Online Review Spam Detection

Smriti Singh

Roll: 710CS1033
CSE department of NIT Rourkela
Under the guidance of
Prof. Sanjaya Kumar Jena

Table of Contents

What is Review Spam?

The proliferation of E-commerce sites has made web an excellent source of gathering customer reviews about products; as there is no quality control anyone one can write anything which leads to review spam. There has been a growth in **deceptive review spam** - fictitious opinions that have been deliberately written to sound authentic.

Why are product reviews so important?

1. They describe the products usability in the real life scenario.
2. It is a direct feedback on the product from the customers.
3. The customers who visit the website read the reviews written by other people and based on that take the decision to buy or not buy the product.
4. Reviews shape up market scenarios for products. Reviews written on major sites tend to make a product famous or push it down the drain. The social media outrage makes sure that such opinion is aired for everyone to tune in to.

Issues with spam detection

1. The untruthful reviews are crafted to look like genuine reviews with similar keywords.
2. Reviews are highly subjective and hence, can vary from a short simple description to a long paragraph.
3. There are a number of online sites available, so it is very difficult to ascertain if the reviewer has actually bought and used the product or not.
4. Sarcasm and witty reviews are commonplace in the online world. Such reviews are tougher to analyze.
5. There is no tagged dataset available to train spam models. Even when people were asked to tag reviews as spam, the concurrence rate amongst themselves was around 60%.

MOTIVATION AND OBJECTIVE

1. To investigate opinion spam in reviews and proposes some novel techniques to study spam detection.
2. The main task is to find a set of effective features for model building.
3. To detect review spams using the well known classifiers for labelled dataset.
4. Incorporate sentiment analysis into the spam review detection.
5. Also device an unsupervised method of review spam detection using clustering on unlabelled dataset.

Categories of opinion spam

Type 1 : Untruthful opinions

1. To promote some target objects (hype spam).
2. To damage reputation of other targets (defaming spam).

Type 2 : Reviews on brands only

1. not comment on the products for the products but only the brands, the manufacturers or the sellers.

Type 3: Non reviews

1. These are not actual reviews
2. Promotion of some other unrelated product
3. Advertisements

Type of spammers

An individual spammer

1. They register multiple times at a site using different user-ids.
2. They write either only positive reviews on own products or only negative on the products of competitors, but not both.
3. They give reasonably high rating, but write critical review.

A group of spammers

1. They write reviews when product is launched to take control of the product.
2. Every member reviews same product to lower rating deviation.
3. They divide group in sub-groups so that each sub-group can spam at different web sites.

Some observations about the nature of spammers

1. Given that the spammers write reviews professionally, it can be assumed that they have a set of words they may use frequently.
2. The spammer may want to mix in with the reviews of other people for the same product and hence his review can be very similar. [1]
3. The time when a review is posted is crucial. Early reviews get more weightage than the later ones.
4. Some sites provide a helpfulness score for the reviews, which can be seen as an indication to the authenticity of the review. Reviews written in quick succession can be seen as a red flag.

Data Set used

Item: 20 Hotels in Chicago area
Data Set size: 1.6MB
Number of reviews : 1600
Number of reviews per hotel: 80
Number of spam reviews per hotel : 40
Number of non-spam reviews per hotel:40
Number of positive reviews: 40
Number of negative reviews: 40
[2]

This data corpus contains:

1. 400 truthful positive reviews from TripAdvisor
2. 400 deceptive positive reviews from Mechanical Turk
3. 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp
4. 400 deceptive negative reviews from Mechanical Turk

Figure 1: Online Review Websites

Feature Vector Generation

Linguistic Characteristics as features

F1 : Quantity (Total Number of Words)
F2 : Complexity (Average number of words per sentence)
F3 : Diversity (Number of Unique words used)
F4 : Branding (Frequency of brand names used)
F5 : Average Word Length (Ratio of number characters to number of words)
F6 : Digits (Number of digits used)
[3] [4]

Genre Identification: POS Tagging as a feature

Frequency distribution of part-of-speech (POS) tags in a text often differentiates between informative and imaginative writing, namely that the former typically consists of more nouns, adjectives, prepositions, determiners, and coordinating conjunctions, while the latter consists of more verbs, adverbs, pronouns, and pre-determiners.

