Fake Review Detection

Arry Fajar Firdaus
University of Southern California
Los Angeles, CA
firdaus@usc.edu

Jayavanth Shenoy
University of Southern California
Los Angeles, CA
jshenoy@usc.edu

Raghav Bharadwaj
University of Southern California
Los Angeles, CA
rjayasim@usc.edu

Abstract

E-commerce activity is increasing consistently around the world. Buyers rely on reviews on these websites to buy products. This situation has led to an increasing amount of opinion spam of product reviews. These deceptive reviews are written by people who are paid by some organizations to glorify their product or vilify a competitor's product. We explore several techniques for constructing a dataset of fake reviews. The constructed dataset is then used to build models that can classify a review as fake. With our dataset and models, we get an F1 score of 80% for our truthful opinion model and an F1 score of 75% for our deceptive opinion model. We have also developed the first publicly available fake review dataset for Amazon reviews.

1 Introduction

In this era of unprecedented technology and connectivity, there has been an exponential increase in online shopping activity - buying and selling on the web. With myriad products and sellers in a highly competitive market place, it is not uncommon that businesses enhance their sales by resorting to unethical sales strategies – writing fake reviews being one of them.

Detecting fake reviews and eliminating them would make the online market place fair and transparent. It not only helps customers make better decisions but also help manufacturers improve their product based on honest feedback from customers.

Also, the task is challenging - for it is hard to get real fake reviews to train a model in a supervised learning approach. Most existing research is based on fake reviews that were written primarily for research and may not accurately reflect the real fake reviews written by professionals.

In this project, we attempt to explore solutions to detecting fake reviews on Amazon. Complete overview of the techniques discussed in this paper is illustrated in Fig. 2. Further sections discuss each part of it.

We choose Amazon's Electronics domain for this study. Detecting fake reviews on sites like tripadvisor has already been explored by Ott et al. (2011). The scale of Amazon.com would give a true understanding of the complexity of the problem, with products ranging across a multitude of domains.

An extension to this work would be to explore various methods for Behavioral Analysis of reviews discussed in Mukherjee et. al (2011) and extend our domain to cover all of Amazon's product categories.

2 Related Work

Fake review or opinion spam detection techniques can be categorized into supervised and non-supervised techniques (Liu, 2012). A straightforward supervised approach is to manually annotate the reviews with deceptive and truthful label like in (Li et al., 2011). Detecting a fake review is hard task to do manually, but it may be easier to make one. Ott et al. (2011) developed gold standard fake review data set using Mechanical Amazon Turk crowdsourcing service to gather reviews on 20 hotels. Another method is based on the assumption that a fake reviewer use the same words for most reviews they write. Jindal and Liu (2008) developed a heuristic to detect fake reviews based on review similarity.

Some unsupervised spam detection techniques are based on detecting unusual behavior of reviewers. Thus, these methods focus on detecting fake reviewers rather than the reviews. Lim et al

(2010) identified some behavior patterns that indicate opinion spam or fake review:

- Fake reviewers tend to target several products within short span of time
- Fake reviewers tend to focus on limited number of products
- Ratings given by fake reviewers deviate from other reviewers
- Some fake reviewers give reviews shortly after product launch

Another unsupervised method described in (Jindal et al, 2010) uses data mining technique to discover class association rule that deviates from normal behavior. This technique is used in this project and will be further explained in later section.

Another approach is to detect group of fake reviewers as proposed in (Mukherjee et al., 2011). A reviewer may work individually or as a member of a group. A group may be individual users who work in collusion or a single user with multiple user ID. The algorithm seeks a group of reviewers who may collaborate to promote or demote some products.

3 Data

We use a supervised learning approach to solve this problem.

We gather data using the strategies for truthful and deceptive reviews discussed in further sub-sections to build our data set.

3.1 Data for deceptive reviews

We obtained about 900 reviews for our deceptive review data set.

3.1.1 Manual Annotation

Prof. Bing Liu's website[1] led us to interesting articles about spotting a fake review manually. We made a list of those techniques and used them as annotation guidelines. We could identify some reviews as clearly deceptive, however, this process was time consuming and we were left with a lot of reviews that were ambiguous. We realised this method would not scale to our needs and decided to move on.

3.1.2 Crowd-sourcing

Previous research for similar studies (Ott et al. ,2011 and Mukherjee et. al. 2013) used crowd-sourcing approach to build their fake review data set. We made an appeal on social media platforms like facebook and our class discussion board Piazza to contribute a fake review. We used Google Forms to list items. This did not give us a good yield as we ended up with less than a hundred reviews from all users.

