Impact of Feature Selection Techniques for Tweet Sentiment Classification

Joseph D. Prusa, Taghi M. Khoshgoftaar, David J. Dittman
jprusa@fau.edu, taghi@cse.fau.edu, ddittman@fau.edu

Abstract

Sentiment analysis of tweets is a powerful application of mining social media sites that can be used for a variety of social sensing tasks. Common feature engineering techniques frequently result in a large numbers of features being generated to represent tweets. Many of these features may degrade classifier performance and increasing computational cost. Feature selection techniques can be used to select an optimal subset of features, reducing the computational cost of training a classifier, and potentially improving classification performance. Despite its benefits, feature selection has received little attention within the tweet sentiment domain. We study the impact of ten filter-based feature selection techniques on classification performance, using ten feature subset sizes and four different learners. Our experimental results demonstrate that feature selection can significantly improve classification performance in comparison to not using feature selection. Additionally, both choice of ranker and feature subset size significantly impact classifier performance. To the best of our knowledge, this is the first work which extensively studies feature selections effect on tweet sentiment classification.

Introduction

Microblogs, such as Twitter, have greatly changed how we experience media, share ideas, and interact with each other. The text mining of microblogs can be used to conduct social sensing and opinion mining. By collecting a large numbers of posts relating to a topic of interest and performing sentiment analysis (any of a number of methods that can determine the emotional polarity of a text passage), a statement can be made about the publics opinion on that topic. Tweet sentiment refers to the emotional polarity of a tweet and may be a binary positive/negative classification scheme, or may be more complicated and involve additional classes such as of neutral sentiment, attribute multiple sentiments to a single instance, or have different magnitudes of sentiment instead of binary classification.

Numerous studies have been conducted training tweet sentiment classifiers, but have primarily been concerned with feature engineering (the process of creating metrics based on the base data for use in subsequent analysis) and very few have examined using feature selection techniques. The purpose of studying feature selection techniques in combination with tweet sentiment is two-fold. Due to the diverse nature of tweets, feature engineering methods for Twitter data can potentially generate tens of thousands of features, though each instance will only contain a few features of the entire feature set (the remaing features being blank) as tweets are limited to 140 characters in length. Feature selection techniques select a subset of features, much smaller than the total number of features, reducing computational time needed to train and classify tweets. Additionally feature selection can improve classifier performance by eliminating redundant or irrelevant features and reducing over fitting.

This paper evaluates the performance of ten filter-based feature ranking techniques on a large high-dimensional dataset of collected tweets, each labeled to either having a positive sentiment or a negative sentiment. In order to test the feature rankers, we utilize ten feature subset sizes and a collection of four diverse classifiers. The results show that feature selection can have a great impact on the classification performance of models built for sentiment classification. In particular five feature ranking techniques (Chi-Squared, the Kolmogorov-Smirnov statistic, Mutual Information, area under the Precision-Recall curve, and area under the Receiver Operating Characteristic curve) improve classification performance over no feature selection. Additionaly, the results show that using between 75 and 200 features improves classification results over using the full feature set and using 50 features produces nearly identical results. Statistcal analysis (ANalysis Of VAriance (ANOVA) and Tukey's Honestly Significant Difference tests) confirm the the feature rankers and feature subset sizes mentioned above significantly improves classification performance. Thus, we can state that feature selection can be significantly beneficial to tweet sentiment classification performance and we recommend the use of Chi-Squared or Mutual Information with 100 to 200 features for the subset size as there are no significant differences between these rankers or subset sizes but they are significantly better than the other options.

The rest of the paper is organized as follows. The Related Works section contains previous research which relates to our experiment. The Methodology section introduces the specifics of our experiment. The Results section presents the results of our work. Lastly, the Conclusion section presents our conclusions and topics for future work.

Related Works

Feature selection seeks to choose an optimal subset of features by eliminating features that are irrelevant or offer no additional information compared to features within the optimal subset. Forman (Forman 2003) demonstrated many available feature selection techniques can be used to reduce dimensionality while improving classifier performance for a wide range of text classification problems. Guyon and Elisseeff (Guyon & Elisseeff 2003) expressed that performance increases from feature selection are in part due to reduction of over fitting.

