-known works in Section 4.4.

We briefly present the features which are devised in the current study in following subsections and for brevity, we omit the details of remaining features which are adopted from previous works. As most of the features from reviewer-centric setting are similar to those from product-centric setting, therefore, the features from these two settings are described together in common subsection 3.2.2.

Research Paper

Table 3. Brief description of behavioral and textual features employed under reviewer-centric, review-centric and product-centric settings.

Setting	Feature type	Feature	Description
Reviewer-centric and Product-centric	Behavioral	ARD	Average rating deviation (Fei et al., 2013)
		WRD	Weighted rating deviation (Rayana and Akoglu, 2015)
		MRD*	Maximum rating deviation
		BST	Burstiness (Mukherjee et al., 2013a)
		ERR*	Early review ratio
		MNR	Maximum number of reviews (Mukherjee et al., 2013a)
		RPR	Ratio of positive reviews (Rayana and Akoglu, 2015)
		RNR	Ratio of negative reviews (Rayana and Akoglu, 2015)
		FRR	First review ratio (Mukherjee et al., 2013a)
		EXRR*	Extreme rating ratio
		TRRR*	Top ranked reviews ratio
		BRRR*	Bottom ranked reviews ratio
	Textual	MCS	Maximum content similarity (Mukherjee et al., 2013a)
		ACS	Average content similarity (Lim et al., 2010)
		AFPP*	Average first-person pronouns ratio
		ASPP*	Average second-person pronouns ratio
		AFTAPP*	Average first-and-third-person to all-person pronouns ratio
		ASAPP*	Average second-person to all-person pronouns ratio
		ASW*	Average subjective words ratio
		AOW*	Average objective words ratio
		AI _n W*	Average informative words ratio
		AI _m W*	Average imaginative words ratio
		ARL	Average review length (Rayana and Akoglu, 2015)
Review-centric	Behavioral	RD	Rating deviation (Mukherjee et al., 2013a)
		ERD*	Early rating deviation
		ETF	Early time frame (Mukherjee et al., 2013a)
		EXT	Extreme rating (Mukherjee et al., 2013a)
		TRR*	Top ranked review
		BRR*	Bottom ranked review
		RR	Review rank (Rayana and Akoglu, 2015)
	Textual	RL	Review length (Mukherjee et al., 2013c)
		RPW	Ratio of positive words (Li et al., 2011)
		RNW	Ratio of negative words (Li et al., 2011)
		RFPP	Ratio of first-person pronouns (Li et al., 2011)
		RSPP	Ratio of second-person pronouns (Li et al., 2011)
		RFTAPP*	Ratio of first-and-third-person to all-person pronouns
		RSAPP*	Ratio of second-person to all-person pronouns
		RSW	Ratio of subjective words (Li et al., 2011)
		ROW	Ratio of objective words (Li et al., 2011)
		RI _n W	Ratio of informative words (Ott et al., 2011)
		RI _m W	Ratio of imaginative words (Ott et al., 2011)

Note: * indicates those features introduced in this study; rest are adopted from previous studies.

3.2.1 Review-centric setting

The features related to the review-centric setting are discussed in this Section. As mentioned above, the features are categorized into two types, viz., behavioral features which are derived from the meta-data information related to a review like its rating, time etc., and textual features, i.e. those constructed from review text.

A. Behavioral Features

To capture the behavioral evidences of spamming, meta-data information of a review with respect to all the other reviews on the product has been utilized. Review related behavioral features devised in this study are as follows:

Early Rating Deviation (ERD): To get the maximum impact for their ratings, spammers try to post their reviews as early as possible. By doing so, they aim to attain enough attention from subsequent reviewers to manipulate their views on the product (Lim et al., 2010). *ERD* takes this factor into account and modifies rating deviation by multiplying it by a weight representing earliness of a review.

$$f_{ERD}(r) = \frac{|r_p - \bar{r}_p|}{4} * w_{rp} \quad (1)$$

where r_p is the rating of r^{th} review on product p , $\bar{r}_p$ is the average rating of other reviewers on product p and $w_{rp} = \frac{1}{(t_{rp})^\alpha}$ represents weight of r^{th} review on product p . Here, t_{rp} shows the time of review r on product p defined as $t_{rp} = r$ for r^{th} review on product p and parameter $\alpha = 1.5$ is decay parameter for w_{rp} .

Top Ranked Review (TRR): Reviews getting top ranks (according to dates) on a product are likely to get more user attention and are more suspicious to be spam (Jindal & Liu, 2008). After a product is made available online, people's perception about the product largely depends on the first few reviews. Spammers exploit this idea and try to project the product in the wrong direction.

$$f_{TRR}(r) = \begin{cases} 1, & f_{RR}(r) \leq \gamma_1 \\ 0, & otherwise \end{cases} \quad (2)$$

$f_{RR}(r)$ indicates the rank of review r and γ_1 is a threshold deciding whether a review is top ranked or not. This threshold γ_1 has been decided empirically as $\gamma_1 = (20\% \text{ of } \lceil |N_p| \rceil)$, where N_p is the set of all reviews posted on different dates for product p .

Bottom Ranked Review (BRR): Contrary to *TRR* as defined above, this feature decides whether a review is bottom ranked or not. Intuitively, reviews coming after enough time of product launch have lesser probability to be spam.

$$f_{BRR}(r) = \begin{cases} 1, & f_{RR}(r) \geq \gamma_2 \\ 0, & otherwise \end{cases} \quad (3)$$

Again, threshold γ_2 is decided empirically and taken as $\gamma_2 = |N_p| - (20\% \text{ of } \lceil |N_p| \rceil)$ indicating review falls in the bottom ranked if rank exceeds this threshold.

B. Textual Features

This set of features includes all those features which are directly related to the text part of a review rather than meta-data as is the case of behavioral features.

Studies have shown that spammers deviate from the normal users in terms of language they use in their reviews. From very simple to highly advanced like linguistic, psycholinguistic, sentimental etc., the textual features encompass a broad spectrum of review text characteristics.

First-and-Third-Person vs. Second Person Pronouns Ratio: To share a true experience about something, first and third-person pronouns are more likely to be used as compared to second-person pronouns. For example, a genuine reviewer may describe a hotel like “we both enjoyed a lot there”, “she went there first time”, “their rooms were pretty nice”, etc. It indicates that third-person pronouns come closer to first-person in case of a genuine review. Therefore, we compute the *ratio of first-and-third-person to all-person pronouns* and *ratio of second-person to all-person pronouns*, viz., *RFTAPP* and *RSAPP*, respectively.

