Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Reducing Feature Set Explosion to Facilitate Real-World Review Spam Detection

Michael Crawford and **Taghi M. Khoshgoftaar** and **Joseph D. Prusa**
michaelcrawf2014@fau.edu, taghi@cse.fau.edu, jprusa@fau.edu

Abstract

Online reviews are quickly becoming one of the most important sources of information for consumers on various products and services. With their increased importance, there exists an increased opportunity for spammers or unethical business owners to create false reviews in order to artificially promote their goods and services or smear those of their competitors. In response to this growing problem, there have been many studies on the most effective ways of detecting review spam using various machine learning algorithms. One common thread in most of these studies is the conversion of reviews to word vectors, which can potentially result in hundreds of thousands of features. However, there has been little study on reducing the feature subset size to a manageable number or how best to do so. In this paper, we consider two distinct methods of reducing feature subset size in the review spam domain. The methods include filter-based feature rankers and word-frequency based feature selection. We show that there is not a one size fits all approach to feature selection, and the best way to reduce the feature subset size is dependent upon both the classifier being used and the feature subset size desired. It was also observed that the feature subset size had significant influence on which feature selection method is used.

Introduction

Consumers have always sought to find reviews or recommendations for various products and services before they buy them. Previously, consumers relied on publications and services such as Consumer Reports or AAA; however, the explosion of Web 2.0 has provided a way for consumers to share their experiences directly with each other, and not rely on third party companies for this type of information. Sites like Yelp, TripAdvisor, and Amazon provide a way for consumers to give feedback on their experience with various products or services, which others can then view before deciding to make a purchase.

While this has worked well for many years, the influence of reviews on product and service consumption is loosing its authenticity. Spammers and unscrupulous businesses have tampered the review sites with with fake and untruthful reviews. The Canadian government recently estimated that a full third of all online reviews are fake, which prompted them to issue a warning "encouraging consumers to be wary of fake online endorsements that give the impression that they have been made by ordinary consumers." (Bureau 2014)

Methods for detecting fake reviews have become forefront in recent years since these types of untruthful reviews can erode consumer trust or negatively affect their purchasing habits. Most research on detecting review spam involves training classifiers using a labeled dataset and then applying it to unlabeled reviews reviews to determine if they are fake or not (Crawford et al. 2015)(Ott, Cardie, and Hancock 2013). Researchers commonly use the occurrence of words in reviews as the features which describe a given review instance. However, in an area which could potentially have millions of reviews and hundreds of thousands of distinct words, it is important to identify which words (or features) are important or even potentially detrimental as traditional machine learning techniques have shown difficulty in scaling to feature set sizes of this magnitude (López et al. 2015). Existing research has, for the most part, focused on the creation of datasets, features, and evaluation of classifiers (Shojaee et al. 2013)(Ott et al. 2011)(Fei et al. 2013). How many of the word features are needed to effectively train a review spam classifier has generally been ignored. To the best of our knowledge this is the first study on the number of text based features which are needed for detection of review spam.

This study provides an empirical evaluation of the effectiveness of two methods of feature selection for reducing the feature set size in review spam detection. We compare feature selection using word frequency versus a Chi-Squared feature selection method across various classifiers. The dataset used in this study spans three distinct domains: restaurants, hotels, and doctors. We found that for lower feature subset sizes, the Chi-Squared feature selection technique outperformed using word frequency for all of the classifiers in this study. At higher feature subset sizes, there was no distinguishable difference between the two methods.

The remainder of this paper is structured as follows. The Related Works section contains some of the previous research in the review spam domain. The Empirical Design section discusses the dataset used in this study along with the experimental protocol which was followed. The Results section presents the results of our study along with commen-

tary and statistical analysis of them. The Conclusion section contains our conclusions along with possible areas of future study.