Sentiment classification has received significant attention from web mining researchers. Go et al. (Go, Bhayani, & Huang 2009) proposed a method to collect and label tweets from which they extracted n-gram and part of speech features and trained classifiers using the resulting datasets. Numerous experiments have been conducted seeking to improve classification performance by augmenting the feature space with additional types of features. Asiaee et al. (Asiaee T. *et al.* 2012) added twitter specific features including hashtags and emoticons. Kouloumpis et al. (Kouloumpis, Wilson, & Moore 2011) examined using word polarity based on prior probabilities as additional features. Saif et al. (Saif, He, & Alani 2012) developed sentiment-topic features and semantic features to be used in conjunction with unigrams to achieve higher accuracy than unigrams alone. Sentiment classification has been used to address real world problems such as election prediction (Wang *et al.* 2012), and product sales (Liu *et al.* 2007).

While feature selection has been used in many data mining and machine learning applications and is common in other text classification domains it has received little attention in the domain of tweet sentiment classification. Saif et al. (Saif, He, & Alani 2012) studied the application of Information Gain (IG) as a feature ranker to select between 42 and 34,855 features (consisting of a combination of unigrams and either sentiment-topic features or semantic features) used to describe 1000 instances from the Stanford Twitter Corpus. They conclude that using more than 500 features yielded no significant improvement in classification performance; however, they only tested a single ranker and learner: information gain and Naïve Bayes. Chamlertwat et al. (Chamlertwat *et al.* 2012) reported optimal performance for classification of tweets as subjective or objective was achieved by combining SVM with IG; however, they did not report the number of feature selected, or what other classifiers were tested. Narayanan et al. (Narayanan, Arora, & Bhatia 2013) conducted an experiment showing the benefit of applying feature selection in the related domain of movie review sentiment classification, but only tested a single ranker, mutual information, using Naïve Bayes.

In this work, we conduct an investigation comparing the impact of various filter-based feature selection techniques and subset sizes against using no feature selection. We select ten feature selection techniques previously unstudied in this domain. We evaluate the performance of these techniques on tweet sentiment data using four learners and multiple feature subset sizes and compare their performance to using no feature selection.

Methodology

Dataset

The dataset for this experiment was constructed from the sentiment140 corpus, a publically available collection of 800,000 positive and 800,000 negative tweets (Go, Bhayani, & Huang 2009). Tweets were collected and labeled by searching Twitter for tweets containing specific emoticons (an icon or text representation of a facial expression used to convey emotions) associated with positive or negative sentiment, and then assigning the tweet to be either positive or negative based on the polarity of the emoticon used in the search. For our experiment, the first 1500 positive and 1500 negative instances from this corpus were used. Unigrams (individual words within the text of the tweet) were extracted as features with the requirement that each unigram be at least two characters in length and appear in at least two tweets in the dataset. The resulting dataset consisted of 3000 instances and 2388 features.

Feature Selection Techniques and Feature Subset Size

We chose three forms of filter-based feature selection: a commonly used feature ranker, Chi-Squared(CS); six Threshold-Based Feature Selection (TBFS) techniques, Gini-Index (GI), Kolmogorov-Smirnov (KS) statistic, Mutual Information (MI), Probability Ratio (PR), area under the Precision-Recall Curve (PRC), and area under the Receiver Operating Characteristic (ROC) curve; and three First-Order Statistics (FOS) based feature selection techniques called Signal-to-Noise (S2N) ratio, Significance Analysis of Microarrays (SAM) and Wilcoxon Rank Sum (WRS). For all of the feature rankers we used ten feature subset sizes: 5, 10, 15, 20, 25, 50, 75, 100, 150, and 200. These sizes were chosen to represent a diverse range of feature subset sizes.

The Chi-Squared test compares the observed distribution of class-feature value pairs to the distribution predicted by a chi-squared random distribution, and those features which are distinct from this null distribution are preferred. TBFS techniques treats feature values as ersatz posterior probabilities and classifies instances based on these probabilities, allowing us to use performance metrics as filter-based feature selection techniques. FOS rankers utilize first-order statistical measurements, such as mean and standard deviation, to create feature ranking techniques. For more details on the specifics of the CS ranker, TBFS rankers, and FOS rankers please refer to (Witten & Frank 2011), (Wang, Khoshgoftaar, & Van Hulse 2010), and (Khoshgoftaar *et al.* 2012) respectively.