F7 : NN Number of Nouns
F8 : JJ Number of Adjectives
F9 : PRP Number of Prepositions
F10 : DT Number of Determiners
F11 : VB Number of Verbs
F12 : RB Number of Adverbs
F13 : PR Number of Pronouns
F14 : CC Number of Connector Words
F15 : IMM Number of First person pronouns

Text Categorisation: n-gram as a feature

Steps:

1. To model this behaviour we consider the unigram and bigram feature sets, with the corresponding features(words and expressions) lowercased and unstemmed and maintain a dictionary from the training set.
2. Each review was then broken into corresponding N-gram and checked for the following scores.
3. This score is calculated on the basis of presence or absence of an N-gram in the spam set or the non-spam set in terms of 1 or 0.

F16: SpamHitScore
F17: NonSpamHitScore (Scores indicating how much the words of a review are similar to the spam reviews)

Sentiment as a feature

The fake negative reviewers are seen to over-produce negative emotion terms relative to the truthful reviews in the same way that fake positive reviewers over-produced positive emotion terms. Therefore, fake hotel reviewers exaggerate the sentiment. [1]

1. Extract features/aspect nouns from each sentence in the review.
2. We corresponding sentiment words and their polarity.
3. Strength of the sentiment word on the feature decreases with the distance from the feature word.
4. We calculate the number of negation words to reverse polarity due to negative words present.
5. Finally the aggregation if all feature scores and then its mean gives us the sentiment score in the range [-1,+1].

r = review
f = aspect/feature in a sentence
o(wj): sentiment polarity of a word wj (+1 or -1)
cn: no. of negation words in one feature, default = 0
dist(wj,f) = distance between feature f and word wj.
totss(r) = total sentiment score of a review

$$totss(r) = \frac{\sum(-1)^{c_n} o(w_j)}{dist(w_j, f)}$$

Figure 2: Sentiment Score Calculation

F18 : Sentiment Score

Linguistic Features Analysis

The linguistic model works averagely well and the results are shown in Table. However an important observation is that such a simplified analysis can also yield results that are comparable to that of a human classifying the same data for the same dataset. [2]

Approach	Features Considered	Train set size (in %)	Classifier Used	Accuracy (%)
Linguistic Features	Linguistic features vector	70	Naive Bayes	72.04
			SVM	72.1
			Decision Tree	64.60
		80	Naive Bayes	73.25
			SVM	73.25
			Decision Tree	69.00
		90	Naive Bayes	74.02
			SVM	70.89
			Decision Tree	73.2

POS Features Analysis

POS Features analysis also gives us an average result but its not as good as the results given by the linguistic analysis. Hence in the next sections, we combine the two methods.

Approach	Features Considered	Train set size (in %)	Classifier Used	Accuracy (%)
POS Features	POS Features vector	70	Naive Bayes	68.6
			SVM	63.8
			Decision Tree	66.6
		80	Naive Bayes	67.75
			SVM	62.25
			Decision Tree	71.11
		90	Naive Bayes	72.89
			SVM	66.52
			Decision Tree	68.5

n-gram Features Analysis

The accuracy is better than the linguistic and POS models.
The frequent word set used by the spammers and those who write genuine reviews is different enough to help us tag spam behaviour.
This validates our initial hypothesis.

Approach	Features Considered	Train set size (in %)	Classifier Used	Accuracy (%)
n-gram Features	n-gram Features vector	70	Naive Bayes	73.33
			SVM	73.65
			Decision Tree	72.6
		80	Naive Bayes	72.7
			SVM	76.11
			Decision Tree	73.62
		90	Naive Bayes	96.5
			SVM	88.5
			Decision Tree	96.65

n-gram	Classifier	Accuracy
Bigram	Naive Bayes	73.5
Bigram	SVM	63.75
Bigram	Decision Tree	73.5
Unigram + Bigram	Naive Bayes	71.1
Unigram + Bigram	SVM	60.01
Unigram + Bigram	Decision Tree	71.83

Sentiment Features Analysis

The sentiment scores definitely bring about an increase in the accuracy obtained when combined with the other features.