Related Works

As the field of online review spam detection is relatively new, the first study that we are able to find on the subject matter was done by Jindal and Liu in 2007 (Jindal and Liu 2007), which they later expanded upon in 2008 (Jindal and Liu 2008). In their study, they use a dataset of nearly six million reviews collected from Amazon. All reviews which are duplicates or near duplicates are labeled as spam. Reviews with a Jaccard similarity score of over 90% were considered duplicates. A method known as w-shingling (Broder et al. 1997) was used in order to accomplish this task on such a large dataset . They trained a Logistic Regression classifier on this data and use it to try to identify potential spam which are not simply duplicate reviews. Upon manually analysis of 100 of the non-duplicate reviews which had been classified as spam, 52 of them were definitely spam.

Finding labeled datasets is always a challenge for machine learning researchers and review spam detection is no different. Ott et al. (Ott et al. 2011) devised a novel method of using Amazon Mechanical Turk (AMT) to generate fake reviews for their dataset and combined them with "truthful" reviews which were collected from TripAdvisor. In all, 400 deceptive and 400 truthful reviews were collected to construct their final dataset. They evaluated the performance of Naïve Bayes and Support Vector Machine (SVM) classifiers using unigrams, bigrams and trigrams. The authors observed that SVM with bigrams had the highest performance; however, no statistical analysis was done to determine if this difference was significant and the dataset was relatively small.

In a recent study by Li et al. (Li et al. 2014), the dataset previously created by Ott et al. was expanded to include two additional domains (restaurants and doctors). While AMT was once again used to collect some of the fake reviews, a study by Mukherjee et al. (Mukherjee et al. 2013) had found that reviews created via AMT were more easily identified than reviews which had been flagged as spam on the Yelp website. To combat this, Li et al. also solicited fake reviews from actual employees and thus dubbed them "expert" reviews since these are people with direct knowledge of the locations they are reviewing and can potentially provide more "accurate" fake reviews. The authors then studied the effectiveness of training and evaluating an SVM classifier on each of the domains. They observed that unigram features demonstrate the best performance; however, it is not clear how many features were used or how they were selected. An important insight from Li et al.'s study was that it was easier to distinguish between truthful and AMT reviews than truthful and expert reviews.

Empirical Design

Dataset

In our study, we use the publicly available version of a dataset from (Li et al. 2014), which features reviews from three domains (restaurants, hotels, and doctors). The truthful reviews in the dataset were collected from actual review websites, while the fake reviews were solicited from AMT and industry experts. The breakdown of the class and domain distribution of the dataset is detailed in Table 1. The instances themselves are simply the text of the review and class indicator (spam or truthful). The text was later split into individual words (unigrams), as described in the next section, which served as the features of the instances.

Domain	Truthful	Spam	Total
doctors	200	356	356
hotels	800	1080	1880
restaurants	200	200	400
Total	1200	1636	2836

Table 1: Dataset domain and class distribution

Classifiers, Cross-Validation, and Performance Metric

We consider five different classifiers, Decision Tree (C4.5), Logistic Regression (LR), Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), and Support Vector Machine (SVM) in our study. All classifiers were implemented using the Weka machine learning toolkit (Hall et al. 2009) using the default values except SVM, for which the complexity constant (c) was set to 5 and the $buildLogisticModels$ parameter was set to $true$. Please refer to (Hall et al. 2009) for further information on the specific implementation of these classifiers.

Four runs of five-fold cross validation were used in all experiments. This means that for each run, the dataset was randomly divided into five parts. Each of the five parts was then used exactly one time to evaluate a classifier which was trained on the remaining four parts. All classifiers were trained such that feature creation and selection was performed using the four training folds of cross-validation. This process was then repeated four times, which results in training and evaluating twenty models for each combination of classifier and feature subset size. By doing the experimentation in this manner, we reduce the chance of any bias due to good or bad data splits.

The performance of each model is evaluated using the Area Under the Receiver Operating Characteristic Curve (AUC). The curve is a graph with the false positive rate on the *X* axis versus the true positive rate on the *Y* axis. These rates are inversely correlated and collected by varying the threshold value which splits positive and negative class predictions. By determining the area under this curve, one can effectively estimate the true predictive capability of a given model.

Feature Selection Techniques

In addition to word frequency, ten filter-based feature selection techniques were also considered for this study: Signal-to-Noise (S2N), Chi-Squared (CS), Mutual Information (MI), area under the Precision-Recall Curve (PRC), AUC, Wilcoxon Rank Sum (WRS), Probability Ratio (PR) and

Technique	Group	AUC	stdev
S2N	A	0.822	0.061
CS	A	0.821	0.062
AUC	B	0.818	0.054
KS	B	0.818	0.054
MI	B	0.817	0.054
PRC	B	0.816	0.054
SAM	C	0.661	0.060
WRS	D	0.614	0.050
GI	E	0.605	0.072
PR	E	0.604	0.070

Table 2: Tukey HSD Test of feature selection techniques across all classifiers with feature subset sizes from 100 to 1,000 (alpha=0.05)

Gini-Index (GI). Filter-based techniques were used in favor of wrapper based techniques since the filter-based techniques are relatively less computationally complex (Prusa, Khoshgoftaar, and Dittman 2015). An experiment was conducted to evaluate each of these ten filter-based feature selection techniques against the combined dataset from Table 1 with various classifiers. The number of features to be selected was varied from 100 to 1,000 in increments of 100 for each combination of classifier and feature selection technique. A Tukey's HSD (Honest Significant Difference) test was run on the results (Table 2) and shows that S2N and CS are in the top group (A), while 4 others are in a second group (B), and the last 4 are far below in the other groups (C,D,E). The results are also summarized visually in Figure 1. In this figure you can clearly see that there is a stark contrast between the top six (S2N, CS, MI, PRC, AUC and KS) and the bottom four (SAM, WRS, GI and PR). Because there isn't much of a difference in the top performers (and no statistically difference between the top two) we chose to use CS for this study as it is the most commonly available and widely used in the literature for other text mining domains.

In the case of attribute selection using word frequency, we use a modified version of the StringToWordVector filter from the Weka toolkit. The out-of-the-box filter allows you specify a minimum number of attributes but then includes all ties. So if you specify 100 words, it will determine what the cut-off frequency is for 100 words and then include all words which have at least the number of occurrences as this cutoff value. This means that while you specify 100 words, you may in fact get a higher number of attributes (words). Our modified implementation, ExactStringToWordVector, instead randomly selects a subset of the words which have the same number of occurrences such that the exact number of features asked for is returned. So if you ask for 100 words, you will get exactly 100 words, presuming 100 distinct words exist.

Results

In this study we attempt to empirically determine how many features are needed for several types of classifiers in the review spam domain, as well as determine if it is better to use a Chi-Squared feature selection filter or simply rely upon the word count frequencies in the documents themselves to select the features. An analysis of variance (ANOVA) test was run on the results (Table 3), and it shows that all three factors (classifier, feature selection technique, and subset size), and their interactions, are significant.

Additional experiments were conducted to analyze each of the classification techniques across a larger range of subset sizes, specifically looking at the word frequency and Chi-Squared feature selection techniques. In the remainder of this section, we will analyze each of the classification techniques individually, look at them as a whole, and discuss our recommendations. For C4.5 and Naïve Bayes we show feature subset sizes from 100 to 5,000 as the performance does not change beyond that. The remaining classifiers (Logistic Regression, SVM and Multinomial Naïve Bayes) we evaluate feature subset sizes from 100 to 10,000 as they continued