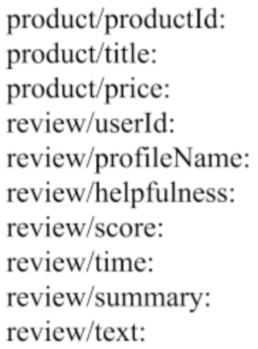
3.1.3 Cosine Similarity

Mukherjee et. al. (2013), in their research observe that over 70% of opinion spammers have a very high content similarity in their reviews. They also observe that over 70% of non-spammers have very low content similarity.

We considered reviews with a cosine similarity of over 90% for the reviews authored by the same reviewer as opinion spam and used this data in the language model.

3.1.4 Class Association Rule (CAR)

We used Amazon review data sets from Stanford Network Analysis Project[2] referred in McAuley et. al., (2013). They have a large collection of Amazon review data set across various domains. The reviews were available in the format described in Fig. 1. Having parsed them and formatted the reviews based on our needs, we used the ideas of Cosine Similarities and Class Association Rules described in Section 4. We obtained a large amount of fake review data using this technique.



product/productId:
product/title:
product/price:
review/userId:
review/profileName:
review/helpfulness:
review/score:
review/time:
review/summary:
review/text:

Figure 1: SNAP data format

[1] http://www.cs.uic.edu/~liub/FBS/fake-reviews.html

[2] https://snap.stanford.edu/data/web-Amazon.html

3.2 Data for truthful reviews

We built our positive truthful reviews data set by crawling (using Apache Nutch) and scraping (manual) reviews on Amazon.com which were five-star rated, most-helpful and verified purchases of best seller products on Amazon. We obtained about five hundred reviews for our truthful review data set.

4 Technical approach

4.1 Cosine Similarity

Mukherjee et. al. (2013), in their research observe that over 70% of opinion spammers have a very high content similarity in their reviews. They also observe that over 70% of non-spammers have very low content similarity.

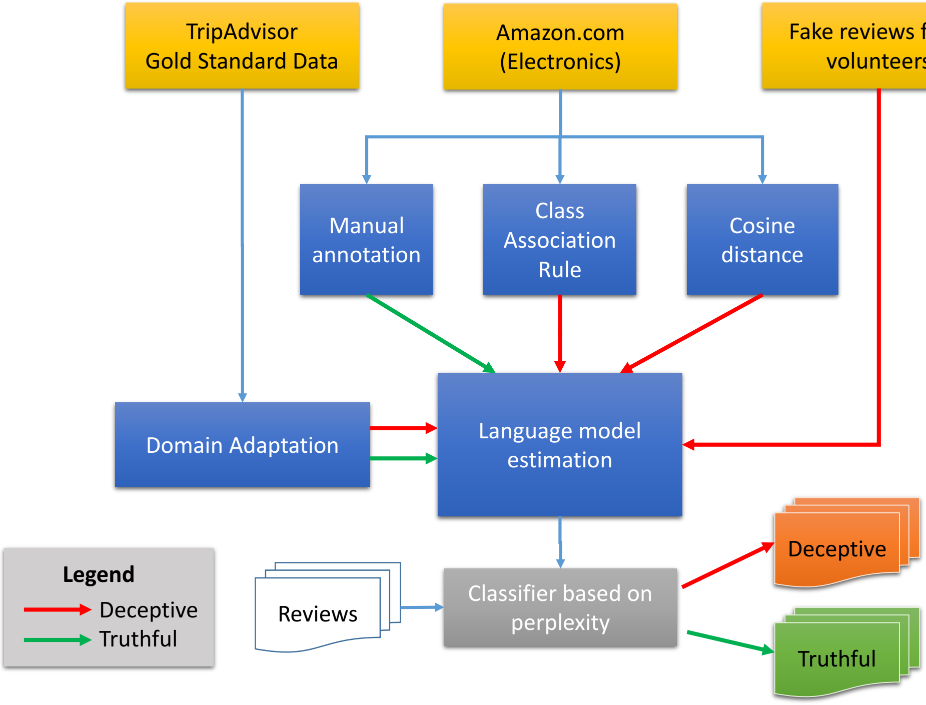


Figure 2: Architecture

We compute pairwise Cosine similarity using various modules in the Natural Language ToolKit (NLTK and some functions in Numpy library. We build an n-dimensional vector space using all the documents and represent each document as a TF-IDF vector in the space. We then compute pairwise Cosine Similarity. Identical documents have a cosine similarity measure of 1.

