Classification for the Detection of Opinion Spam

GHISLAINE VAN DEN BOOGERD (9489479), ANTONIO OLIVA HERNÁNDEZ (0685143), and OTTO MÄTTAS (6324363), Utrecht University, The Netherlands

1 INTRODUCTION

The world of online review websites is growing in popularity. Considering word to mouth advice is still seen as one of the most effective ways for advertisement [4], the rise in popularity for these websites comes paired with possibilities for monetary gain. With 92% of users making a purchase after reading reviews on Yelp[1]- an online review website, it gives online reviews a lot of value. However, this also opens the ways to deceptive reviews on the platforms to lure more business. Detecting fake reviews can be a very effective tool to keep online review websites fair and useful. Classifying deceptive text has been done before, but there are still new ways in which we can approach the subject. This paper aims to classify the detection of opinion spam from the Deceptive Opinion Spam Corpus [2][16, 18].

2 RELATED WORK

Spam has historically been studied in the contexts of e-mail [3], and the Web [5, 14]. In the past few decades, researchers have began to look at opinion spam as well [6, 23, 24]. In relation to current research, negative opinion spam has been most prominently investigated by Ott et al.[15, 17]

Jindal and Liu[6] find that opinion spam is both widespread and different in nature from either e-mail or Web spam. Using product review data, and in the absence of gold-standard deceptive opinions, they train models using features based on the review text, reviewer, and product, to distinguish between duplicate opinions (considered deceptive spam) and non-duplicate opinions (considered truthful). Wu et al.[23] propose an alternative strategy for detecting deceptive opinion spam in the absence of gold-standard data, based on the distortion of popularity rankings. Both of these heuristic evaluation approaches are unnecessary in our work, since we compare gold-standard deceptive and truthful opinions.

Yoo and Gretzel [24] gather 40 truthful and 42 deceptive hotel reviews and, using a standard statistical test, manually compare the psychologically relevant linguistic differences between them. In contrast, we have a much larger data set of 800 opinions that we use to develop and evaluate automated deception classifiers.

Research has also been conducted on the related task of psycholinguistic deception detection. Newman et al.[13], and later Mihalcea and Strapparava[12], ask participants to give both their true and untrue views on personal issues (e.g., their stance on the death penalty). Zhou et al. [25, 26] consider computer-mediated deception in role-playing games designed to be played over instant messaging and e-mail. However, while these studies compare n-gram–based deception classifiers to a random guess baseline of 50%, the team additionally evaluate and compare two other computational approaches described in 4.

Lastly, automatic approaches to determining review quality have been studied — directly [22], and in the contexts of helpfulness [2, 8, 19] and credibility [21]. Unfortunately, most measures of quality employed in those works are based exclusively on human judgments, whereas the researchers aim at computational and objective methods of evaluation with this work.

3 DATA

The paper uses the Deceptive Opinion Spam Corpus. The corpus consists of both truthful and deceptive reviews about the 20 most popular Chicago hotels. The corpus is created by Ott et al. and consists of two parts, where one discusses positive sentiments reviews[18] and the other negative sentiment reviews[16].

The first part of the corpus consists of both truthful and deceptive 5-star reviews [18]. The truthful positive reviews are sourced from TripAdvisor [3] of the 20 most popular hotels in the Chicago area. The reviews are chosen from the most popular hotels to minimize the risk of mining opinion spam and labeling it as truthful. It is hypothesized that popular establishments are less likely to have deceptive positive reviews since the effect and impact of deceptive reviews would be very small [7, 9].

The deceptive popular reviews are sourced from Mechanical Turk [4]. The reviews are written by American *Turkers* who got the assignment to pretend they were part of the marketing team of a hotel and write a positive review to promote it. Turkers were assigned a hotel and could not spend more than 30 minutes on the review. Furthermore, they could only sumbit one review to ensure diversity in writing. Reviews who where deemed of insufficient quality were rejected (e.g. plagiarized, unreasonably short, or written for the wrong hotel). Finally, 400 written reviews found by Mechanical Turk are combined with the 400 truthful reviews found on TripAdvisor to create the positive part of the corpus.

The second part of the corpus is build from both truthful and deceptive 1 and 2-star reviews from popular online review websites (Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor, and Yelp). While there is no guarantee these reviews don't contain deceptive reviews, recent work suggests that the amount of deceptive reviews on travel review portals are relatively small [11]. The deceptive negative reviews were sourced in the same way as the positive reviews, again using Mechanical Turk. [16].

Although the data is cleaned and kept as consistent as possible, slight differences can occur.

As explained above, the corpus contains a total of 1600 reviews. Table 1 shows an overview on the distribution and origin of the reviews. Each data set specified in Table 1 consists out of 20 reviews for all of the 20 hotels considered. This research will focus on the

[1]https://www.yelp-press.com/company/fast-facts/default.aspx
[2]https://myleott.com/op-spam.html

Authors' address: Ghislaine van den Boogerd (9489479), g.l.vandenboogerd@students.uu.nl; Antonio Oliva Hernández (0685143), a.olivahernandez@students.uu.nl; Otto Mättas (6324363), o.mattas@uu.nl, Utrecht University, Utrecht, Utrecht, The Netherlands.

[3]https://www.tripadvisor.com
[4]http://mturk.com

negative reviews and will try to classify deceptive and truthful reviews accurately.

Number	Sentiment	Source
400	Truthful positive	TripAdvisor
400	Deceptive positive	Mechanical Turk
400	Truthful negative	Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor, Yelp
400	Deceptive negative	Mechanical Turk

Table 1. Distribution and origin of reviews in the corpus.

4 METHODS

Standard n-gram-selection text categorization techniques have been shown to be effective at detecting deceptive opinion reviews [18]. For this paper this method will be combined with other features- i.e. capitalization-ratio, sentiment analysis, verb tenses, proper nouns, and use of singular or plural pronouns- to classify the text.

The text is not pre-processed for the n-gram-selection and sentiment analysis. For the capitalization-ratio, use of singular or plural pronouns, and proper nouns the text is tokenized and lemmatized with NLTK[5]. To extract the verb tenses features, the text is first translated with the googletrans library [6] to Spanish. This is done to circumvent the issue that verb tenses in English are very ambiguous, and to extract verb conjugation, a system a lot more complex would be needed to be built, whereas in Spanish, verb tenses already contain all the information and are mostly unique and distinguishable. This can be shown with the following example, where the verb in the sentence "*they walk*" is the same as in "*I walk*" in English, but in the Spanish equivalent, "*ellos andaron*" is very different from "*yo ando*". After translating, the text is tokenized and lemmatized using the pattern library [7].

All features used in the experiment were chosen with different motivations. For one, the capitalization-ratio, singular or plural pronouns, proper nouns, and verb tenses features are chosen based on a study from 2017 [20]. The paper by Shu et al. (2017) described that deceptive (generated) text often defers from truthful text based on linguistics. Therefor, exploiting linguistics as a feature to find deceptive opinion spam would seem reasonable.

The sentiment features are added because they add information to the review about if its sentiment is classified as either negative, neutral, or positive. Studies show that in fake text, people tend to use less adjectives, which is one of the things the sentiment analyzer looks for in text [10]. This could mean that reviews classified as more neutral could be more likely to be deceptive.

The data is then analyzed with multinomial Naive Bayes (generative linear classifier), Regularized logistic regression (discriminative linear classifier), Classification trees, (flexible classifier) and Random forests (flexible classifier) for both unigram and bigram selection. This means we have a total of 8 classifiers. We use the first four folds (640) reviews for training and hyper-parameter testing and use the last fold (160) to estimate the performance of the classifiers.

[5]https://www.nltk.org
[6]https://pypi.org/project/googletrans/
[7]https://pypi.org/project/Pattern/

4.1 Normalization and tf-idf

To normalize and smooth out the produced n-grams, it was attempted to use the tf-idf statistic applied for each document, and train the algorithms on that data matrix instead of the raw n-grams; however after consulting with the results, the tf-idf data input seemed to create worse predictors than the raw n-grams. As such, the method has been scrapped for the text processing in the final version of the program.

4.2 Bootstrapping

As for our cross validation method we will be using bootstrapping. Whereas cross validation is a re-sampling method without replacement, bootstrapping is. This means the new "surrogate" data sets will contain the same number of cases as the original data set. It does however mean that there can be duplicates, or in some cases that samples don't end up in the surrogate data set at all. However, this added variance aids in polishing any leftover bias towards the testing set at the time of evaluation, giving us a broader data set than the one we were given, and ideally a better idea of the algorithm's performance.

5 RESULTS

The precision, recall, accuracy, negative prediction value, specificity, and f1 score are shown below for all 8 models. Section 6 will analyze and interpret the results.

5.1 Multinomial Naive Bayes - unigram

Table 2 shows the quality metrics of the classifier with unigrams.

Recall	0.910
Precision	0.898
Negative prediction value	0.909
Accuracy	0.904
Specificity	0.897
f1 score	0.903

Table 2. Quality measures multinomial Naive Bayes unigram

5.2 Multinomial naive Bayes - unibigram

Table 3 shows the quality metrics of the classifier with unibigrams.

Recall	0.784
Precision	0.924
Negative prediction value	0.812
Accuracy	0.859
Specificity	0.936
f1 score	0.848

Table 3. Quality measures multinomial Naive Bayes bigram

5.3 Regularized logistic regression - unigram

Table 4 shows the quality metrics of the classifier with unigrams.

Recall	0.899
Precision	0.806
Negative prediction value	0.886
Accuracy	0.841
Specificity	0.784
f1 score	0.848

Table 4. Quality measures Regularized logistic regression unigram

5.4 Regularized logistic regression - unibigram

Table 5 shows the quality metrics of the classifier with unibigrams.

Recall	0.898
Precision	0.827
Negative prediction value	0.888
Accuracy	0.855
Specificity	0.813
f1 score	0.860

Table 5. Quality measures Regularized logistic regression bigram

5.5 Classification trees - unigram

Table 6 shows the quality metrics of the classifier with unigrams.

Recall	0.720
Precision	0.697
Negative prediction value	0.711
Accuracy	0.704
Specificity	0.688
f1 score	0.707

Table 6. Quality measures Classification trees unigram

5.6 Classification trees - unibigram

Table 7 shows the quality metrics of the classifier with unibigrams.

Recall	0.619
Precision	0.712
Negative prediction value	0.660
Accuracy	0.682
Specificity	0.753
f1 score	0.658

Table 7. Quality measures Classification trees bigram

5.7 Random forests - unigram

Table 8 shows the quality metrics of the classifier with unigrams.

Recall	0.874
Precision	0.875
Negative prediction value	0.875
Accuracy	0.875
Specificity	0.875
f1 score	0.874

Table 8. Quality measures Random forests unigram

5.8 Random forests - unibigram

Table 9 shows the quality metrics of the classifier with unibigrams.

Recall	0.911
Precision	0.805
Negative prediction value	0.897
Accuracy	0.845
Specificity	0.779
f1 score	0.854

Table 9. Quality measures Random forests bigram

5.9 Features to detect deceptive reviews

The models for all classifiers were analyzed to find the the top-5 most used features to detect deceptive reviews. The top-5 is selected by reducing impurity on the nodes down the tree. The features that were deemed most important were:

- The frequency of the word: spend
- The frequency of the word: accepting
- The frequency of the word: handed
- The frequency of the word: initials
- The frequency of the word: thoroughly

5.10 Features to detect truthful reviews

The models for all classifiers were analyzed to find the the top-5 most used features to detect truthful reviews. The top-5 is selected by reducing impurity on the nodes down the tree. The features that were deemed most important were:

- The frequency of the word: spend
- The frequency of the word: accepting
- The frequency of the word: handed
- The frequency of the word: initials
- The frequency of the word: thoroughly

Additionally, the seventh most informative feature was the capitalisation of starting sentences, indicating that the custom features chosen by the team made an impact on the results.

Source	Degrees of Freedom	Sum of Square	Mean Square	F Statistic	P-value
Groups (between groups)	3	3.7391	1.2464	1394.2091	0.000
Error (within groups)	636	0.5686	0.0008940		
Total	639	4.3076	0.006741		

Table 10. Unigram ANOVA results

Source	Degrees of Freedom	Sum of Square	Mean Square	F Statistic	P-value
Groups (between groups)	3	3.5079	1.1693	1251.0014	$-4.441e^{-16}$
Error (within groups)	636	0.5945	0.0009347		
Total	639	4.1024	0.006420		

Table 11. Unibigram ANOVA results

Source	Degrees of Freedom	Sum of Square	Mean Square	F Statistic	P-value
Groups (between groups)	1	0.1529	0.1529	242.2022	$2.220e^{-16}$
Error (within groups)	318	0.2007	0.0006312		
Total	319	0.3536	0.001108		

Table 12. Multinomial Naive Bayes ANOVA results

Source	Degrees of Freedom	Sum of Square	Mean Square	F Statistic	P-value
Groups (between groups)	1	0.01564	0.1564	18.8094	0.00001942
Error (within groups)	318	0.2643	0.0008312		
Total	319	0.2800	0.0008777		

Table 13. Regularized logistic regression ANOVA results

Source	Degrees of Freedom	Sum of Square	Mean Square	F Statistic	P-value
Groups (between groups)	1	0.03614	0.03614	26.1862	$5.374e^{-7}$
Error (within groups)	318	0.4389	0.001380		
Total	319	0.4750	0.001489		

Table 14. Classification trees ANOVA results

Source	Degrees of Freedom	Sum of Square	Mean Square	F Statistic	P-value
Groups (between groups)	1	0.06845	0.06845	84.0051	$8.882e^{-16}$
Error (within groups)	318	0.2591	0.0008148		
Total	319	0.3276	0.001027		

Table 15. Random forests ANOVA results

6 ANALYSIS

To better understand the differences between the eight training methods, one could use statistical tests to measure a method's performance over larger populations. Here, a population refers to a set of cross-validated evaluation runs carried out using one method. That said, eight populations with a sample size of $N = 160$ each are compared, one for every training method.

Each training run results in performance metrics such as recall, precision and accuracy among others, while a specific metric to be assessed was given by the assignment. In this case, researchers are investigating accuracy which is a common metric for evaluating a model with an even class distribution.

Important to note is that if the model's class distribution is uneven and it is especially important to investigate false positives for example, other metrics might be better suited for using in the statistical tests. To stay in the scope of the assignment, the team is going ahead with accuracy as the key metric.

Another important factor about deciding on statistical test is that for each training method, there is randomness which is not considered. This is for the work to fit the scope of this assignment. While developing the technical implementation, personal experience dictates that the randomness is not skewing any training method from the results presented in 5 to the extend where it be rendered random.

Pair	Difference	Standard Error	Q	Lower Confidence Interval	Upper Confidence Interval	Critical Mean	P-value
Unigram results							
x_1-x_2	0.06254	0.002364	26.4578	0.05393	0.07115	0.008611	9.165e-11
x_1-x_3	0.1995	0.002364	84.3812	0.1908	0.2081	0.008611	9.165e-11
x_1-x_4	0.02895	0.002364	12.2476	0.02034	0.03756	0.008611	9.169e-11
x_2-x_3	0.1369	0.002364	57.9234	0.1283	0.1455	0.008611	9.165e-11
x_2-x_4	0.03359	0.002364	14.2102	0.02498	0.04220	0.008611	9.169e-11
x_3-x_4	0.1705	0.002364	72.1336	0.1619	0.1791	0.008611	9.165e-11
Unibigram results							
x_5-x_6	0.004846	0.002417	2.0050	-0.003959	0.01365	0.008805	0.4887
x_5-x_7	0.1770	0.002417	73.2297	0.1682	0.1858	0.008805	9.165e-11
x_5-x_8	0.01449	0.002417	5.9942	0.005683	0.02329	0.008805	0.0001514
x_6-x_7	0.1721	0.002417	71.2247	0.1633	0.1810	0.008805	9.165e-11
x_6-x_8	0.009642	0.002417	3.9893	0.0008375	0.01845	0.008805	0.02539
x_7-x_8	0.1625	0.002417	67.2354	0.1537	0.1713	0.008805	9.165e-11
Method results							
x_1-x_5	0.04371	0.001986	22.0092	0.03819	0.04924	0.005526	1.311e-10
x_2-x_6	0.01398	0.002279	6.1334	0.007638	0.02032	0.006342	0.00001942
x_3-x_7	0.02125	0.002937	7.2369	0.01308	0.02943	0.008172	5.376e-7
x_4-x_8	0.02925	0.002257	12.9619	0.02297	0.03553	0.006279	1.311e-10

Table 16. Tukey test results where the HSD statistic Q indicate statistical significance in difference between all pairwise comparisons.

6.1 ANOVA test

Researchers are tasked with comparing eight distinct independent variables simultaneously. Such multivariate problems require something other than a t-test owing to the sheer number of independently varying relationships.

ANOVA stands for "analysis of variance" and addresses precisely the problem just described. It accounts for the rapidly expanding degrees of freedom in a sample as variables are added.

The null hypothesis for the test is that the two means are equal. Therefore, a significant result means that the two means are unequal. The team sees this as a valid statistical test to conduct for the purposes of assessing differences in accuracy between different methods.

First, in order to have samples to compare, the researchers have created populations for each method with a sample size $N = 160$. Each method was simply evaluated 160 times and the results were collected. After pre-processing the data to separate accuracy values for each run, they were used as input for the ANOVA test. A common significance level $\alpha = 0.05$ is used. The results can be found in Tables 10 to 15.

Looking at Tables 10 to 15 where the $P - value$ is lower than α, this indicates statistical significance. More specifically, the null hypothesis (H_0) is rejected, some of the groups' averages are considered to be not equal. In other words, the difference between the averages of some groups is big enough to be statistically significant.

The ANOVA test indicates that there is a need to carry out a Tukey HSD test in order to gain knowledge about which method differs from the other.

6.2 ANOVA test limitation

A one way ANOVA test will tell if at least two groups were different from each other. But it won't tell which groups were different. For this purpose Tukey's Honest Significant Difference (Tukey's HSD) / Kramer test was chosen[1].

To run the test, sixteen distinct pairs were created to investigate. For method comparisons using unigrams: $\{x_1; x_2\}$, $\{x_1; x_3\}$, $\{x_1; x_4\}$, $\{x_2; x_3\}$, $\{x_2; x_4\}$, $\{x_3, x_4\}$. For method comparisons using unibigrams: $\{x_5; x_6\}$, $\{x_5; x_7\}$, $\{x_5; x_8\}$, $\{x_6; x_7\}$, $\{x_6; x_8\}$, $\{x_7, x_8\}$. For comparisons within methods: $\{x_1; x_5\}$, $\{x_2; x_6\}$, $\{x_3; x_7\}$, $\{x_4; x_8\}$. The mappings are given as follows:

- x_1 = Multinomial Naive Bayes - unigram,
- x_2 = Regularized logistic regression - unigram,
- x_3 = Classification trees - unigram,
- x_4 = Random forests - unigram,
- x_5 = Multinomial Naive Bayes - unibigram,
- x_6 = Regularized logistic regression - unibigram,
- x_7 = Classification trees - unibigram,
- x_8 = Random forests - unibigram.

The results from the Tukey test results can be found in Table 16.

7 DISCUSSION

7.1 Linear Models

This paper used a generative linear model classifier in the form of multinomial Naive Bayes. Compared to the discriminative linear model in the form of the regularized logistic regression the generative linear model performed statistically significantly better. This could be explained by the fact that discriminative models make a hard split over the entire space, whereas generative models are

much more flexible because they try to fit a cluster. This could be an explanation why for this specific problem the discriminative models worked less.

7.2 Random Forests

A random forest could improve on the linear classifiers where the use case allows for it. There might be flexibility in the feature selection and classification what is not possible with linear models. Namely, clusters that sit in the separate space of the planes can be fitted.

Also, this is apparent looking at the metrics in 5.7 and 5.8 which outperform most linear classifier metrcis accordingly.

7.3 N-Gram Features

During the research classifiers were differentiated between themselves by testing a classifier that only uses unigram features and a classifier that uses both unigram and bigram features. It turned out that adding bigram features actually lowered the accuracy compared to the same model using only unigram features. This could be explained because of the short length of the reviews. Because of the length of the reviews, it makes it difficult to gain any useful information by linking words together.

However, the researchers do want to note that in some cases linking the words could be a necessity to classifying the texts properly. For example, the case "super bad" would result in mixed classification while using unigrams, whereas with bigrams the case would have a completely different effect.

7.4 Important Features

The features that were deemed the most important for classifying the reviews were the same for detection of truthful reviews as for the detection of deceptive reviews. This hints that the problem on hand is a two-class classification problem. Classifying with a certain feature impacts both the detection of truthful and deceptive reviews. A multi-class classification problem would probably yield different important features.

7.5 Overfitting

Looking at the outcome, one might wonder why do the results between used methods differ?

Namely, the accuracy of the single tree classification method performs significantly lower than the other methods. The researchers believe this can be attributed to overfitting as the method uses only one discreet way of splitting the tree into the training data. The results do not translate well to the real world (nor the testing data) as the model would be coupled with the training data. If the input stays the same, the tree and its splits will stay the same. If there are new trends in the real world, these can not be predicted by using the model either.

This informs the researchers that generating a classification model with more complex methods would make sense. Most importantly, some randomness in the training phase can contribute a lot in the final performance of the model. Loosening the grip on training data has benefits in real-world scenarios with even unforeseen cases.

7.6 Optimisation

Currently the implementation is noticeably very slow. This is due to several factors, with the search for hyperparameters and the verb analysis taking up most of the processing time. For the verb analysis, the main time issue is the fact that the translator being used, Google Translator, bans IP addresses on multiple successive requests. This measure is very common among online translators, and some even require an API key to retrieve translations. Thus to prevent this, a delay between translations was implemented, to the detriment of the speed of processing of the translations. A different translation method could have been used, or even more translators in parallel. However, it was decided it fell out of the scope of this assignment, although it would definitely be considered for a more developed version of the program.

As for the hyper-parameter search, the program trains and tests thousands of models, with very varying sizes. This was a design choice done to try to hone in more finely to better hyper-parameters, but as it stands, it costs quite a lot of time to train and test them. For a more developed version, a deeper look into optimizing the hyper-parameter ranges would greatly speed up the processing time.

Furthermore, the extra text meta-features were not found to be statistically significantly better, and they take a reasonable amount to compute. While the code producing them could most probably be further optimized, the features maybe should be substituted by other different meta-features or removed entirely.

7.7 Limitations

For the scope of this project, the researchers kept to analyzing accuracy as the key performance metric. There is still plenty left to learn about the models produced by the different methods.

When data is not balanced or the relative risk of having false positives/false negatives is important, the need for special metrics arises. Precision and recall are two popular choices used widely in different classification tasks. These, of course, are not the only methods used for evaluating the performance of a classifier. Other metrics like F1 score, specificity, and ROC AUC also enjoy widespread use.

Again, for the sake of adhering to this assignment's scope and focusing on learning the difference between different training methods, these extra steps are omitted.

Also, one model is trained for each training method. To have a complete statistical overview of the comparisons, a large sample size should be considered. Namely, at least a commonly accepted sample size of 30 models per method should be compared. For this assignment, the team focused on improving each method through intuition and analysing each method in relation to other methods pragmatically.

7.8 Conclusion

Overall, the multinomial Naive Bayes classifier with unigrams seems to work the best on this particular corpus, with an accuracy of 90%. The statistical test also show that the accuracy for this classifier is statistically significantly better compared to the other classifiers.

Interesting to note is that the same classifier with unibigrams does not seem to work better compared to other classifiers. Even

though the accuracy is still considered to be one of the higher scores (86%), statistical tests do not show a significant difference to for example the unibigram logistic regression classifier (accuracy = 86%). This is an interesting finding because the logistic regression classification method is much less computational heavy compared to the multinomial Naive Bayes classifier. So looking at computational power, this can be an interesting finding to be considered during future text classification problems.

For future work, there are multiple avenues still left unexplored. Following one's intuition might provide a relevant direction as proven by the researcher's work. A feature, namely "capitalisation" showed up as one of the most informative features towards truthful reviews and would have not been discovered without the customised approach devised by the team.

Additionally, there is plenty of information to be extracted from diving into informative features and trying to distinguish further between features indicating deceptive and truthful reviews.

8 ACKNOWLEDGMENTS

This research project work was supported by Dr. A.J. (Ad) Feelders, an assistant professor at Utrecht University at the Science department. He is also the lecturer for the Data Mining course in the scope of which this project has been carried out.

REFERENCES

[1] Herve Abdi and Lynne Williams. 2021. Tukey's Honestly Signiflcant Difierence (HSD) Test. (10 2021).
[2] Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. 2009. How Opinions are Received by Online Communities: A Case Study on Amazon.Com Helpfulness Votes. *Proceedings of the 18th International Conference on World Wide Web, WWW '09* (06 2009). https://doi.org/10.1145/1526709.1526729
[3] Harris Drucker, Donghui Wu, and V.N. Vapnik. 1999. Support Vector Machines for Spam Categorization. *Neural Networks, IEEE Transactions on* 10 (10 1999), 1048 – 1054. https://doi.org/10.1109/72.788645
[4] Shyam Gopinath, Jacquelyn S Thomas, and Lakshman Krishnamurthi. 2014. Investigating the relationship between the content of online word of mouth, advertising, and brand performance. *Marketing Science* 33, 2 (2014), 241–258.
[5] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. 2004. Combating Web Spam with TrustRank. *Intl Conference on Very Large Data Bases*, 576–587. https://doi.org/10.1016/B978-012088469-8/50052-8
[6] Nitin Jindal and Bing Liu. 2008. Opinion Spam and Analysis. https://doi.org/10.1145/1341531.1341560
[7] Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 international conference on web search and data mining*. 219–230.
[8] Soo-Min Kim, Patrick Pantel, Timothy Chklovski, and Marco Pennacchiotti. 2006. Automatically Assessing Review Helpfulness. *EMNLP*, 423–430. https://doi.org/10.3115/1610075.1610135
[9] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. 939–948.
[10] David M Markowitz and Jeffrey T Hancock. 2014. Linguistic traces of a scientific fraud: The case of Diederik Stapel. *PloS one* 9, 8 (2014), e105937.
[11] Dina Mayzlin and Hema Yoganarasimhan. 2012. Link to success: How blogs build an audience by promoting rivals. *Management Science* 58, 9 (2012), 1651–1668.
[12] Rada Mihalcea and Carlo Strapparava. 2009. The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language. 309–312.
[13] Matthew Newman, James Pennebaker, Diane Berry, and Jane Richards. 2003. Lying Words: Predicting Deception from Linguistic Styles. *Personality social psychology bulletin* 29 (06 2003), 665–75. https://doi.org/10.1177/0146167203029005010
[14] Ntoulas, Alexandros, Najork, Marc, Manasse Mark, Mark, Fetterly, and Dennis. 2006. Detecting Spam Web Pages through Content Analysis. https://doi.org/10.1145/1135777.1135794
[15] M. Ott, Claire Cardie, and Jeffrey Hancock. 2013. Negative Deceptive Opinion Spam. (01 2013), 497–501.
[16] Myle Ott, Claire Cardie, and Jeffrey T Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*. 497–501.
[17] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey Hancock. 2011. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. (07 2011).
[18] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557* (2011).
[19] Michael O'Mahony and Barry Smyth. 2009. Learning to recommend helpful hotel reviews. *RecSys'09 - Proceedings of the 3rd ACM Conference on Recommender Systems*, 305–308. https://doi.org/10.1145/1639714.1639774
[20] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.
[21] Wouter Weerkamp and Maarten Rijke. 2008. Credibility Improves Topical Blog Post Retrieval. 923–931.
[22] Markus Weimer, Iryna Gurevych, and Max Mühlhäuser. 2007. Automatically Assessing the Post Quality in Online Discussions on Software. https://doi.org/10.3115/1557769.1557806
[23] Guangyu Wu, Derek Greene, Barry Smyth, and Padraig Cunningham. 2010. Distortion as a Validation Criterion in the Identification of Suspicious Reviews. https://doi.org/10.1145/1964858.1964860
[24] Kyung-Hyan Yoo and Ulrike Gretzel. 2009. Comparison of Deceptive and Truthful Travel Reviews. 37–47. https://doi.org/10.1007/978-3-211-93971-0_4
[25] Lina Zhou, Judee Burgoon, Douglas Twitchell, Tiantian Qin, and Jay Jr. 2004. A Comparison of Classification Methods for Predicting Deception in Computer-Mediated Communication. *J. of Management Information Systems* 20 (03 2004), 139–165. https://doi.org/10.1080/07421222.2004.11045779
[26] Lina Zhou, Yongmei Shi, and Dongsong Zhang. 2008. A Statistical Language Modeling Approach to Online Deception Detection. *Knowledge and Data Engineering, IEEE Transactions on* 20 (09 2008), 1077 – 1081. https://doi.org/10.1109/TKDE.2007.190624