3.2.2 Reviewer-centric and product-centric setting

As mentioned in Section 3.2, features related to both the remaining settings i.e. reviewer-centric and product-centric, are defined and briefly explained in this Section. In reviewer-centric setting, our goal is to detect the persons responsible for opinion spam i.e. opinion spammers, whereas in product-centric view, we aim to identify targeted products. To get the features related to a reviewer, we go beyond a single review and accumulate all the reviews written by him. In the similar fashion, product related features are computed by combining all the reviews posted on a product. Again, similar to review-centric setting, the features are categorized into two types i.e. behavioral and textual for both these settings.

A. Behavioral Features

Features related to a reviewer behavior indicate the behavioral facets like rating pattern, reviewing pattern with respect to time etc., and are based on meta-data associated with reviews written by a reviewer. To capture the behavioral evidences of a reviewer, we first go through all the reviews written by him/her and then combine them by some aggregate operation to come up with a final value for a feature. Here, it is noteworthy that almost all the behavioral features related to a reviewer are also applicable to product-centric setting, where we consider all the reviews posted on a product rather than those written by a reviewer. Only one feature, i.e., *first review ratio (FRR)*, doesn't apply for a product since a product always has one and only one review as a first review.

Maximum Rating Deviation (MRD): To capture the extreme case of rating deviations, *MRD* takes the maximum of rating deviations of all the reviews of a reviewer or a product and is defined in a similar way to *ARD* as:

$$f_{MRD}(u) = \max_{r \in R_u} \frac{|r_{rp} - \bar{r}_p|}{4} \quad (4)$$

where r_p is the rating of r^{th} review on product p , $\bar{r}_p$ is the average rating of other reviewers on product p and R_u is the set of all reviews of a reviewer or a product.

Early Review Ratio (ERR): As mentioned in Section 3.2.1, reviews coming in the early phase of a product, known as early reviews, have bigger influence on people's perceptions about the product. *ERR* computes the ratio of early reviews to the total number of reviews of a reviewer or a product.

$$f_{ERR}(u) = \frac{|\{r : r \in R_u \ \& \ f_{ETF}(r) = 1\}|}{|R_u|} \quad (5)$$

where $f_{ETF}(r)$ denotes the early time frame of a review r described in Mukherjee et al. (2013a) and R_u is the set of all reviews of a reviewer or a product. A reviewer (or product) getting *ERR* value close to 1 may be considered as spammer (or spam-targeted).

Extreme Rating Ratio (EXRR): *EXRR* computes the ratio of reviews having extreme ratings to the total number of reviews for a reviewer or a product.

$$f_{EXRR}(u) = \frac{|\{r : r \in R_u \ \& \ f_{EXT}(r) = 1\}|}{|R_u|} \quad (6)$$

Where $f_{EXT}(r)$ indicates extremity of rating given to review r described in Mukherjee et al. (2013a) and R_u is the set of all reviews of a reviewer or a product.

Top Ranked Reviews Ratio (TRRR): As already mentioned, posting reviews very early is an indication of spam. If a reviewer has most of his/her reviews as top ranked reviews i.e. most of the time, his/her reviews come right after the launch of the corresponding product, then his/her behavior may be considered suspicious. This feature *TRRR* computes the ratio of top ranked reviews to the total reviews of a reviewer or a product.

$$f_{TRRR}(u) = \frac{|\{r : r \in R_u \ \& \ f_{TRR}(r) = 1\}|}{|R_u|} \quad (7)$$

where $f_{TRR}(r)$ indicates whether a review is top ranked review or not, defined in Section 3.2.1 and R_u is the set of all reviews of a reviewer or a product.

Bottom Ranked Reviews Ratio (BRRR): Contrary to *TRRR* as defined above, this feature calculates the ratio of bottom ranked reviews to the total reviews of a reviewer or a product. The idea here is that a genuine reviewer is likely to rate a product or service after experiencing it and thus will take time for the same, as contrast to spammers who rate early to change the perception of consumers.

$$f_{BRRR}(u) = \frac{|\{r : r \in R_u \ \& \ f_{BRR}(r) = 1\}|}{|R_u|} \quad (8)$$

Research Paper

where $f_{BRR}(r)$ is defined in Section 3.2.1 and R_u is the set of all reviews of a reviewer or a product.

Here, it is important to note that for both the above features $TRRR$ and $BRRR$, computation of top ranked review (i.e. $f_{TRR}(r)$) and bottom ranked review (i.e. $f_{BRR}(r)$) is somewhat different for product-centric setting. We consider a review written for a product as top ranked (or bottom ranked) if it falls in the first (or last) 20% of reviews for the entire duration of the product (where duration is the absolute difference between the first and last review date of the corresponding product).

B. Textual Features

This type of features is engineered from review text. Here, we consider all the reviews of a reviewer or product and combine them by some aggregate operation to get the final value for a feature. Again, all the textual features for reviewer-centric setting are also applicable to product-centric.

Average First-Person vs. Second-Person Pronouns Ratio: These features are derived from the two features $RFPP$ and $RSPP$. For the same, we take the average of $RFPP$ and $RSPP$ individually over all the reviews for a reviewer or product, resulting in two features, viz., *average first-person pronouns ratio (AFPP)* and *average second-person pronouns ratio (ASPP)*, respectively.

Average First-and-Third-Person vs. Second-Person Pronouns Ratio: To compute these two, we take the average of $RFTAPP$ and $RSAPP$ (discussed in Section 3.2.1) individually over all the reviews of a reviewer or product. These features indicate the average distribution of first and third-person pronouns against second-person pronouns in total reviews of a reviewer or a product, resulting in two features, viz., *average first-and-third-person to all-person pronouns ratio (AFTAPP)* and *average second-person to all-person pronouns ratio (ASAPP)*, respectively.

Average Subjective vs. Objective Words Ratio: These features are again the individual average of two features RSW and ROW , over all the reviews of a reviewer or product, yielding to features, viz., *average subjective words ratio (ASW)* and *average objective words ratio (AOW)*, respectively.