We consider reviews with a cosine similarity of over 0.9 for the reviews authored by the same reviewer as opinion spam and used this data in the language model.

4.2 Class Association Rule

This method identifies atypical behavior of reviewers based on association rule mining. The difference between association rule mining and class association rule is that the latter has a predetermined target class attribute (Liu, 1998).

Each record in the source data contains a set of attributes $A = \{A_1, A_2, \ldots, A_m\}$ and a class $C = \{c_1, c_2, \ldots, c_n\}$. The objective of class association rule (CAR) is to find attributes $X \in A$ that fulfill implication $X \rightarrow c_i$ within bound of minimum confidence Pr $(c_i|X)$ and support Pr (X, c_i).

The benefit of this method is that it is domain independent and only depend on the data and type of rules but not application. In fake review detection application, the dataset contains a set of attributes *reviewer-id*, *brand-id*, *product-id* and a class (Jindal, 2010). The class represent sentiment of the reviewer on the product, i.e. *positive*, *neutral* and *negative*. Rating of 4 and 5 is assigned positive class. Rating of 1 and 2 is negative, and rating of 3 is neutral. In this application we limit the number of conditions in CAR mining to two because longer rules will result to little amount of data that is impractical to do operation with.

A measure is needed to compare resulting rules against natural distribution of values. Such measure is called unexpectedness and defined as deviation from expectation. (Jindal, 2010) defines four types of unexpectedness but only two are used in this project. The first one is confidence unexpectedness that is defined as how unexpected the confidence of a rule is. The other type, support unexpectedness, measures how unexpected proportion of records involved.

The unexpectedness measure then can be used to rank rules, and the result, for example, can be used to answer these questions:

- Confidence unexpectedness (one condition rule): Which reviewer writes only positive/negative reviews?
- Confidence unexpectedness (two condition rule): Which reviewer only writes positive reviews for a specific brand?
- Support unexpectedness (one condition rule): How many positive/negative reviews a reviewer write?
- Support unexpectedness (two condition rule): How many positive/negative reviews a reviewer write for a pecific brand?

The electronics category of Amazon review dataset obtained from (McAuley, 2013) is used to derive the class association rules in this project. The Amazon dataset has product data but does not contain brand, so the missing data is obtained by looking up Amazon website for each product using `amazon_get_brand2.py` script.

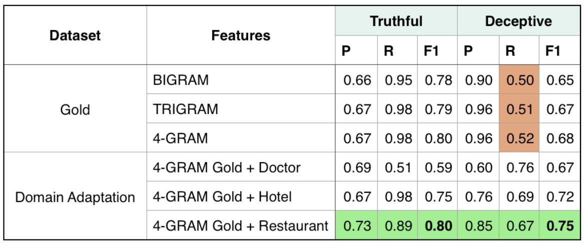
`amazon_preprocess.py` preprocess Amazon review dataset into a format that is easier to process on subsequent steps. The class association rule mining itself is done using `amazon_car.py` script. The output of the script is the names of suspected fake reviewers. Reviews by these suspects will be the input of language model estimation described in next section.

Top ranked rules show that out of 70,090 reviewers who wrote more than 3 reviews, 31,559 of them wrote only positive reviews and there are 2,206 wrote only negative reviews.

Other interesting findings obtained by CAR on Amazon review dataset is that some reviewers wrote 30-50 all positive reviews for a single brand. Some of the positive reviews goes to a non-name brands. These behaviors are highly suspicious.

5 Evaluation and Analysis

Using the data that is obtained from using construction methods described in Section 3, we develop two language models: Truthful and Deceptive. We use The CMU Statistical Language Modeling Toolkit (Clarkson et. al., 1997) to build language models.