Classification, Cross-Validation, and Performance Metric

We used four different classifiers to create inductive models using the sampled data and the chosen features. 5 Nearest Neighbor (k-nearest neighbors classifier with k =

Table 1: Classification Results

Classifier	Ranker	Feature Subset Size									
		5	10	15	20	25	50	75	100	150	200
5-NN	CS	0.60775	0.64098	0.65854	0.67309	0.68049	**0.69613**	0.69427	0.69469	0.69896	0.69301
	GI	0.49942	0.50075	0.49894	0.49928	0.50188	0.50675	0.52898	0.53729	0.54867	0.55433
	KS	0.60736	0.63384	0.64033	0.64413	0.64825	0.65781	0.65978	0.65665	0.65643	0.65340
	MI	0.60646	0.64009	**0.65905**	**0.67379**	**0.68150**	0.69461	**0.69712**	**0.69614**	0.68692	0.68131
	PR	0.50634	0.50511	0.50550	0.50773	0.50839	0.51378	0.51862	0.51914	0.52354	0.52428
	PRC	0.60461	0.63410	0.63417	0.64056	0.64595	0.66168	0.66232	0.65977	0.65771	0.65813
	ROC	0.60881	0.63532	0.63852	0.64495	0.65134	0.65528	0.65928	0.65642	0.65600	0.65308
	S2N	0.49933	0.49875	0.50092	0.49917	0.49958	0.55302	0.66389	0.69105	**0.70387**	**0.70086**
	SAM	0.50175	0.50193	0.49671	0.49540	0.50272	0.50353	0.50709	0.52114	0.52563	0.55014
	WRS	0.50050	0.50058	0.49899	0.49843	0.50204	0.50371	0.50521	0.50888	0.52105	0.51907
	None	**0.65191**	**0.65191**	0.65191	0.65191	0.65191	0.65191	0.65191	0.65191	0.65191	0.65191
C4.5	CS	0.60174	0.61949	0.64452	0.66184	**0.67591**	**0.69836**	**0.70404**	0.69819	0.69663	0.69898
	GI	0.50000	0.50000	0.50000	0.50000	0.50000	0.50371	0.52396	0.53270	0.53947	0.54231
	KS	0.59454	0.62584	0.64234	0.65454	0.66062	0.68084	0.68529	0.68440	0.68773	0.68710
	MI	0.59967	0.61746	0.64399	0.66013	0.67168	0.69624	0.70214	**0.70071**	0.69312	0.69241
	PR	0.50635	0.50509	0.50548	0.50773	0.50839	0.51351	0.51771	0.51678	0.52067	0.52099
	PRC	0.59718	0.62004	0.63001	0.64336	0.65224	0.68605	0.69424	0.68937	0.68781	0.68129
	ROC	0.59501	0.62160	0.64240	0.65360	0.65827	0.68091	0.68444	0.68246	0.68714	0.68470
	S2N	0.50000	0.50000	0.50000	0.50000	0.50000	0.54960	0.65122	0.67474	**0.69817**	**0.70288**
	SAM	0.50000	0.50000	0.50000	0.50000	0.50000	0.50062	0.50803	0.51765	0.51629	0.53831
	WRS	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000
	None	**0.66392**	**0.66392**	**0.66392**	**0.66392**	0.66392	0.66392	0.66392	0.66392	0.66392	0.66392
LR	CS	**0.60817**	0.64454	**0.66514**	0.68230	0.69105	0.72512	0.73821	0.74207	0.74192	0.73558
	GI	0.50000	0.50000	0.50000	0.50000	0.50000	0.50394	0.52875	0.53704	0.55014	0.55611
	KS	0.60696	**0.64312**	0.66167	0.67285	0.68257	0.72518	0.73915	0.74672	**0.75226**	**0.74872**
	MI	0.60718	0.64160	0.66640	**0.68391**	**0.69256**	**0.72599**	**0.74144**	**0.74761**	0.74088	0.74120
	PR	0.50637	0.50501	0.50549	0.50755	0.50827	0.51427	0.51907	0.51940	0.52598	0.52584
	PRC	0.60507	0.64172	0.65185	0.66468	0.67479	0.72395	0.73593	0.74296	0.74014	0.73485
	ROC	0.60804	0.64496	0.66029	0.67196	0.68357	0.72451	0.73833	0.74559	0.75048	0.74623
	S2N	0.50000	0.50000	0.50000	0.50000	0.50000	0.55141	0.67275	0.70292	0.71912	0.72685
	SAM	0.50000	0.50000	0.50000	0.50000	0.50000	0.50102	0.50772	0.52294	0.52397	0.55192
	WRS	0.50000	0.50000	0.50000	0.50000	0.50000	0.50242	0.50610	0.51223	0.52453	0.52641
	None	0.59623	0.59623	0.59623	0.59623	0.59623	0.59623	0.59623	0.59623	0.59623	0.59623
MLP	CS	**0.61039**	**0.64024**	0.66022	**0.67063**	**0.67943**	**0.70027**	**0.71318**	0.71308	**0.72117**	**0.72731**
	GI	0.50015	0.49979	0.49985	0.49964	0.49959	0.50125	0.52047	0.52463	0.54167	0.53968
	KS	0.60306	0.63145	0.64126	0.64734	0.65995	0.68358	0.69684	0.70518	0.71540	0.71398
	MI	0.60743	0.63865	**0.66215**	0.66980	0.67784	0.69958	0.71123	**0.71832**	0.71876	0.71565
	PR	0.50629	0.50699	0.50772	0.50785	0.50803	0.50684	0.50832	0.50636	0.50170	0.50298
	PRC	0.60028	0.62780	0.63258	0.64674	0.65543	0.68334	0.70139	0.70390	0.71708	0.72097
	ROC	0.60380	0.63004	0.64131	0.65454	0.65688	0.68255	0.69936	0.70027	0.71606	0.71591
	S2N	0.50010	0.50041	0.50018	0.50010	0.49962	0.54384	0.65915	0.68680	0.70451	0.70605
	SAM	0.49998	0.49930	0.49941	0.49968	0.49933	0.49968	0.50233	0.50749	0.50843	0.51215
	WRS	0.50006	0.49979	0.49984	0.50072	0.50053	0.50062	0.50083	0.50198	0.50427	0.50490
	None	0.53133	0.53133	0.53133	0.53133	0.53133	0.53133	0.53133	0.53133	0.53133	0.53133