Features Used	Classifier	Accuracy
Sentiment Score + Linguistic	Naive Bayes	74.5
Sentiment Score + Linguistic	SVM	72.02
Sentiment Score + Linguistic	Decision Tree	75.8
Sentiment Score + POS	Naive Bayes	72.5
Sentiment Score + POS	SVM	70.02
Sentiment Score + POS	Decision Tree	75.7
Sentiment Score + Ling + POS	Naive Bayes	78.9
Sentiment Score + Ling + POS	SVM	74.5
Sentiment Score + Ling + POS	Decision Tree	76.6

Unified Features Model Analysis

The N-gram model alone seems to overfit the data points and does not include any features of the spammer other than the words they use. Combining the previous two models gives more reasonable results and a more realistic modelling of the data set, as shown in Table. The accuracy levels still remain fairly higher than most work in this area. We obtain about 92.11 % accuracy level obtained by combining the POS, linguistic sentiment and the unigram feature vectors.

Features Used	Classifier	Accuracy
Sentiment Score + Ling + Unigram Model	Naive Bayes	91.9
Sentiment Score + Ling + Unigram Model	SVM	88.7.1
Sentiment Score + Ling + Unigram Model	Decision Tree	92.11

Observations

1. Unified model work quite effectively in detecting spam based on just the review text.
2. The linguistic/POS features offer secondary support to the decision model.
3. User metadata like number of reviews written, the timeframe in which he writes the reviews, the geolocation or check in data if available from other sources to verify if the user was actually present at the venue, age of the user etc. can be crucial elements in determining if a review is fraudulent.

Data Set Collection

Amazon provides its review data in public interest. The data set is available as categorized in various genres of products. For this analysis, a data set for Cell Phones and Electronics products was used. The data set has 78,930 reviews with each review described as a key value pair shown below.

```
product/productId: B00006HAXW
product/title: Rock Rhythm & Doo Wop: Greatest Early Rock
product/price: unknown
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or
better than the 1st ones. Remember once these performers are gone, we'll
never get to see them again. Rhino did an excellent job and if you like or
love doo wop and Rock n Roll you'll LOVE this DVD !!
```

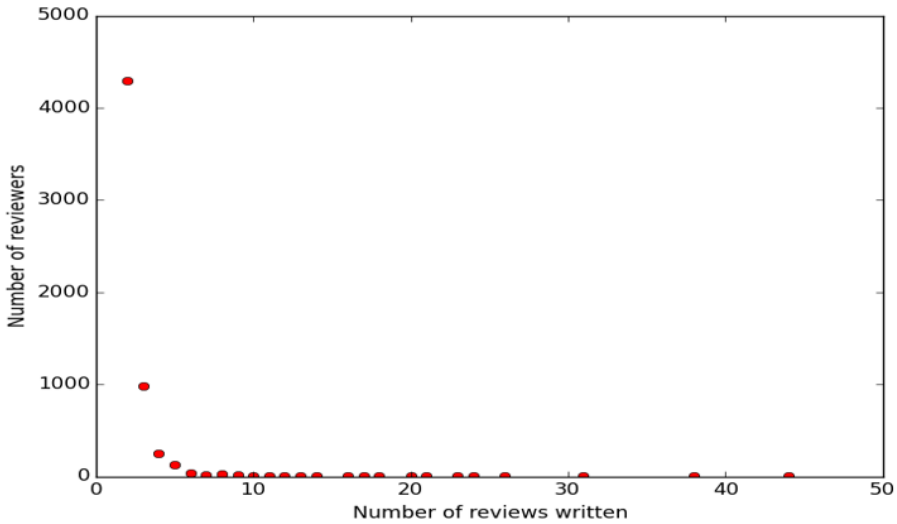
Figure 3: Example review data

Bulk Analysis Of Review Data Set

The dataset was analyzed for a number of features described in the next sub section.