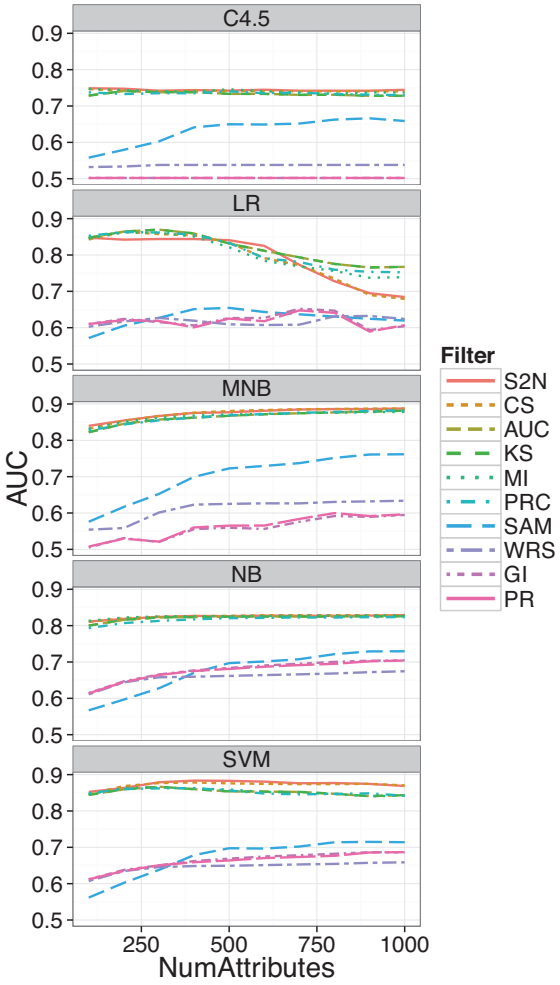

Figure 1: Comparison of feature selection techniques across classifiers. While difficult to distinguish between the individual classifiers, it is easy to see there generally 2 observable groups. (The legend is in ordered of mean AUC from top to bottom)

	Df	SS	MS	F-value	Pr(>F)
Classifier(A)	4	24.51	6.13	9835.3	0.0000
Num Attr(B)	44	1.84	0.04	67.0	0.0000
Technique(C)	1	1.33	1.33	2139.7	0.0000
A:B	156	4.49	0.03	46.2	0.0000
A:C	4	1.55	0.39	622.0	0.0000
B:C	34	0.64	0.02	30.1	0.0000
A:B:C	136	3.82	0.03	45.1	0.0000
Residuals	7220	4.50	0.00		

Table 3: ANOVA analysis of the number of attributes, classifier, and feature selection technique

to exhibit variation at higher subset sizes.

Decision Tree

With the C4.5 Decision Tree implementation, the Chi-Squared feature selector has a higher AUC score than frequency for all subset sizes (Fig 2). While using the Chi-Squared feature selection method achieves near its top AUC score for all subset sizes, frequency has a sharp increase in AUC from 100 to 1,000 and then levels off for larger subset sizes. This would indicate that CS should always be used in favor of frequency for this classifier and domain.

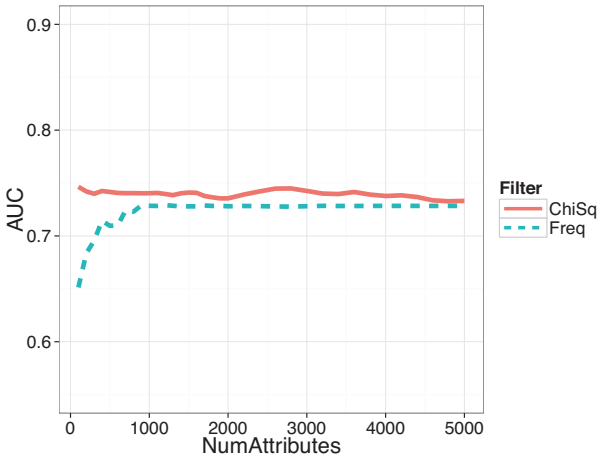

Figure 2: Average AUC score for C4.5 using feature subset sizes from 100 to 5,000

Logistic Regression

With Logistic Regression, we see an interesting trend that there is a spike early on and then a steep decline (Fig 3). Both the Chi-Squared and frequency versions then recover at higher attribute levels. While using frequency is good at very high numbers of attributes, having a feature set of this size is counter-productive to our goals. Also of note is that Chi-Squared does not have as big of a drop in performance and recovers more quickly from the downturn. In general, the LR classifier appears to be very unstable with respect to feature subset set size on this dataset. Ensemble learners using LR may stabilize the classification performance, and can be investigated in future work.

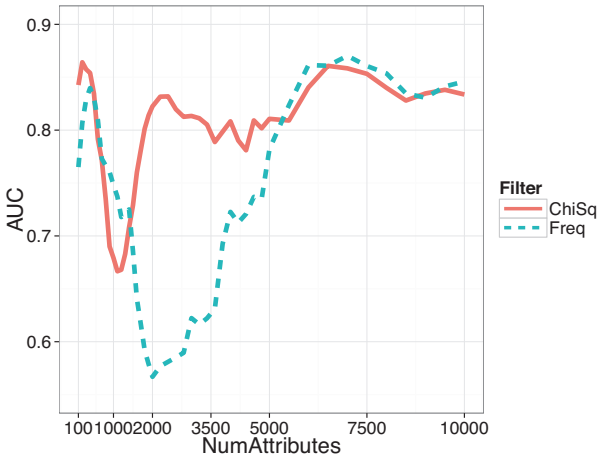

Figure 3: Average AUC score for Logistic Regression using feature subset sizes from 100 to 10,000

Naïve Bayes

Examining Naïve Bayes we can observe that at lower feature subset sizes (i.e. less than 1000), the Chi-Squared feature selection technique has a higher AUC than frequency (Fig 4); however, frequency yields slightly higher beyond 2,500. This would again suggest that the choice in feature selection technique would be dictated by the feature subset size.

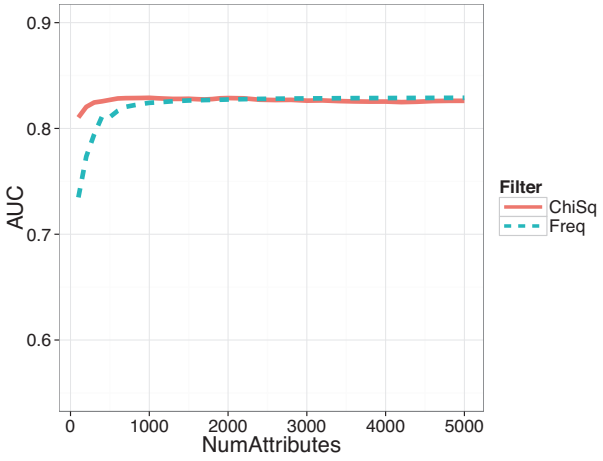

Figure 4: Average AUC score for Naïve Bayes using feature subset sizes from 100 to 10,000

Support Vector Machine

The results for Support Vector Machine (Fig 5) have some of the most stark contrasts with Chi-Squared performing better at lower subset sizes, but frequency having a higher mean AUC beyond 1,700 features. Although the average AUC for frequency shows a steady increase with the addition of additional features, it never reaches the peak obtained by the Chi-Squared feature selection technique at 400 features. This would suggest that Chi-Squared feature selection should be

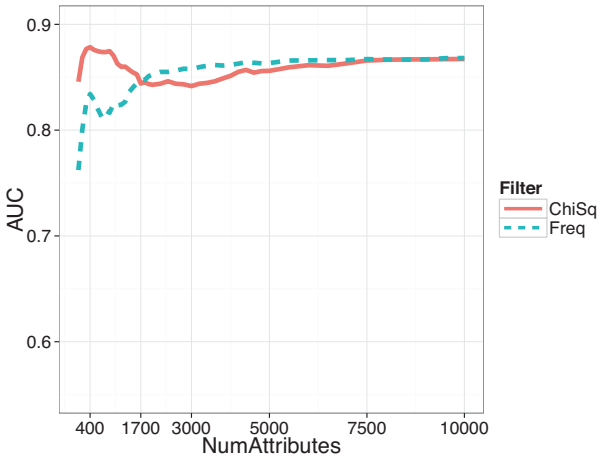

Figure 5: Average AUC score for Support Vector Machine using feature subset sizes from 100 to 10,000

used when using the SVM classifier and smaller feature subset sizes are desired.

Multinomial Naïve Bayes

Multinomial Naïve Bayes is similar to Naïve Bayes in shape except that the AUC scores are higher (Fig 6). The Chi-Squared feature selector has a higher base AUC at 100 features and quickly rises to a local minimum around 800 features then levels off. Conversely, selecting features based upon frequency has much lower performance with 100 features and a steady rise as more features are added. Of note is that the mean AUC using frequency is actually higher than Chi-Squared for higher feature subset sizes. This would indicate that for this particular classifier, the number of features that you want to use in your final model should dictate which method to use to pick them.