five; denoted as 5-NN in this work) and Logistic Regression (LR), C4.5 decision tree (C4.5), and Multilayer Perceptron (MLP), implemented using the WEKA toolkit (Witten & Frank 2011) using default values unless otherwise noted. Due to space limitations (and because these four classifiers are commonly used) we will not go into the details of these techniques. However it should be noted that for 5-NN the choice of k = five and the weight by distance parameter being set to "Weight by $1/distance$" and for MLP a network of one hidden layer and the $validationSetSize$ parameter was set to "10" was chosen based on preliminary research. For more information on these learners, please refer to (Witten & Frank 2011).

Cross-validation refers to a technique used to allow for the training and testing of inductive models without resorting to using the same dataset. In this paper we use five-fold cross-validation. Additionally, we perform four runs of the five-fold cross validation so as to reduce any bias due to a chance split. However, it should be noted that the process of feature selection is performed on every training dataset generated by the four runs of five-fold cross-validation.

The classification performance of each model is evaluated using the Area Under the Receiver Operating Characteristic Curve (AUC) (Khoshgoftaar *et al.* 2012). Mathematically, this is the same metric as described above in the Feature Selection Technique and Feature Subset Size section, but there is a major distinction: for feature selection, we use an ersatz posterior probability to calculate the metric, but when used for evaluating classification models, the actual posterior probability from the model is used. To reduce confusion, we use AUC when referring to the performance metric.

Results

In this work, we seek to observe the impact of using feature selection on tweet sentiment classification. We use a combination of ten feature rankers, ten subset sizes, and four classifiers. Table 1 presents the results of our experiments. In each column the best model for each feature subset size is indicated in **boldface**. It is important to note that "None", meaning no feature selection was performed, is included as if it was an additional ranker in the bottom row of each table. This allows the impact of feature selection to be evaluated against not using feature selection. It should be noted that the feature subset sizes listed in the tables are not relevant for None as it uses all 2388 features available from the dataset and is repeated for each subset size.

First, we look at the results of 5-NN. It can be observed that 5-NN requires at least 15 features to improve upon None. When selecting between 15 and 100 features MI achieves the highest AUC, excluding 50 features, where CS is the best ranker. S2N achieves the highest AUC values for 150 and 200 features, with S2N and 150 feature being the best performing model trained with 5-NN. This last result is interesting as S2N performs poorly for small subset sizes, being the worst performer when selecting 5, 10, 20 or 25 features. In general GI, PR, SAM and WRS perform poorly for all subset sizes and never outperform None.