1. Number of reviews vs. number of reviewers

An interesting observation from the plot is 91% have written 1 review, 99.25% of reviewers have written 3 or less number of reviews.



2. Number of reviews vs. number of products

We can see that a large number of products get very few reviews and a small number of products get a large number of reviews.

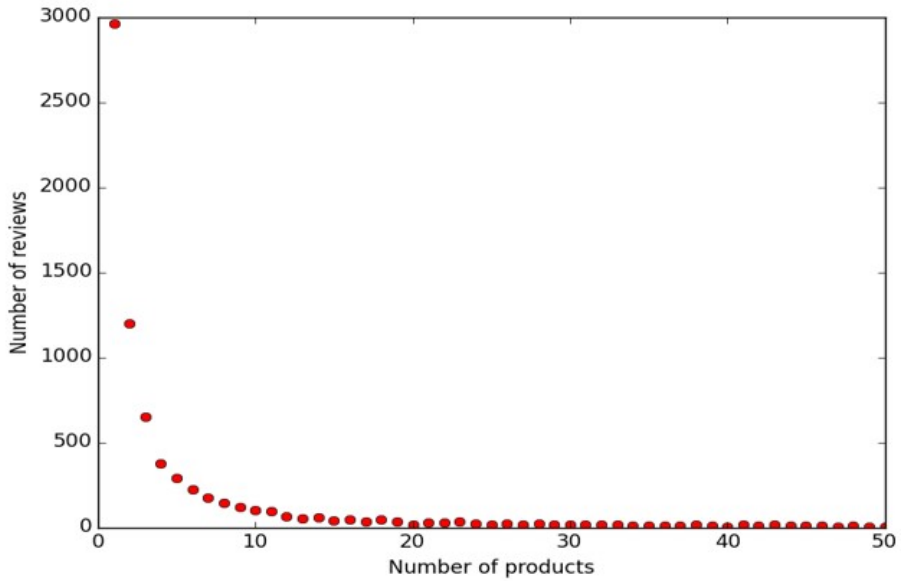


Figure 5: Number of reviews vs. number of products

3. Rating vs. percent of reviews 60.77% reviews have a rating of 4 and above.

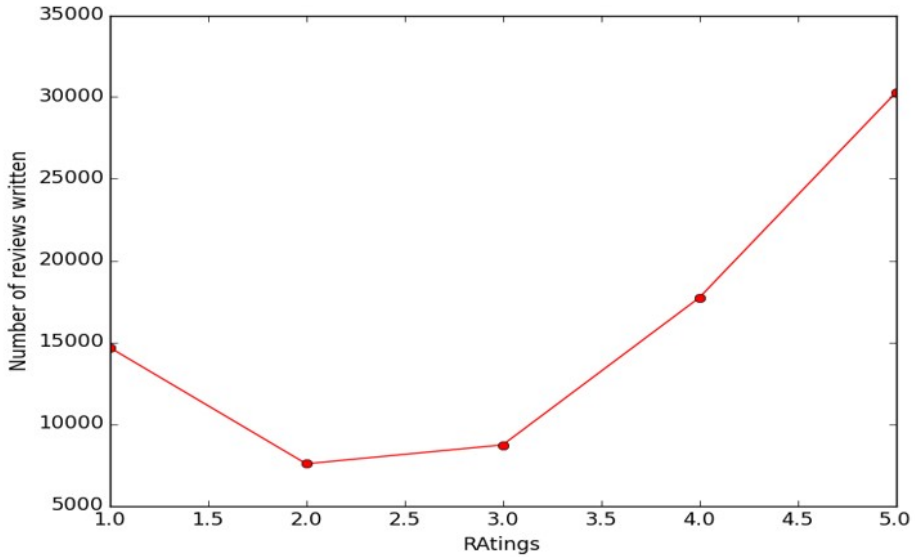


Figure 6: Rating vs. percent of reviews

Feature Vector Generation

The main task is to find a set of effective features for model building. Information contained in the reviews can be categorised as three main types:

1. Review Centric Features
2. Reviewer Centric Features
3. Product Centric Features

A. REVIEW CENTRIC FEATURES (I):

F1. Number of feedbacks
F2. Number of helpful feedbacks
F3. Percentage of helpful feedbacks that the review gets
F4. Length of the review title
F5. Length of review body
F6. Position of the review in the reviews of a product sorted by date, in both ascending
F7. And descending order.
F8. If a review is the first review
F9. If a review is the only review.

A. REVIEW CENTRIC FEATURES(II):

Textual features:
F10. Percent of positive and
F11. Negative opinion bearing words
F12 Percent of numerals,
F13 capitals and
F14 all capital words in the review.
Rating related features:
F15. Rating of the review
F16. Deviation from product rating.
F17. Binary features indicating whether a negative review was written just after the first good review of the product and vice versa
F18.

B. REVIEWER CENTRIC FEATURES:

F19. Ratio of the number of reviews that the reviewer wrote which were the first reviews

F20. Ratio of the number of cases in which he/she was the only reviewer

F21. Average rating given by reviewer

F22. Standard deviation in rating given by reviewer

C. PRODUCT CENTRIC FEATURES:

F23. Price of the product.

F24. Average rating of a product

F25. Standard deviation in ratings of the reviews on the product.

Outlier Detection Using k-NN Approach For Outliers

```
X = get_OutlierScores(D,k)
For all p,OutlierScorep in X
  if OutlierScorep >= M
    Add p to L

get_OutlierScores(D,k)
  if |D| != Null
    for all p in D
      S = getK_NearestNeighbours(D,p,k)
      for all q in S
        T = getK_NearestNeighbours(D,q,k)
        if p in T:
          Add q to ForwardNNk(p)
          | ForwardNNk(p) | = | ForwardNNk(p) | +1
  for p in D
    OutlierScore(p) = 1 - |ForwardNNk(p)| / (|D|-1)  return [p,OutlierScore(p)]

getK_NearestNeighbours(D,p,k)
  if D != Null
    for all q in D and p !=q
      Compute dist(p,q)
  sort(dist(p,q))
  Add k shortest distant objects from p to NNk(p)
  return NNk(p)
```

Results

Table 1: Outlier Spam Detection

Total Number of reviews	78930
Number of Spam reviews detected	6064
Percentage of Spam	7.68 %

This simplistic model gives us around 6064 potentially fake entries which can be used as a training set for other regressions. For example, Ott et al. (2012) [5] have estimated between 1% and 6% deceptive reviews in any dataset. Thus our results are consistent with their observations.

Conclusion

1. About 6000 potentially fake entries were obtained from our model. This can be used to build a training once they are authenticated to be spams.
2. The findings in the bulk analysis can be incorporated into the sequential analysis so that fake reviews can give a red flag as soon as they are submitted.
3. The sentiment of the reviews is also something that can be incorporated in the model.

Selected References I

[1] Qingxi Peng and Ming Zhong.
Detecting spam review through sentiment analysis.
Journal of Software, 9(8):2065–2072, 2014.

[2] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock.
Finding deceptive opinion spam by any stretch of the imagination.
In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics, 2011.

[3] C Harris.
Detecting deceptive opinion spam using human computation.
In *Workshops at AAAI on Artificial Intelligence*, 2012.

[4] Kyung-Hyan Yoo and Ulrike Gretzel.
Comparison of deceptive and truthful travel reviews.
Information and communication technologies in tourism 2009, pages 37–47, 2009.

Selected References II

[5] Myle Ott, Claire Cardie, and Jeff Hancock.
Estimating the prevalence of deception in online review communities.
In *Proceedings of the 21st international conference on World Wide Web*, pages 201–210. ACM, 2012.

[6] Xia Hu, Jiliang Tang, and Huan Liu.
Online social spammer detection.
In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[7] Arjun Mukherjee, Bing Liu, and Natalie Glance.
Spotting fake reviewer groups in consumer reviews.
In *Proceedings of the 21st international conference on World Wide Web*, pages 191–200. ACM, 2012.

[8] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S Glance.
What yelp fake review filter might be doing?
In *ICWSM*, 2013.


Thank You!