As Multinomial Naïve Bayes provides the best performance for this particular dataset (Table 6), we investigate further the differences between the feature selection techniques at different subset sizes while providing statistical analysis.

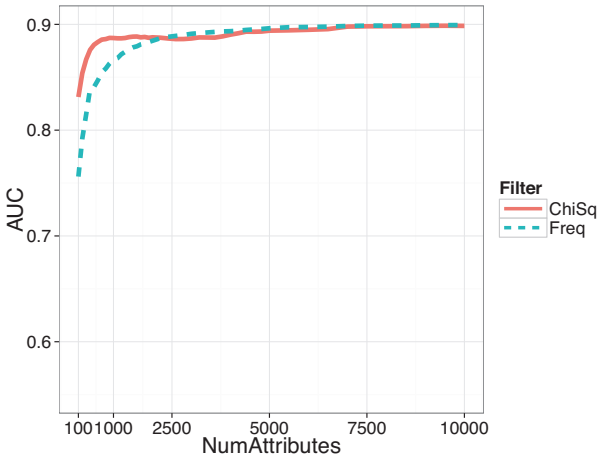

Figure 6: Average AUC score for Multinomial Naïve Bayes using feature subset sizes from 100 to 10,000

Technique:Subset Size	Group	AUC	stdev
ChiSq:1000	A	0.887	0.017
ChiSq:500	AB	0.881	0.018
Freq:1000	B	0.867	0.018
Freq:500	C	0.841	0.022

Table 4: Tukey HSD Test comparing feature subset size and selection technique for subset sizes of 500 and 1,000

Technique:Subset Size	Group	AUC	stdev
Freq:5000	A	0.896	0.014
ChiSq:5000	A	0.894	0.015
Freq:4000	A	0.894	0.015
Freq:3000	A	0.891	0.016
ChiSq:4000	A	0.890	0.015
ChiSq:2000	A	0.888	0.016
ChiSq:3000	A	0.887	0.016
Freq:2000	A	0.884	0.017

Table 5: Tukey HSD Test comparing feature subset size and selection technique for subset sizes of 2,000 to 5,000

Classifier	Group	AUC	stdev
MNB	A	0.885	0.017
SVM	B	0.874	0.013
NB	C	0.828	0.024
LR	D	0.750	0.060
C4.5	E	0.741	0.022

Table 6: Tukey HSD Test of each classifier using Chi-Squared feature selection and feature subset sizes between 500 and 1000 (alpha=0.05)

Looking at relatively small subset sizes of 500 and 1,000, it can be observed that Chi-Squared performs significantly better (Table 4). However, when looking at subset sizes from 2,000 to 5,000 (Table 5), there is not a significant difference between the two. This tells us that for smaller subset sizes, Chi-Squared performs better; however, if you chose a larger subset size there is no gain from the additional cost of using Chi-Squared since simple frequency performs equally well.

Comparison of Classifiers

In order to compare the classifiers, a Tukey HSD test was performed comparing the AUC scores of each classifier when using Chi-Squared feature selection (Table 6). Each classifier is found to be significantly different than the others, with Multinomial Naïve Bayes and SVM being the best while Logistic Regression and C4.5 are significantly worse than the others. Standard Naïve Bayes fall in the middle. The stark contrast can be seen more clearly in Figure 7. It is important to note that this figure shows the confidence interval of the mean AUC score across all subset sizes and thus LR and C4.5 overlap, where as the Tukey HSD test takes into consideration that feature subset size is an additional

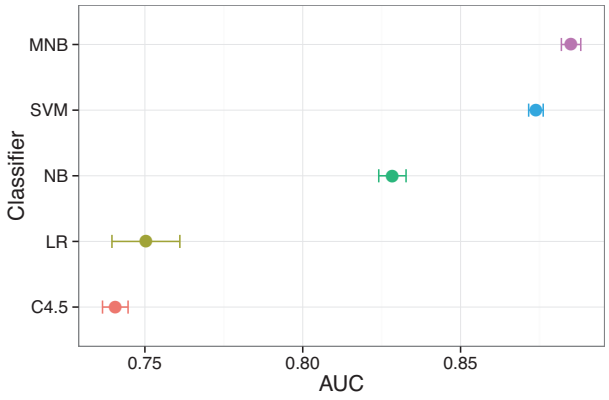

Figure 7: 95% confidence interval of mean AUC for each classifier using the Chi-Squared feature selector with feature subset sizes between 500 and 1000

factor in determining groups and thus does not group them together.

Conclusion

Methods for detecting online fake reviews have become increasingly important in recent years as both the popularity of online review sites and the number of fake reviews has increased. Current research has primarily focused on supervised classification using features extracted from the text of reviews, with little regard for how many features are being extracted. However, as the number of review sites and reviews grows, methods for curtailing the number of features is becoming necessary, since the feature set sizes can grow beyond what can be handled by traditional machine learning techniques.

In this study, we consider two common approaches from other text mining domains for limiting the number of features in a corpus and examine which method is better for multiple classifiers and desired feature subset sizes. The first method is to simply select the words which appear most often in the text. Alternatively, one can use filter based feature rankers (i.e. Chi-Squared) to rank features and then select the top ranked features. The crucial finding of this study is that there is not a one size fits all approach that is always better. In general, using Chi-Squared feature selection performs better for smaller subset sizes, while the less computationally intense method of word frequency performs just as well, if not better, for larger feature subset sizes.

Further work should include determining if the domain of the user reviews has any affect on the results. Also, other datasets (especially those which do not include AMT reviews) should be examined to verify that this trend generalizes.

Acknowledgments

We acknowledge partial support by the NSF (CNS-1427536). Opinions, findings, conclusions, or recommendations in this material are the authors' and do not reflect the views of the NSF.

References

Broder, A. Z.; Glassman, S. C.; Manasse, M. S.; and Zweig, G. 1997. Syntactic clustering of the Web. *Computer Networks and ISDN Systems* 29(8–13):1157–1166.

Bureau, C. C. 2014. Don't buy into fake online endorsements —Not all reviews are from legitimate consumers - www.competitionbureau.gc.ca.

Crawford, M.; Khoshgoftaar, T. M.; Prusa, J. D.; Richter, A. N.; and Al Najada, H. 2015. Survey of review spam detection using machine learning techniques. *Journal of Big Data* 2(1).

Fei, G.; Mukherjee, A.; Liu, B.; Hsu, M.; Castellanos, M.; and Ghosh, R. 2013. Exploiting Burstiness in Reviews for Review Spammer Detection. *ICWSM* 13:175–184.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11(1):10–18.

Jindal, N., and Liu, B. 2007. Review Spam Detection. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, 1189–1190. New York, NY, USA: ACM.

Jindal, N., and Liu, B. 2008. Opinion Spam and Analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, 219–230. New York, NY, USA: ACM.

Li, J.; Ott, M.; Cardie, C.; and 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)*, 1566–1576. Baltimore, Maryland: Association for Computational Linguistics.

López, V.; del Río, S.; Benítez, J. M.; and Herrera, F. 2015. Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets and Systems* 258:5–38.

Mukherjee, A.; Venkataraman, V.; Liu, B.; and Glance, N. 2013. What Yelp Fake Review Filter Might Be Doing? In *Seventh International AAAI Conference on Weblogs and Social Media*.

Ott, M.; Choi, Y.; Cardie, C.; and 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*, HLT '11, 309–319. Stroudsburg, PA, USA: Association for Computational Linguistics.

Ott, M.; Cardie, C.; and Hancock, J. T. 2013. Negative Deceptive Opinion Spam. In *HLT-NAACL*, 497–501.

Prusa, J. D.; Khoshgoftaar, T. M.; and Dittman, D. J. 2015. Impact of feature selection techniques for tweet sentiment classification. In *The Twenty-Eighth International Flairs Conference*.

Shojaee, S.; Murad, M.; Bin Azman, A.; Sharef, N.; and Nadali, S. 2013. Detecting deceptive reviews using lexical and syntactic features. In *2013 13th International Conference on Intelligent Systems Design and Applications (ISDA)*, 53–58.