When we observe the results for the decision tree learner C4.5, we see that None achieves higher AUC values than rankers for models trained with 5, 10, 15 or 20 features. Using CS yields the highest performance for models trained with 25 to 75 features, while MI is best for 100 features. Again S2N yields the best performance for 150 and 200 features. The highest AUC value for an individual model was achieved by CS with 75 features. GI, PR, SAM and WRS perform poorly for all subset sizes. S2N is again observed to be a worst performer for small subset sizes, achieving lower AUC values than None for 75 or less features.

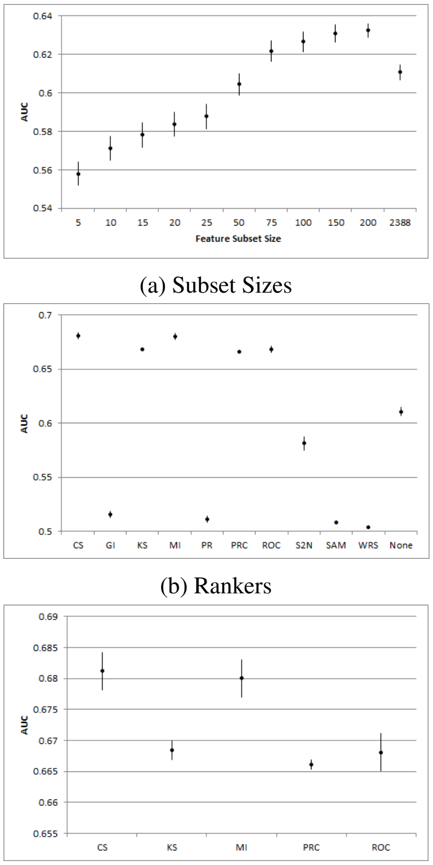
Table 2: ANOVA Results: # of Features and Rankers

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
# Features	6.453267	9	0.71703	884.7823	0
Ranker	49.86691	10	4.986691	6153.352	0
Error	13.15177	8780	0.001498		
Total	69.47194	8799			

We can see from the results from the LR classifier, that the top performing option for any subset size is a ranker, and again GI, PR, SAM and WRS with any subset size fail to outperformed None. CS is best for 5 or 15 feature and ROC for 10 features. For subset sizes between 20 and 100 features MI achieves the highest AUC values. The best performance for 150 and 200 features was achieved with KS, with KS and 150 features achieving the highest AUC value among the models trained with LR. S2N is never the best ranker for a subset size and is among the worst rankers for small subsets.

Lastly, the results for MLP show that CS is the best performing ranker for 8 out of 10 subset sizes; MI is the best for 15 and 100 features. GI, PR, SAM, WRS perform poorly for all subset sizes and S2N performs poorly for small subset sizes. Like LR, the top classification performance with any of the tested subset sizes occurs when using a ranker. The highest AUC value was achieved by the model trained using CS and 200 features.

In summary filter-based feature selection improved classification performance for all learners tested; though, using a small feature subset was inferior to None for both 5-NN and C4.5. The highest AUC values were achieved for 5-NN using S2N with 150 features, for C4.5 using CS with 75 features, for LR using KS with 150 features, and finally for MLP using CS with 200 features. The best performing model was the result of training a classifier using LR with 150 features selected using KS. In general models performed better when rankers selected larger numbers of features, best performing models for each learner had 75 or more features. Other results of note include: S2N performs poorly with small subsets for all learners, but performs well for larger subsets (it was the best ranker for 5-NN and C4.5 for 150 and 200 features); PR, SAM and WRS performed worse than using no feature selection for all learners and subset sizes, while GI managed improved performance over None only when selecting 150 or 200 features with C4.5; and PRC is never the best performing ranker, and is generally inferior to CS, KS, MI and ROC, but better than PR, SAM, and WRS.



(a) Subset Sizes

(b) Rankers

(c) Top Rankers

Figure 1: Tukey HSD Results

Statistical Analysis

Statistical significance of the results presented in this work was tested by performing two-way ANOVA with a 5% confidence level using Microsoft Excel. The results are presented in Table 2 and show both choices of ranker and feature subset size to be significant factors in determining classifier performance. In addition, to the ANOVA values we present the Tukey's Honestly Significant Difference test to compare the factors. Figures 1a through 1c present plots of mean AUC values across all learners and the four runs of five-fold cross-validation associated with each subset size or ranker with accompanying confidence intervals.

Figure 1a presents mean AUC values for each feature subset size, including 2388 features, the result of using no feature selection. It can be observed that using 200 features is best on average, but not significantly different from 150, or 100 features. 75 features significantly outperforms None but is significantly less than 100 to 200. Most importantly, Figure 1 shows that the performance gains due to selecting subsets of features are significant. Additionally, using 50 or less features is confirmed to perform worse than None (though 50 is not significantly lower than None).