Dataset	Features	Truthful			Deceptive		
		P	R	F1	P	R	F1
Gold	BIGRAM	0.66	0.95	0.78	0.90	0.50	0.65
	TRIGRAM	0.67	0.98	0.79	0.96	0.51	0.67
	4-GRAM	0.67	0.98	0.80	0.96	0.52	0.68
Domain Adaptation	4-GRAM Gold + Doctor	0.69	0.51	0.59	0.60	0.76	0.67
	4-GRAM Gold + Hotel	0.67	0.98	0.75	0.76	0.69	0.72
	4-GRAM Gold + Restaurant	0.73	0.89	**0.80**	0.85	0.67	**0.75**

Figure 3: Language Model Evaluation

Fig. 3 describes our experiments with models trained on BIGRAMS, TRIGRAMS and 4-GRAMS. We got decent F1 scores of 0.80 (Truthful) and 0.52 (Deceptive) on our model when the model was trained on 4-GRAMS. On observation, we can see that for the original Amazon Electronics dataset (a.k.a Gold), the recall for the deceptive model – marked in Red – is low. This indicates that the model has trouble recognizing a fake review when given one. A low recall is also indicative of less data provided to train that model. So, we decided to make use of readily available fake review data from Myle Ott et. al. (2011). Note that the data in that work is from different domains viz. Doctor, Hotel (from Tripadvisor), and Restaurant reviews. Myle Ott et. al. were able predict with an accuracy of up to 90% using this dataset. This is considered the state-of-the-art in the domain of fake review detection in general. Our domain adaptation technique on this dataset is described in the next sub-section. We have made all the data constructed by us in this experiment public[3]

We conducted several experiments which did not give fruitful results. Here are some interesting observations on them:

a. Models built on UNIGRAM and >4-GRAM models were performing badly. Usually, this happens when models have less context (for UNIGRAM) and less data (for 5-GRAM and above).
b. Models based entirely on fake data constructed from Section 3.13 did not do well. Primarily because this method is

[3] bit.ly/fake_review

not fool-proof. For instance, reviews with ≈150 characters with >95% similarity are completely different. We expected to find reviews that had same structure but different product names. But having high cosine similarity reviews with a high character count increased the F-Score.

c. Models based just on documents separated using manual annotation and crowd-sourced data. This failed mainly because of lack of data and the quality of data

d. Using >68% of reviews from other domains for training the models. There can be many reasons behind this. Myle Ott et. al. (2011) discussed in his paper that these domains have some suprising characteristics by itself – Models trained only on UNIGRAMS outperformed all other approaches. Furthermore, since we are using language models, lack of data in those datasets (just 200 reviews per class per domain) might be a reason for low F1 scores.

5.1 Domain Similarity

We compute domain similarity as a measure of cosine similarity between two domains. McClosky et. al. (2010) discusses that the success of domain adaptation depends on how similar the two domains are. We observed that the reviews from hotel and restaurants domains are closer to Amazon Electronics reviews than reviews from Doctor domain. Refer Fig. 4 for numbers.

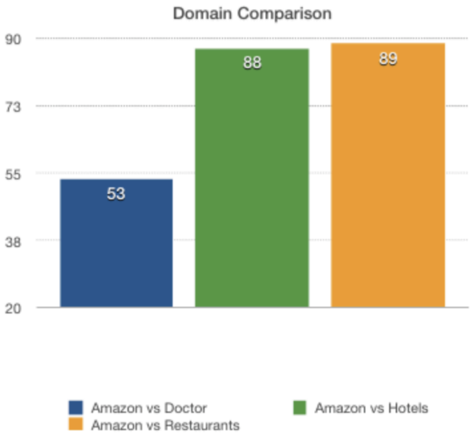


Fig. 4: Domain Comparison

5.2 Domain Adaptation

We repeated our experiments after adapting the three domains described in the previous sections. The detailed results are given in Fig. 3 and the domain F1-scores are compared in Fig.5, which directly follows the structure of Fig.4. In the best case, we got F1 scores of **0.80** (Truthful) and **0.75** (Deceptive) for Restaurant domain and just following that is the F1 for Hotel and Doctor domains. As observed in Fig. 3, our results with domain adaptation agrees with the hypothesis described in section 5.1.

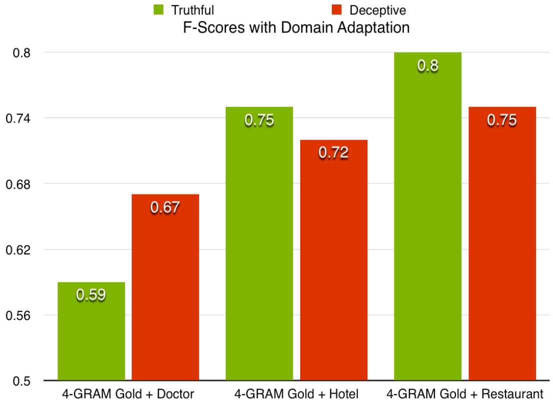


Figure 5: Domain Adaptation comparison

6 Conclusion

We have observed that methods used by humans to detect fake reviews is hard to scale. We used methods described in Sections 3.2 and 4.1 to find fake reviews and fake reviewers and used this data to build a model of fake reviews. We also used data (domain adapted) from publicly available fake review datasets to amplify our language models.

Finally, we have developed the first publicly available gold standard dataset of Amazon fake reviews[4]. We have also open-sourced our code[5]

7 Future Work

Directions for future work include training with more data and from all domains on Amazon.com, implementing other existing methods of finding fake reviews given in Mukherjee et. al (2013), using psychoanalytic features as an addition to the language model, experimenting with classification techniques, exploiting metadata of reviews to either construct new features or as a sort of filtering mechanism. Incorporating behavioral features of users in the model have

[4] bit.ly/fake_review

[5] https://bitbucket.org/nlpfakereview/fake-review-detection/

been known to boost F1 scores (Mukherjee et. al 2013).

Acknowledgements

We would like to thank Prof. Kenji Sagae and Justin Garten for giving us this opportunity and for guiding us throughout the project.

We would also like to acknowledge our friends for contributing to the fake review data. We saw boosts of up to 2% on our deceptive F-scores.

References

Arjun Mukherjee, Bing Liu, Junhui Wang, Natalie Glance, and Nitin Jindal. *Detecting Group Review Spam*. In *Proceedings of International Conference on World Wide Web (WWW-2011, poster paper)*. 2011.

Bing Liu, Wynne Hsu, and Yiming Ma. *Integrating classification and association rule mining*. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD-1998)*. 1998.

Bing Liu. "Sentiment analysis and opinion mining." *Synthesis Lectures on Human Language Technologies* 5.1 (2012): 1-167.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady W. Lauw. *Detecting Product Review Spammers using Rating Behaviors*. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2010)*. 2010.

Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. *Learning to Identify Review Spam*. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI -2011)*. 2011.

Nitin, Jindal and Bing Liu. *Opinion spam and analysis*. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM-2008)*. 2008.

Julian McAuley and Jure Leskovec. *Hidden factors and hidden topics: understanding rating dimensions with review text*. RecSys, 2013.

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 (AC L-2011)*. 2011.

Nitin Jindal, Bing Liu, and Ee-Peng Lim. *Finding Unusual Review Patterns Using Unexpected Rules*. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2010)*. 2010.

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, Natalie Glance. *What Yelp Fake Review Filter Might Be Doing?*. In *Proceedings of The International AAAI Conference on Weblogs and Social Media (ICWSM-2013), July 8-10, 2013, Boston, USA.*

P Clarkson, R Rosenfeld. *Statistical language modeling using the CMU-cambridge toolkit.* In *Eurospeech, 1997 - 158.130.67.137.*

David McClosky, Eugene Charniak, Mark Johnson. *Automatic Domain Adaptation for Parsing.* In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 28–36, Los Angeles, California, June 2010.*

Work distribution:

Distribution of work according to the person's contribution to the Project and their contribuition in writing the Report.

	Project	**Report**
Arry Fajar Firdaus	Initial experimental model using KenLM, Class Association Rule, Obtained large amounts of the fake review data after post-processing reviews obtained from using Class Association Rule, Associated scripts	Created formatted document and initial structure of sections, Related Work (Section 2), Class Association Rules (Section 4.2), Fig. 2, Collection of bibliography (References section)
Jayavanth Shenoy	Conducted experiments listed in evaluation section with CMU LM and different data sets, Domain Adaptation, Crawled using Nutch to get truthful reviews, Manually scraped reviews for truthful reviews (50%), Designed google form for crowd-sourced fake review http://bit.ly/1ONKx1Y, Associated scripts	Abstract, Evaluation and Analysis (Section 5), Fig. 3, Fig. 5, Footnotes and links Conclusion and Future Work (Section 6 and 7)
Raghav Bharadwaj	Manually scraped reviews for truthful reviews (50%), Manual Annotation based on guidelines, Cosine similarity for separating fake reviews from Amazon review dataset, Script to extract reviews in required format from raw data sets for cosine similarity, Conducted experiments on Domain Similarity using Cosine similarity, Associated scripts	Introduction (Section 1), Data (Section 3), Cosine similarity (Section 4.1), Domain similarity (Section 5.1), Figure 1 and 4