Average Informative vs. Imaginative Words Ratio: Informative and imaginative words distribution among all the reviews of a reviewer or a product can be found by taking the average of two features, RI_nW and RI_mW , separately over all the reviews of that reviewer or product. This finally yields two features, viz., *average informative words ratio (AI_nW)* and *average imaginative words ratio (AI_mW)*, respectively.

3.3 Experimental setup

3.3.1 Classification

To analyze the effect of features for opinion spam detection, in this study, we use four supervised classifiers viz., support vector machine (SVM) (Cortes & Vapnik,

1995), logistic regression (LR) (Mitchell, 2005), multilayer perceptron (MLP) (Schurmann, 1996) and naïve Bayes (NB) (Rish, 2001). In SVM, a hyperplane is constructed in higher dimensions by using kernel trick for non-linear classification. This hyperplane is constructed in such a way which maximizes the margin between the nearest data-points of classes (two or more). Logistic regression classifier is a probabilistic model which uses logistic function (like sigmoid function) to discriminate the classes. To estimate the coefficients of the training model, maximum-likelihood estimation (MLE) is used by observing training data points. Multilayer perceptron is a neural network-based model having multiple layers, where each layer consists of neurons utilizing a nonlinear activation function, to classify non-linearly separable training examples. Naïve Bayes classifier is again a probabilistic model which uses a strong assumption of class independence between the features under consideration. Many of the studies in the literature have used naïve Bayes as a strong baseline to compare their works. The reason for using four classifiers is to examine the impact of different classification mechanisms on opinion spam detection.

3.3.2 Model training

In this paper, our main goal is to analyze the performance of meta-data-based features in comparison to content-based features in classifying opinion spam. We also aim at analyzing whether the features proposed in this work can provide any improvement over existing features used in some state-of-the-arts. For the same, as discussed in Section 3.2, two categories of features, viz., behavioral and textual under different settings, viz., reviewer, review and product-centric, have been engineered. This results in two feature sets[®] namely behavioral and textual for each of the three predefined settings. In this way, we get a behavioral feature set (set of all behavioral features) and a textual feature set (set of all textual features) for each reviewer, review and product-centric settings. Since we observe class imbalance in feature sets i.e. an overwhelmingly large proportion of the instances are skewed towards majority class, we first balance out the data such that each feature set (either behavioral or textual) ends up with equal number of class instances (equal spam and not-spam instances). This data imbalance problem is generally solved using random under sampling as used in Rastogi and Mehrotra (2018). Since random under sampling randomly eliminates majority class instances, this may result in missing out on such instances which can have drastic effect on model training.

To overcome this problem, we devise a novel approach to get a balanced view of an unbalanced feature set. The pseudo code for this approach is described in

[®] A feature set represents an $(n \times m)$ matrix with n number of features and m number of instances.

Research Paper

Algorithm 1 in Table 4. This algorithm takes an unbalanced feature set F as input and returns its balanced partitions as output. First, we randomly shuffle the instances in F and count the number of majority and minority class instances (Lines 1–6). Secondly, based on the number of minority class instances, we divide the majority class into equal size bins (Lines 7–8). For example, if there are p instances in majority class and q instances in minority class, then number of bins of majority class would be $k = \lceil q / p \rceil$. This results in k nearly equal size bins of majority class. Thereafter, each bin of majority class is combined with minority class, resulting in k balanced partitions of unbalanced feature set F (Lines 9–12). Now, as a result, in each balanced partition, we have approximately equal number of spam and not-spam class instances. This process of balancing is repeated for both the feature sets (textual and behavioral) to get corresponding balanced partitions under each setting (reviewer, review, product-centric).

Table 4. Algorithm for balancing the feature set.

Algorithm 1: Balancing Algorithm for Feature Set

Input: Unbalanced feature set F .

Output: k balanced partitions each containing nearly equal number of instances from both the classes.

1. Randomly shuffle the instances in F ;
2. Divide F into two sets S_1 and S_2 representing majority class and minority class, respectively;
3. $S_1 \leftarrow$ minority class instances;
4. $S_2 \leftarrow$ majority class instances;
5. $p \leftarrow \text{count}(S_1)$;
6. $q \leftarrow \text{count}(S_2)$;
7. $k \leftarrow \left\lceil \frac{q}{p} \right\rceil$;
8. $S_3 \leftarrow$ Divide S_2 into k nearly equal size bins;
9. **foreach** bin $z \in S_3$ **do**
10. Combine S_1 with z to get a balanced partition;
11. **end**
12. **return** k balanced partitions for unbalanced feature set F ;

The balanced partitions of each feature set thus produced serve as input to our model training process. For this study, as mentioned in Section 3.3.1, four classifiers viz., SVM, LR, MLP, and NB have been employed. The reason for using four classifiers is to examine the impact of different classification mechanisms on spam identification. For model training, we use balanced partitions of a feature set and perform 5-fold cross validation (CV) on each partition with stratified sampling. Different models are trained for each feature set (behavioral or textual) under each setting (Reviewer, review or product-centric) by using different classifiers independently. It is important to note that all the experiments have been conducted in default parameter setting using *scikit-learn machine learning library in python* (Pedregosa et al., 2011) for all the classifiers. Finally, we take the average of results

thus produced across each fold, which in turn are averaged over k balanced partitions for different classifiers independently. To get a clearer understanding of our experimental setup, the whole process from dataset to model training through feature extraction and balancing is shown in Fig. 3.

Figure 3. Overall process from dataset to model training through feature extraction and balancing for any of the reviewer, review and product-centric settings.

3.3.3 Model evaluation

To evaluate the performance of our classification models, following evaluation metrics have been utilized: average precision, recall and f1-score (at macro and micro levels). Opinion spam detection is a binary classification task as it has only two classes “spam” and “not-spam”. In binary classification, precision indicates how strong a classifier is in predicting spam, while recall indicates how strong a classifier is in detecting spam.

In more clearer terms, precision is the ratio of instances which are actually spam among all the instances which are predicted as spam, also known as “positive predictive value”, and mathematically it is defined as:

$$\text{Precision} = \frac{tp}{tp + fp} \quad (9)$$

Research Paper

On the other hand, recall is the ratio of instances which are predicted as spam among all the instances which are actually spam, also known as “true positive rate” or “sensitivity”, and mathematically it is defined as:

$$\text{Recall} = \frac{tp}{tp + fn} \quad (10)$$

where tp , fp and fn are the true positives, false positives and false negatives, respectively.

Average precision is the weighted average of precision and recall scores achieved at different thresholds and is defined as:

$$\text{Avg. Precision} = \sum_n (R_n - R_{n-1})P_n \quad (11)$$

where P_n and R_n are precision and recall at n^{th} threshold.

F1-score combines both precision and recall by taking their weighted harmonic mean and is defined as:

$$\text{F1-score} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (12)$$

We compute the f1-score by taking the average at macro and micro levels independently. At macro level, first f1-score is computed for each class independently and then average is taken, while at micro level, contribution of both the classes is taken together to calculate the final f1-score.

To further validate our results, besides these evaluation metrics, we perform graphical analysis by using Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves. ROC curve shows the relative trade-off between true positive rate and false positive rate at different discriminating thresholds and therefore provides a better understanding of a classifier’s performance (Fawcett, 2006). Similar to ROC, PR curve is used to show the relative trade-off between precision and recall at different discriminating thresholds. A single metric, area under the curve (AUC), is used to summarize the results of both the ROC and PR curves.

4 Results and discussion

This Section discusses the results obtained from model training[®]. For each setting, we evaluate the performance of our classification models in two ways: using evaluation metrics and graphical approach. For evaluation metrics, as discussed in Section 3.3.3, we use traditional precision, recall and f1-score measures, and for the

[®] The source codes and results of all the experiments performed in this paper, along with the datasets, are publicly available at GitHub. (<https://github.com/AjayRastogi/Code-and-Results>)

latter, two curve analyses namely ROC and PR curves have been performed. Furthermore, we combine behavioral and textual features to compare the effect of hybrid features against the former two independently. For the same, average precision, f1-score and ROC-AUC score have been utilized. Moreover, features used in this study (behavioral and textual both), together with existing features, are compared against some previous related works to the know the importance of proposed set of features in identifying spam, spammer and spam-targeted products. In addition to this, we also perform the computation time analysis of behavioral and textual features for feature extraction. The statistical significance analysis of the results contrasting behavioral and textual features is further validated by performing two-tailed Z-test.

4.1 Performance evaluation using evaluation metrics

As stated in Section 3.3.1, four classifiers, SVM, LR, MLP, and NB, have been used for model training across all the settings. Models trained on behavioral and textual features have been assessed using generic evaluation measures, viz., average precision, recall and f1-score (at macro and micro levels). Experiments are conducted on two datasets YelpZip and YelpNYC as described in Section 3.1. The results thus produced are shown in Table 5 and Table 6, respectively.

Table 5. Classifiers performance on YelpZip dataset using both behavioral and textual features over all three settings.

	SVM		LR		MLP		NB	
(a) Reviewer-centric								
	Behavioral	Textual	Behavioral	Textual	Behavioral	Textual	Behavioral	Textual
AP	0.7342	0.6682	0.7377	0.6717	0.7417	0.6783	0.6934	0.6558
Recall	0.5301	0.6063	0.5907	0.6537	0.6395	0.6497	0.5902	0.3140
F1 (Macro)	0.6700	0.6260	0.6841	0.6340	0.6943	0.6343	0.6681	0.5491
F1 (Micro)	0.6767	0.6268	0.6868	0.6343	0.6952	0.6353	0.6701	0.5808
(b) Review-centric								
AP	0.6873	0.6461	0.6826	0.6232	0.6994	0.6581	0.6401	0.6478
Recall	0.7821	0.4348	0.7413	0.3775	0.7121	0.5947	0.6907	0.3612
F1 (Macro)	0.6394	0.5888	0.6544	0.5655	0.6637	0.6180	0.6233	0.5663
F1 (Micro)	0.6471	0.5998	0.6574	0.5830	0.6650	0.6187	0.6259	0.5876
(c) Product-centric								
AP	0.8692	0.8440	0.8717	0.8421	0.8741	0.8499	0.8691	0.8432
Recall	0.8218	0.7795	0.8101	0.7774	0.7926	0.7731	0.9004	1.0000
F1 (Macro)	0.7488	0.7279	0.7526	0.7293	0.7569	0.7347	0.6758	0.3537
F1 (Micro)	0.7541	0.7321	0.7569	0.7333	0.7598	0.7384	0.7020	0.5472

Table 5 shows the performance of different classifiers trained on YelpZip dataset for behavioral and textual features, where (a), (b) and (c) represent the results under

Research Paper

Table 6. Classifiers performance on YelpNYC dataset using both behavioral and textual features over all three settings.

SVM		LR		MLP		NB		
(a) Reviewer-centric								
	Behavioral	Textual	Behavioral	Textual	Behavioral	Textual	Behavioral	Textual
AP	0.8001	0.6866	0.8022	0.6999	0.8136	0.7029	0.7594	0.6860
Recall	0.6638	0.5945	0.6859	0.7011	0.7045	0.6878	0.5503	0.3103
F1 (Macro)	0.7140	0.6250	0.7148	0.6495	0.7224	0.6472	0.6749	0.5462
F1 (Micro)	0.7143	0.6259	0.7149	0.6517	0.7226	0.6491	0.6791	0.5759
(b) Review-centric								
AP	0.7313	0.6566	0.7311	0.6444	0.7461	0.6708	0.6811	0.6640
Recall	0.7189	0.5270	0.7475	0.5073	0.7548	0.6089	0.3462	0.3611
F1 (Macro)	0.6573	0.6052	0.6728	0.5897	0.6822	0.6173	0.5621	0.5655
F1 (Micro)	0.6598	0.6069	0.6758	0.5917	0.6852	0.6179	0.5849	0.5851
(c) Product-centric								
AP	0.8839	0.8345	0.8876	0.8367	0.8896	0.8345	0.8909	0.8357
Recall	0.8474	0.3396	0.8282	0.6844	0.8186	0.6770	0.8419	0.7006
F1 (Macro)	0.7865	0.6177	0.8016	0.7369	0.7974	0.7048	0.8066	0.7332
F1 (Micro)	0.7880	0.6601	0.8024	0.7385	0.7983	0.7109	0.8073	0.7344

reviewer, review and product-centric settings, respectively. As we can see from Table 5 (a), under reviewer-centric setting, behavioral features trump textual features across average precision and f1-score (at both macro and micro levels) for all the classifiers. It indicates that models trained on behavioral features perform better spammer prediction as compared to those trained on textual features. While recall might be slightly better for textual features, this advantage is nullified by a better f1-score for the behavioral. This indicates that overall behavioral features are better able to classify spammers. By observing Table 5 (b), we found that across all the metrics and for all the classifiers, behavioral features outperform the textual in opinion spam classification (review-centric setting). Interestingly, the contrast seen in recall under reviewer-centric setting is not seen here, with recall scores being far better for behavioral features. This effectiveness of behavioral features can also be seen by observing Table 5 (c) depicting results for product-centric setting with only one exception in recall for naive Bayes classifier. On one hand, for textual features, naïve Bayes produced a very high recall value i.e. 1.0, while on the other hand, it's f1-score is comparatively far lower than those of behavioral features (See Table 5 (c)), which indicates a highly undesired bias towards positive class. From the above discussion, it is quite clear that behavioral features are generally better spam indicators, be it review spam, review spammers or targeted products, irrespective of the classifier used.

We further substantiate the superiority of behavioral features by observing the results on another dataset YelpNYC. The results obtained on YelpNYC are shown in Table 6 in the similar fashion as in YelpZip in Table 5. Remarkably in case of YelpNYC, not only are the results similar to YelpZip, but are even better indicators of behavioral features being superior to textual features.

From reviewer-centric perspective as shown in Table 6 (a), again it can be seen that behavioral features surpass textual ones across all the evaluation metrics with only one exception in recall. In case of LR classifier, recall is a bit higher for textual features, but still higher average precision and f1-score illustrate the dominance of behavioral features over the textual. Here, it is important to note that average precision and f1-score are not only better for behavioral features, but they beat textual features by a big margin (in case of YelpZip, this margin was small) which again shows how effective the meta-data information is in detecting spammers. Similar type of results can be seen in case of review-centric setting as shown in Table 6 (b) except for naïve bayes classifier. NB performs equally respectable for both the features, but its performance is not as good as the performances of other classifiers. The underperformance of NB might be reasoned with a strong assumption of its class independence, which is very rare in general. To sum up, behavioral features outperform textual ones across all the metrics irrespective of the classifier used, portraying the primacy of behavioral features as being better review-spam indicators as well. This significance of behavioral features is even more pronounced on product-centric setting as shown in Table 6 (c).

From the above discussion, we found a consistent win for behavioral features across all the classifiers on both the datasets (Table 5 and 6). Moreover, across all the evaluation metrics, our results are in strong favor of behavioral features with some exceptions in recall (in reviewer-centric setting only). However, these exceptions are not significant which is validated by significant f1-scores throughout all the classifiers.

4.2 Performance evaluation using graphical approach

In addition to evaluation metrics as discussed in Section 4.1, we perform ROC and PR curve analyses to further enhance our understanding of behavioral and textual features in opinion spam detection. These examinations are conducted using three classifiers SVM, LR and MLP under all three settings on both the datasets. As was seen in the previous Section 4.1, in some cases, naïve Bayes classifier produced slightly unbalanced (or skewed) results because of its class independence assumption, therefore, we discard it for this analysis. Fig. 4 and Fig. 5 represent the results of ROC curve analysis, for YelpZip and YelpNYC datasets, respectively.

Figure 4. ROC curves of different classifiers trained using behavioral and textual features on YelpZip over (a-c) reviewer-centric, (d-f) reviewer-centric, and (g-i) product-centric setting.

Figure 5. ROC curves of different classifiers trained using behavioral and textual features on YelpNYC over (a-c) reviewer-centric, (d-f) review-centric, and (g-i) product-centric setting.

From reviewer-centric perspective, as shown in Fig. 4 (a-c), it is quite clear that across all three classifiers, behavioral features outperform the textual, covering more area under the ROC curve (AUC). This can also be verified with naked eye as the ROC curve for behavioral features consistently stays above the textual by consistently maintaining the higher true positive rate over false positive. Similar results are observed when we focus on review-centric setting as shown in Fig. 4 (d-f). Interestingly, here for LR model, the AUC score for behavioral features leads with a significant margin over the textual, as the “belly” formed between the two curves is considerably bigger than the ones formed under SVM and MLP classifiers. Moving towards the product-centric setting, again AUC scores are consistent and in favor of behavioral features across all the three classifiers as shown in Fig. 4 (g-i). From this analysis, it becomes clearer that behavioral features are better spam indicators as compared to the textual as discussed in Section 4.1 as well and this analysis further strengthens our results in favor of behavioral features.

Similar and even stronger results are observed when we analyze the ROC curves on YelpNYC as shown in Fig. 5. Fig. 5 (a-c), (d-f) and (g-i) collectively illustrate that across all the settings, ROC curves for behavioral features have more coverage of area than those for textual features (specified by the AUC scores shown in the corresponding figures). Most importantly, here in case of YelpNYC, AUC scores are significantly high for behavioral features and beat textual features by a considerable margin as well. For example, the difference between the AUC scores of behavioral and textual features (by taking best classifier) for reviewer-centric setting is 0.10 (Fig. 5 (a, c)), for review-centric setting, this difference leads to 0.11 (Fig. 5 (e)) and for product-centric setting, the difference is 0.07 (Fig. 5 (h, i)). This analysis clearly indicates a win for behavioral features across all the settings over YelpNYC dataset as well.

To further evaluate the performance of our classification models and as a supplement to our ROC curve analysis, one more graphical approach i.e. PR curve analysis has been adopted for both the datasets YelpZip and YelpNYC. Similar results were obtained by performing PR curve analysis, favoring behavioral features over the textual, in all the three settings on both the datasets, and hence we omitted those results for brevity and lack of pages. Results for PR curve analyses can be found here.[®]

From the results obtained in this study, we found that consistency of behavioral features over the textual holds strong, irrespective of the datasets, classifiers and settings. This indicates meta-data information contains more prominent signatures of spam as compared to those found in content-based information. This prominence

encompasses all three spamming contexts, be it review spam, review spammer or spam-targeted product. We understand the reason for this contrast to be intuitive. A spammer might typically be thought to have the “awareness” of the content he or she posts, such that his/her review “appears” to be genuine. For example, to mimic a genuine reviewer, a spammer would look to reduce the extent to which he or she would use imaginative words (Ott et al., 2011) or keep his/her review content more subjective (Li et al., 2011). However, he or she might not be “aware” of the behavioral clues he or she leaves while posting a review. For example, while posting a skewed rating relative to the average rating on a product (which he or she is bound to promote or demote a product), a spammer might not be quite “aware” of the rating deviation clue (Lim et al., 2010) he or she is inducing into his/her behavior. We believe a study could be undertaken where the “awareness” or “unawareness” of a spammer could be gaged to better discriminate him/her from the normal users.

4.3. Performance comparison with hybrid features

In this Section, we analyze the compound effect of behavioral and textual features on spam classification by combining the two. To this end, we trained our three classifiers SVM, LR and MLP on both the datasets under each setting (reviewer, review and product-centric). Using average precision, f1-score and ROC-AUC for evaluation, we compare the resulting hybrid features with behavioral and textual individually. Fig. 6 and Fig. 7 represent the obtained results for YelpZip and YelpNYC datasets, respectively.

On YelpZip, from reviewer-centric perspective, as shown in Fig. 6 (a-c), it can be observed that across all the metrics, hybrid features significantly perform better than the textual on all three classifiers. While hybrid features outperform the behavioral as well, notably, the outperformance is very marginal. When we shift our focus towards review-centric setting (Fig. 6 (d-f)), while textual features stay beaten by the hybrid and be that behavioral features trumping the hybrid as well. For example, in Fig. 6 (d), for SVM and MLP, higher average precision for behavioral features is apparent. Also, in case of SVM, behavioral features end up with higher ROC-AUC and f1-score than that of the hybrid. As for product-centric setting, the trend which was observed in reviewer-centric setting can be found here as well, as depicted in Fig. 6 (g-i). This analysis clearly indicates that hybrid features perform better than the behavioral and textual ones individually, but their outperformance is quite marginal in comparison to the behavioral.

In case of YelpNYC dataset (results are shown in Fig. 7), from reviewer-centric perspective, we observe from Fig. 7 (a-c), for f1-score and ROC-AUC, while comprehensively outperforming the textual, hybrid features produce slightly better results than the behavioral across all the classifiers, and for average precision,

Figure 6. Performance of behavioral, textual and hybrid features using different classifiers on YelpZip over (a-c) reviewer-centric, (d-f) review-centric, and (g-i) product-centric setting.

Figure 7. Performance of behavioral, textual and hybrid features using different classifiers on YelpNYC over (a-c) reviewer-centric, (d-f) review-centric, and (g-i) product-centric setting.

hybrid and behavioral features end up with similar results. As for review-centric setting (from Fig. 7 (d-f)), performance of hybrid features is again similar to the behavioral, with behavioral features exhibiting a slight edge over the hybrid in some cases. For example, for f1-score, from Fig. 7 (e), in LR and MLP, we observe behavioral features with better results than those of the hybrid. Here again, textual remain the worst performing features for spam classification for all three metrics across all the classifiers. Building upon this, we can observe a similar trend emerging in product-centric setting as shown in Fig. 7 (g-i). Interestingly, here for f1-score, behavioral features outclass both the hybrid as well as textual in case of SVM.

From the above discussion, we can draw an important conclusion. While the performance of hybrid features is generally better than the behavioral, the significance of outperformance is very marginal (negligible in some cases). In addition to this, there are cases where behavioral features perform better than the hybrid as well. These findings indicate that behavioral features perform as good as the hybrid, while the textual are thoroughly dwarfed by the former two. This is important because the computation cost associated with training the models on hybrid features is higher than the ones trained on the behavioral. Therefore, given their model training efficiency, using behavioral features instead of the hybrid should be desired.

4.4 Comparison with previous related works

In this section, we compare our work with some of the previously well-known state-of-the-art in the area of review-spam and review-spammer detection. Our aim is to analyse whether the proposed set of features (behavioral and textual both) provide any improvement over existing works in the literature. Therefore, this comparison is based on the features (adopted and devised) used in this paper and those proposed by some related works, in three different settings i.e. review, reviewer and product-centric. For comparison, we use three existing works: i) Mukherjee et al., (2013a), ii) Mukherjee et al., (2013c), and iii) Rayana & Akoglu (2015). Since Mukherjee et al., (2013c) do not provide enough review-related features, we only compare with the rest of the two works in review-centric setting. Features used in this work have already been discussed in Table 3 under Section 3.2. Table 7 summarizes the features used for comparison which are proposed by three comparing methods. Same experimental setup has been used as explained in Section 3.3.2. All four classifiers, SVM, LR, MLP, and NB, have been employed to train the models using features proposed by different works in three different settings. Performance of different works is compared based on two measures -Average Precision and ROC-AUC, as was previously used in Rayana and Akoglu (2015). Results are shown in Fig. 8 and Fig. 9 on YelpZip and YelpNYC datasets, respectively.

Research Paper

Table 7. Brief summary of features used by comparing methods under reviewer-centric, review-centric and product-centric settings.

Mukherjee et al. (2013a) Features		Mukherjee et al. (2013c) Features		Rayana & Akoglu (2015) Features	
<i>Reviewer-centric and Product-centric</i>	<i>Review-centric</i>	<i>Reviewer-centric and Product-centric</i>	<i>Review-centric</i>	<i>Reviewer-centric and Product-centric</i>	<i>Review-centric</i>
<i>CS</i>	<i>DUP</i>	<i>MNR</i>	–	<i>MNR</i>	<i>Rank</i>
<i>MNR</i>	<i>EXT</i>	<i>PR</i>	–	<i>PR</i>	<i>RD</i>
<i>BST</i>	<i>DEV</i>	<i>RL</i>	–	<i>NR</i>	<i>EXT</i>
<i>RFR</i>	<i>ETF</i>	<i>RD</i>	–	<i>avgRD</i>	<i>DEV</i>
		<i>MCS</i>	–	<i>WRD</i>	<i>ETF</i>
				<i>BST</i>	<i>PCW</i>
				<i>ERD</i>	<i>PC</i>
				<i>ETG</i>	<i>L</i>
				<i>RL</i>	<i>PP1</i>
				<i>ACS</i>	<i>RES</i>
				<i>MCS</i>	<i>SW</i>
					<i>OW</i>
					<i>DL_u</i>
					<i>DL_b</i>

From Fig. 8 and Fig. 9, it is quite clear that the features used in this work outperform those used in the other comparing works across all the classifiers on both the datasets with some minor exceptions. It is important to notice that in review-centric setting, for SVM classifier, the work of Rayana and Akoglu (2015) has significantly higher average precision and ROC-AUC score than the rest of the works as shown in Fig. 8 and Fig. 9. We find that the effect of this significance is greatly reduced when we observe the corresponding recall that is ~ 0.67 (for present study) and ~ 0.37 (for Rayana & Akoglu (2015)) on YelpZip and ~ 0.71 (for present study) and ~ 0.42 (for Rayana & Akoglu (2015)) on YelpNYC dataset. This clearly indicates that review-centric features used in the work of Rayana and Akoglu (2015) produce significantly low recall while significantly high average precision and ROC-AUC score, making the SVM classifier slightly biased. For rest of the two settings (reviewer-centric and product-centric), present study consistently outclasses all the comparing methods, making proposed set of features highly important in predicting spam, spammers and spam-targeted products. Our results shown in this section clearly indicate that the features which we have used in this study (some were adopted from previous works and some are introduced in this work) provide consistency and improvements in spam detection over existing features used in some well-established previous works.

4.5 Time computation comparison in feature extraction

In this Section, we perform the computation time analysis of behavioral and textual feature extraction under all the three settings. The motivation for this analysis

Figure 8. Performance comparison of different features used in different works on YelpZip over (a-b) review-centric, (c-d) review-centric, and (e-f) product-centric setting.

Figure 9. Performance comparison of different features used in different works on YelpNYC over (a-b) reviewer-centric, (c-d) review-centric, and (e-f) product-centric setting.

is drawn from the results obtained and discussed above. Our results reflected the distinctively better performance of behavioral features in comparison to the textual for opinion spam detection. This significant performance was consistent throughout all the settings irrespective of the classifiers and datasets as well. Therefore, it is intriguing to further explore the computation time efficiency of behavioral feature extraction relative to the textual. Fig. 10 shows the computation time of feature extraction for both behavioral and textual features over YelpZip (Fig. 10 (a)) and YelpNYC (Fig. 10 (b)) datasets. It can be seen from Fig. 10 (a) that on YelpZip, textual feature extraction requires too much computation time as compared to the behavioral by a factor of ~ 38 , ~ 103 and ~ 39 in reviewer-centric, review-centric and product-centric settings, respectively. Similar results are obtained in case of YelpNYC as shown in Fig. 10 (b). Here, textual features are again thoroughly beaten by the behavioral by factors of ~ 61 , ~ 151 and ~ 59 under reviewer, review and product-centric settings, respectively. These results are quite expected. As working with text involves several text mining and natural language processing techniques, which are computationally expensive and thus cost inefficient. As a concluding remark, these findings indicate that behavioral features not only perform better than the textual for opinion spam detection as discussed in Sections 4.1 and 4.2, but they are computationally inexpensive in feature extraction as well. This strengthens the case for behavioral features as the primary candidates for opinion spam detection.

Figure 10. Computation time analysis of behavioral and textual feature extraction on (a) YelpZip and (b) YelpNYC dataset.

4.6 Statistical significance analysis of obtained results

This section explains the statistical significance of the results presented in this study by performing two-tailed Z-test. The purpose of the same is to examine whether the obtained results contrasting behavioral and textual features are good enough to qualify as statistically significant. In two-tailed Z-test, test statistics is evaluated by using two hypotheses: null hypothesis ($H_0 : \mu_1 = \mu_2$) and alternative hypothesis ($H_a : \mu_1 \neq \mu_2$). Null hypothesis indicates that the mean value of first population (results on behavioral features) is equal to that of the mean of second

population (results on textual features), whereas alternative hypothesis opposes the same, meaning that there is statistically significant difference between the means of results obtained on behavioral and textual features. The Z-test is performed using 5% significance level (i.e. $\alpha = 0.05$) which rejects the null hypothesis if calculated value of Z-test is either greater than an upper critical value or smaller than a lower critical value (i.e. ± 1.96 for two-tailed Z-test). Table 8 shows the Z-test statistics and the corresponding P-values for the results obtained using SVM classifier on behavioral and textual features. As we can see from Table 8, Z-test statistic for each of the results is greater than the upper critical value at 5% significance level (P-value < 0.05) throughout all the settings across both the datasets. Therefore, the null hypothesis is rejected in all the cases, which shows the results obtained in our analyses (Sections 4.1 and 4.2) are statistically significant.

Table 8. Statistical significance of results obtained on behavioral and textual features using Z-test analysis.

		Reviewer-centric		Review-centric		Product-centric	
		Z-test statistic	P-value	Z-test statistic	P-value	Z-test statistic	P-value
YelpZip	ROC-AUC	30.03	~ 0.0	53.40	0.0	3.14	0.0016
	Avg. Precision	27.69	~ 0.0	37.58	~ 0.0	3.31	0.0009
	F1-Score (micro)	20.88	~ 0.0	48.91	0.0	2.07	0.0377
YelpNYC	ROC-AUC	23.02	~ 0.0	47.44	0.0	4.59	~ 0.0
	Avg. Precision	23.35	~ 0.0	33.48	~ 0.0	3.86	0.0001
	F1-Score (micro)	22.41	~ 0.0	30.17	~ 0.0	8.73	~ 0.0

5 Conclusion and future work

In this paper, we studied the impact of different categories of features on opinion spam detection. In essence, we aimed to identify such features which play the most dominating role in being the defining spam signatures. For the same, we examined the effectiveness of behavioral and textual features engineered from two benchmark datasets YelpZip and YelpNYC. To understand their versatility, this examination was conducted in three different spamming contexts i.e. review, reviewer and product-centric settings—each setting being pivotal in its own right. In context to spam, each setting is target-specific, where review-centric deals with review spam, reviewer-centric deals with review spammers and product-centric concerns the spam-targeted products. We performed feature engineering independently and proposed some new behavioral and textual features under three pre-defined settings. To examine the effectiveness of these features in detecting opinion spam, we trained four supervised classifiers i.e. SVM, LR, MLP and NB on both the datasets under the above-mentioned settings. To examine the effectiveness of proposed set of features, we then compared them against some previous well-known related works

Research Paper

under all the three settings. Our results indicate that behavioral features significantly outperform the textual across all the three settings on both the datasets. Moreover, there is significantly higher computation time associated with extracting textual features compared to the behavioral. This resulting cost efficiency of behavioral features further goes on to establish their higher precedence over textual features. Moreover, most importantly, the features used in this work are proved to be more efficient than the features used in previous related works. This study analyzed the efficiency of two categories of features i.e. behavioral and textual for detecting opinion spam. In future, we aim to find the optimal set of features required to further increase the efficiency of opinion spam detection. This is important because irrelevant features may hamper the overall spam detection performance.

Author contributions

Ajay Rastogi (ajay148115@st.jmi.ac.in) conceived of the presented idea, conducted the literature review and performed feature engineering under the guidance of Monica Mehrotra (mmehrotra@jmi.ac.in). Ajay Rastogi (ajay148115@st.jmi.ac.in) and Syed Shafat Ali (shafat159074@st.jmi.ac.in) together performed all the experiments and verified the empirical results. All authors discussed the results and contributed to the final manuscript.

References

- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007%2FBF00994018>.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>.
- Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R. (2013). Exploiting burstiness in reviews for review spammer detection. In *Proceedings of 7th International AAAI Conference on Weblogs and Social Media (ICWSM)*, pp. 175–184.
- Feng, S., Banerjee, R., & Choi, Y. (2012). Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 171–175.
- Fontanarava, J., Pasi, G., & Viviani, M. (2017). Feature analysis for fake review detection through supervised classification. In *Proceedings of 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 658–666. <https://doi.org/10.1109/dsaa.2017.51>.
- Heydari, A., Tavakoli, M., & Salim, N. (2016). Detection of fake opinions using time series. *Expert Systems with Applications*, 58, 83–92. <https://doi.org/10.1016/j.eswa.2016.03.020>.
- Jindal, N., & Liu, B. (2007a). Analyzing and detecting review spam. In *Proceedings of 7th IEEE International Conference on Data Mining (ICDM)*, pp. 547–552. <https://doi.org/10.1109/icdm.2007.68>.
- Jindal, N., & Liu, B. (2008). Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 219–230. <https://doi.org/10.1145/1341531.1341560>.

- Jindal, N., & Liu, B. (2007b). Review spam detection. In Proceedings of the 16th International Conference on World Wide Web (WWW), pp. 1189–1190. <https://doi.org/10.1145/1242572.1242759>.
- Li, F., Huang, M., Yang, Y., & Zhu, X. (2011). Learning to identify review spam. In Proceeding of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11), pp. 2488–2493.
- Li, J., Cardie, C., & Li, S. (2013). Topicspam: a topic-model based approach for spam detection. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 217–221.
- Li, J., Ott, M., Cardie, C., & Hovy, E. (2014). Towards a general rule for identifying deceptive opinion spam. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1566–1576.
- Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B., & Lauw, H.W. (2010). Detecting product review spammers using rating behaviors. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), pp. 939–948. <https://doi.org/10.1145/1871437.1871557>.
- Luca, M., & Zervas, G. (2016). Fake it till you make it: Reputation, competition, and Yelp review fraud. *Management Science*, 62(12), 3412–3427. <https://doi.org/10.1287/mnsc.2015.2304>.
- Mitchell, T.M. (2005). Logistic Regression. *Machine learning*, 10: 701.
- Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., & Ghosh, R. (2013a). Spotting opinion spammers using behavioral footprints. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 632–640. <https://doi.org/10.1145/2487575.2487580>.
- Mukherjee, A., Liu, B., & Glance, N. (2012). Spotting fake reviewer groups in consumer reviews. In Proceedings of the 21st International Conference on World Wide Web (WWW), pp. 191–200. <https://doi.org/10.1145/2187836.2187863>.
- Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2013b). Fake review detection: Classification and analysis of real and pseudo reviews. Technical Report UIC-CS-2013–03, University of Illinois at Chicago, Tech. Rep.
- Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N.S. (2013c). What yelp fake review filter might be doing?. In Proceedings of 7th International AAAI Conference on Weblogs and Social Media (ICWSM), pp. 409–418.
- 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-Volume 1, pp. 309–319.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Nov.), 2825–2830.
- Rastogi, A., & Mehrotra, M. (2018). Impact of behavioral and textual features on opinion spam detection. In Proceedings of the 2018 International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, pp. 852–857. <https://doi.org/10.1109/iccons.2018.8662912>.
- Rastogi, A., & Mehrotra, M. (2017). Opinion Spam Detection in Online Reviews. *Journal of Information & Knowledge Management*, 16(04): 1750036. <https://doi.org/10.1142/s0219649217500368>.

Research Paper

- Rayana, S., & Akoglu, L. (2016). Collective opinion spam detection using active inference. In Proceedings of the 2016 SIAM International Conference on Data Mining, pp. 630–638. <https://doi.org/10.1137/1.9781611974348.71>.
- Rayana, S., & Akoglu, L. (2015). Collective opinion spam detection: Bridging review networks and metadata. In Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994. <https://doi.org/10.1145/2783258.2783370>.
- Rish, I. (2001). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence, 3(22), pp. 41–46.
- Savage, D., Zhang, X., Yu, X., Chou, P., & Wang, Q. (2015). Detection of opinion spam based on anomalous rating deviation. Expert Systems with Applications, 42(22), 8650–8657. <https://doi.org/10.1016/j.eswa.2015.07.019>.
- Schurmann, J. (1996). Pattern classification: a unified view of statistical and neural approaches. John Wiley & Sons, Inc., New York.
- Sun, C., Du, Q., & Tian, G. (2016). Exploiting product related review features for fake review detection. Mathematical Problems in Engineering, 2016(1), 1–7. <https://dx.doi.org/10.1155/2016/4935792>.
- Wang, G., Xie, S., Liu, B., & Yu, P.S. (2012). Identify online store review spammers via social review graph. ACM Transactions on Intelligent Systems and Technology (TIST), 3(4), No. 61. <https://doi.org/10.1145/2337542.2337546>.
- Xie, S., Wang, G., Lin, S., & Yu, P.S. (2012). Review spam detection via temporal pattern discovery. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 823–831. <https://doi.org/10.1145/2339530.2339662>.
- Yuan, Y., Xie, S., Lu, C.T., Tang, J., & Philip, S.Y. (2016). Interpretable and effective opinion spam detection via temporal patterns mining across websites. In proceedings of 2016 IEEE International Conference on Big Data. pp. 96–105. <https://doi.org/10.1109/bigdata.2016.7840593>.

This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).