Figure 1b and figure 1c show mean AUC values for rankers, including using no ranker (labeled None). Figure 2 displays all rankers while figure 3 is a close up of the 5 rankers that achieve significantly better classification performance to None. CS achieves the highest mean AUC value, but is not significantly different than MI. KS, PRC and ROC form the next grouping, still performing significantly better than using None. The remaining rankers are significantly worse than None. S2N achieves a similar, but lower mean AUC value compared to None. As expected GI, PR, SAM and WRS are significantly worse than other rankers.

Conclusion

Feature engineering methods for tweet sentiment classification often generate a very large number of features. Combined with a large number of instances the resulting dataset can be of very high dimensionality. Additionally training classifiers on a large dataset is computationally expensive. Feature selection, which has received little attention in tweet sentiment classification research, selects an optimal subset of features, which reduces the dimensionality of the dataset, helps to reduce computational costs, and possibly improves classification performance.

This study examined ten filter-based feature selection techniques and compares them against using no feature selection across four diverse learners. These techniques are used to select ten different feature subsets from a dataset consisting of 3000 tweets from the sentiment140 corpus. Our experiments show that feature selection can significantly improve classifier performance for all learners. Using 200 features is generally best, but 100 and 150 features also performed similarly. 75 features does outperform None, but performed worse than 100 to 200 features. Using feature selection to select 50 or fewer features generally results in poor performance, inferior to using no feature selection. The statistical significance of our findings was tested by performing ANOVA analysis. It was found the performance improvement achieved by selecting 75 or more features was statistically significant. Additionally it was found that only CS, KS, MI, PRC and ROC resulted in statistically significant performance improvements compared to using no feature selection. Additionally the difference between the top performing rankers, CS and MI, and the top performing subset sizes, 100 to 200, is not statistically significant; however, the gap between these and the other rankers and subset sizes are statistically significant.

From the results of our experiments, we conclude that feature selection techniques can be quite effective in helping to alleviate problems associated with high dimensional datasets within this domain. Both choice of ranker and choice of subset size have significant impact on classifier performance. While not every ranker improves results, we recommend using either CS, MI or KS with between 100 to 200 features to achieve good performance as there are no significant differences between these rankers or subset sizes. These results are promising and future work should investigate more feature selection techniques and using more than 200 features. This study should also be expanded to include other datasets, in order to determine if the trends discovered in this work are present in other datasets.

References

Asiaee T., A.; Tepper, M.; Banerjee, A.; and Sapiro, G. 2012. If you are happy and you know it... tweet. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, 1602–1606.

Chamlertwat, W.; Bhattarakosol, P.; Rungkasiri, T.; and Haruechaiyasak, C. 2012. Discovering consumer insight from twitter via sentiment analysis. *J. UCS* 18(8):973–992.

Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3:1289–1305.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1–12.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3:1157–1182.

Khoshgoftaar, T. M.; Dittman, D. J.; Wald, R.; and Fazelpour, A. 2012. First order statistics based feature selection: A diverse and powerful family of feature selection techniques. In *Proceedings of the Eleventh International Conference on Machine Learning and Applications (ICMLA)*, 151–157. ICMLA.

Kouloumpis, E.; Wilson, T.; and Moore, J. 2011. Twitter sentiment analysis: The good the bad and the omg! *ICWSM* 11:538–541.

Liu, Y.; Huang, X.; An, A.; and Yu, X. 2007. Arsa: A sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, 607–614.

Narayanan, V.; Arora, I.; and Bhatia, A. 2013. Fast and accurate sentiment classification using an enhanced naive bayes model. In *Intelligent Data Engineering and Automated Learning–IDEAL 2013*. Springer. 194–201.

Saif, H.; He, Y.; and Alani, H. 2012. Alleviating data sparsity for twitter sentiment analysis. CEUR Workshop Proceedings (CEUR-WS. org).

Wang, H.; Can, D.; Kazemzadeh, A.; Bar, F.; and Narayanan, S. 2012. A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, 115–120.

Wang, H.; Khoshgoftaar, T.; and Van Hulse, J. 2010. A comparative study of threshold-based feature selection techniques. In *Granular Computing (GrC), 2010 IEEE International Conference on*, 499–504.

Witten, I. H., and Frank, E. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition.