Cross-Domain Sentiment Analysis: An Empirical Investigation

Brian Heredia*, Taghi M. Khoshgoftaar†, Joseph Prusa‡ and Michael Crawford§
Department of Computer and Electrical
Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida
Email: *bheredia@fau.edu, †khoshgof@fau.edu, ‡jprusa@fau.edu, §michaelcrawf2014@fau.edu

***Abstract*—Understanding the sentiment conveyed by a person is a crucial task in any social interaction. Moreover, it can be used to gain insight and understanding of views held by many people. Sentiment classification is not limited to human interaction, as text can also convey the sentiment of the author. Opinion mining in text is a long studied field in machine learning. This study focuses on two of the many text domains used in the field of sentiment analysis: reviews and tweets. In this study, we aim to determine the the effect of performing cross-domain sentiment classification using either reviews or tweets as training data. We conduct an empirical investigation using two tweet datasets and one review dataset, and three classifiers. We conduct 18 experiments, varying the training dataset, the classifier used to build the model, and the dataset used to evaluate the model built. Our results show that training with tweets, for both datasets, yields an effective classifier for reviews. However, the converse, using reviews to classify sentiment in tweets, has the worst performance of all models, producing AUC values ranging from 0.59 to 0.65. Our best model is generated using tweets to train a Multinomial Naïve Bayes classifier, and using reviews to evaluate. Multinomial Naïve Bayes was the best performing learner, producing the highest AUC in 5 out of the 6 combinations of training/test datasets. To the best of our knowledge, this study is the first to examine the effects of cross-domain sentiment classification using tweets and reviews.**

***Keywords*—Sentiment analysis, cross-domain, machine learning, tweet, reviews**

I. Introduction

Sentiment analysis occurs in every human interaction. We use it when gauging a reaction to a response or determining the emotional status of one another. However, sentiment is not limited to human interaction as text can also relay the sentiment of the author. Thus, mining text for sentiment can provide additional insight into the opinions of the author and provide important information regarding public views. Sentiment analysis in text is not a new domain, countless tweets and reviews have been analyzed for sentiment and used to predict important events such as the general election [20] and movie box office performance [12]. Tweets are micro-blog posts generated by users on Twitter [1]. Users express their views and ideas through tweets, making them ripe for sentiment analysis. Reviews are an assessment of a certain entity, whether it be a product, person, or location, and can be found on websites such as Yelp and Amazon. Reviews are, by nature, opinionated, making them a good candidate for sentiment analysis. Tweets and reviews have both been used extensively for sentiment classification.

From a machine learning perspective, determining sentiment in text requires previously labeled data, from which patterns are learned and used to identify sentiment in future documents. The process of using previously labeled data to train a classifier is known as supervised learning. The majority of supervised learning approaches for sentiment analysis use words found in the text as features describing the review and sentiment as the class label. Other features such as syntactic features, lexical features, and frequency of words can also be used at the expense of computational resources. Sentiment labels can range from positive and negative to actual emotions. A classifier probes the data to learn patterns in the text, the classifier then predicts sentiment of new instances based on their features. The performance of the classifier is directly affected by the training data used. Thus, data used to train a classifier is usually from the same domain as the data used to test it (i.e. tweets are used to train a classifier which predicts sentiment of tweets). However, it is possible to use data from a related domain to train a classifier to classify instances from the domain of interest (i.e. use tweets to train the classifier which predicts sentiment of reviews). This process is known as cross-domain sentiment analysis.

Cross-domain classification is appealing due to the large amount of data that exist in certain domains and/or the ease of obtaining certain types of data. For example, the full sentiment140 dataset contains 1.6 million tweets. With such large amounts of labeled data, using tweets to train a model which detects sentiment in a domain with little available labeled data would be very useful. A generalized sentiment analysis classifier, which performs well in multiple domains, would allow for experimentation on domains where there is not a large amount of available data.

In this study, we aim to determine the performance of cross-domain sentiment classification using either tweets or reviews to train a model. We explore the effects of using tweet sentiment data to train a classifier and evaluate the classifier on review sentiment data, and vice versa. To this aim, we perform eighteen experiments using three machine learning classifiers: Support Vector Machines (SVM), Naïve Bayes (NB), and Multionmial Naïve Bayes (MNB). Three distinct datasets are used in our experiments; the first dataset is created using the Sentiment140 corpus [5], the second is the semEval dataset [18], and the third being a dataset spanning three review domains [9]. The Area Under the receiver operator

978-1-5090-3207-5/16 $31.00 © 2016 IEEE
DOI 10.1109/IRI.2016.28

characteristic Curve (AUC) is our chosen performance metric.

We observe better performance when using tweet data to train a classifier and evaluating using review data. The highest AUC was observed when using MNB and tweets (Sentiment140) as training data to determine sentiment in reviews. Moreover, using MNB and Sentiment140 to determine sentiment in reviews performed better than using MNB with the Sentiment140 corpus to determine sentiment in tweets found in the semEval dataset. We also note using the review dataset for training results in a classifier with lower performance than is observed with either tweet dataset. Our results show tweets can be used to train a classifier to detect sentiment in reviews, but reviews should not be used to train a model which classifies sentiment in tweets.

The remainder of this paper is organized as follows. Section II presents previous works related to sentiment analysis using tweets and reviews. Section III presents our methodology, including dataset information, classifiers, experimental setup, and performance metric. Section IV presents our results and statistical analyses. Finally, Section V presents our conclusions and possible avenues for future work.

II. Related Works

Sentiment analysis using tweets and reviews has received significant attention, as it can provide insight on products, public figures, or a plethora of other topics. One of the more prominent studies exploring sentiment analysis using tweets constructed a dataset using an automated sentiment labeling method using emoticons [5]. Emoticons are a combination of symbols used to express emotion in text. For example, :) is considered a smile and associated with a positive emotion. Tweets were collected and labeled using emoticons, creating the Sentiment140 dataset of 1.6 million positive and negative tweets. Three machine learning classifiers were used to predict sentiment: MNB, SVM, and Maximum Entropy. Features used were unigrams, bigrams, unigrams and bigrams, and unigrams with Part-Of-Speech (POS) tags. The classifiers were trained using the Sentiment140 dataset and then tested on a smaller dataset of manually labeled tweets. They found using unigrams and unigrams with bigrams had the best performance. Their results show SVM with the highest performance when using unigrams and maximum entropy with the highest performance using unigrams and bigrams together; however. the difference between all three classifiers is less than 2% for both feature spaces. The authors stated one of the reasons bigram features alone did not perform well is that tweets are short 140 character micro-blogs, which creates a very sparse feature space when using bigrams.

Other studies have looked into more advanced machine learning techniques to improve sentiment analysis using tweets. Saif et al. [19] looked at methods to reduce feature space in tweet sentiment analysis. Kouloumpis et al. [8] explored the effects of the boosting ensemble technique in tweet sentiment analysis, while Hannek et al. [7] determined the effects of bagging on tweet sentiment analysis. More recently, Prusa et al. [15] used a combination of ensemble techniques and feature selection to improve performance of tweet sentiment analysis.

Tweets have only recently gained traction as a source of data for sentiment analysis, but reviews have been used since the start. In 2002, Pang et al. [13] used movie reviews to predict sentiment. Their data was obtained from IMDB, selecting reviews where the authors rating was expressed using some numerical system. Their final dataset consisted of 752 negative reviews and 1301 positive reviews. From the full dataset 700 positive and 700 negative reviews were chosen randomly and split into three folds for cross-validation, maintaining a balanced class ratio. Only unigrams and bigrams were considered as features for this study. They apply three machine learning techniques to movie review data: NB, SVM, and Maximum Entropy. Overall, their results show SVM performing the best, while NB performs the worst and unigrams performing better than bigrams.

Previous cross-domain research has involved finding features which are domain independent and bridging the gap between the domains using these features [10]. Variations of this approach have been used to bridge domains, notably using domain independent topics, part-of-speech tags, and related semantic spaces.

In a study by Aue et al. [2], cross-domain sentiment analysis was done using four different review domains: movie reviews, book reviews, Product Support Services (PSS) web survey data, and Knowledge Base (KB) web survey data. Their feature sets were unique for each dataset and composed of unigrams, bigrams, and trigrams, but only features which occurred three times or more in any of the reviews were included for that feature domain. The authors employed a Support vector machine classifier. Four experimental methodologies were used: (1) training on a mixture of labeled data from other domains, (2) training a classifier from step 1 but limiting the set of features to those observed in the target domain, (3) using an ensemble of classifiers from other domains, and (4) combining labeled data with unlabeled data in the target domain. Their results show that the cross-domain approaches do not perform as well as the in-domain approach. They also propose approaches to overcome the domain specificity issue, such as training one classifier on all the data from all domains, limiting features to those in the target domain, an ensemble of classifiers, and using in-domain unlabeled data.

Our study is unique in that we examine the performance of cross-domain sentiment analysis across two different types of data, tweets and reviews, three machine learning algorithms, and three distinct datasets. Our study differs from Aue et al. [2] in that we employ more machine learning algorithms and we use tweet data, which is distinctly different from reviews or survey data. Tweets are shorter, more informal bodies of text and are usually written in shorthand and use various acronyms. Tweet data is more readily available for use, as there are multiple labeled tweet datasets and methods to automatically label tweets based on emoticons [5]. To the best of our knowledge, no other study has examined the effects of using tweets to classify reviews and vice versa.

III. Methodology

A. Datasets

This section will provide some insight into the data used for our experiments. Three distinct datasets are used in our study: a subsection of the Sentiment140 corpus [5], the semEval dataset [18], and a review spam dataset containing sentiment labels [9]. Table I shows the class distribution of all the datasets. The feature space across all three datasets consists of a bag-of-words model. Features are extracted from the training data and applied to the test data to ensure both datasets have the same feature space. All of the words used in the tweets and reviews were vectorized using the StringToWordVector filter in the WEKA toolkit [6]. We use unigram features in this study.

The Sentiment140 corpus is a collection of tweets pulled directly from Twitter using their API. These tweets were then automatically labeled using emoticons. Eight emoticons were used to label tweets as either positive or negative. This labeling process trades speed for accuracy, as some tweets may be mislabeled due to the context of the emoticon. Once these tweets were labeled, the data was pre-processed by removing the emoticons from the tweets, removing retweets, removing duplicated tweets, and removing instances that contained both positive and negative sentiment in the same tweet. The final dataset consists of 1.6 million tweets, with 800,000 positive and 800,000 negative instances. In our study, we take a subset of this total dataset; we use 10,000 instances sampled randomly, without replacement, from the full dataset.

The semEval dataset is also a collection of tweets, however, it is significantly smaller than the Sentiment140 corpus. Unlike the Sentiment140 corpus, the semEval dataset was labeled manually; thus, it is expected to have a smaller percentage of misclassified instances. The dataset contains positive, negative, and neutral class labels. The neutral instances were removed from the semEval datasets, therefore only positive and negative instances were used in our dataset. This pre-processing step was performed to match the number of classes found in the sentiment140 dataset. The class distribution is imbalanced in this dataset, as there are approximately two times as many positive instances as negative, which can be seen in Table I.

The final dataset comes from a paper by Li et al. [9], whose study was in the domain of untruthful review detection [4]. However, their data also contains sentiment of the review. The data contains reviews from three domains: doctors, hotels, and restaurants. The dataset contains the reviews, the text of the review, the rating given, the domain the review belongs to, and the class label (positive or negative).

Dataset	Positive	Negative	Total
Sentiment140	5,000	5,000	10,000
semEval	3,420	1,399	4,819
reviews	1,896	940	2,836
Total	10,316	7,339	17,655

TABLE I: Dataset Characteristics

B. Classifiers

In our study, we employ three different classifiers: Naïve Bayes, Multinomial Naïve Bayes, and Support Vector Machines. These classifiers were chosen as they commonly perform well in sentiment analysis research [14] [5] [13]. All classifiers were implemented using the WEKA toolkit [6] with default parameters, unless noted otherwise.

Naïve Bayes [17] falls under the category of Bayesian learners. Naïve Bayes aggregates Bayesian probabilities to approximate the posterior probability of an instance belonging to a class based on its feature values [21]. It makes the naïve assumption that all features are independent. Although, in general, this is not the case with the majority of features, Naïve Bayes still offers good performance. This is due to how the dependencies interact on a local and global level. Locally there may be strong dependencies between two features describing a class. However, when looked at globally the dependencies between features cancel each other out, allowing Naïve Bayes to perform optimally [22]. Using Naïve Bayes, the class membership of an instance is calculated as follows:

$$\hat{y} = p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Similar to Naïve Bayes, Multinomial Naïve Bayes [11] is a second Bayesian learner and assumes a multinomial distribution to the data, rather than a Gaussian distribution. MNB still makes the naïve assumption of feature independence. However, the difference between NB and MNB is in the calculation of the likelihood. For example, in MNB for text classification, a document is assigned to the class which has the highest conditional probability. The probability is calculated as a count of the words in the document that overlap with the class, divided by the total number of words. If the product of the prior probability and the words found in class 1 is larger than the number of words overlapped with class 2, then the instance is classified as class 1, otherwise it is classified as class 2.

We have previously used Support Vector Machine in various machine learning problems involving reviews and tweets and found this classifier to have good performance in these domains [3] [16]. SVM is based on the assumption that there is a linear discriminant between the feature space of two classes. SVM attempts to find a hyperplane that divides instances into two groups. The best such hyperplane would be the one that maximizes the distance between the hyperplane and members of each class. For our models, the complexity constant c was set to 5.0 and the buildLogisticModels parameter set to true.

C. Experimental Setup and Performance Metric

	Train	Test	Learner
1	sentiment140	semEval	SVM
2	sentiment140	review	SVM
3	review	sentiment140	SVM
4	review	semEval	SVM
5	semEval	sentiment140	SVM
6	semEval	review	SVM
7	sentiment140	review	NB
8	sentiment140	semEval	NB
9	review	sentiment140	NB
10	review	semEval	NB
11	semEval	sentiment140	NB
12	semEval	review	NB
13	sentiment140	review	MNB
14	sentiment140	semEval	MNB
15	review	sentiment140	MNB
16	review	semEval	MNB
17	semEval	sentiment140	MNB
18	semEval	review	MNB

TABLE II: Experimental setup

Each dataset is used to train a model twice and used to test a model twice for each classifier. For example, we use the sentiment140 subset to train a SVM classifier and the semEval dataset to evaluate the classifier. We then use the same sentiment140 subset to train a SVM classifier, but we now use the review dataset to evaluate it. This process is repeated, alternating the datasets used to train and test the model, and the classifier used until all combinations of datasets and classifiers have been exhausted. Our experiments consist of three learners and six combinations of training and test data. We conduct a total of 18 experiments (3 learners x 6 dataset combinations) to determine the performance of cross-domain sentiment analysis using tweets and reviews. Table II provides the complete combinations and learners used for all the experiments conducted.

Performance of the classifiers were measured using the Area Under the receiver operator characteristic Curve [21]. We elect to use AUC as it plots the performance of the model across all decision thresholds. The AUC is a graph of the False Positive Rate versus the True Positive Rate, and the area under the curve depicts performance of the model across all decision thresholds. Thus, a larger area under the curve means a better performing classifier. AUC values range from 0 to 1.0, with 0.5 being no better than a random guess and 1.0 being the perfect fit.

IV. Results

In this study, we attempt to empirically determine whether tweets are able to successfully classify sentiment in reviews and vice versa. In this section, we present the results of our experiments and determine the effects of cross-domain sentiment analysis using tweets and reviews. For a deeper understanding, results will be grouped by learner and examined in more detail.

A. *Naïve Bayes*

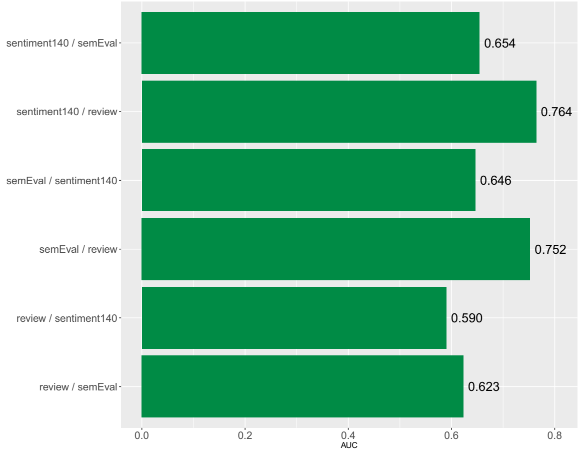

Fig. 1: AUC values using NB for Cross-Domain Sentiment Analysis

The results for the Naïve Bayes classifier are presented in Figure 1, the y axis provides information in terms of which dataset was used to train and which was used to test the classifier (train / test format). From Figure 1, we can see the best performing combination is using the sentiment140 dataset to train the classifier and using the review dataset to evaluate. However, the difference between using sentiment140 and semEval to train and reviews to test is approximately 0.01. It is interesting to note that using tweets to predict review sentiment performs better than using tweets to predict tweet sentiment in a different dataset. We find that using semEval to predict sentiment in the sentiment140 data and vice versa perform significantly worse than reviews. The lowest performance is observed when using reviews to train the model and tweets to evaluate.

B. *Support Vector Machine*

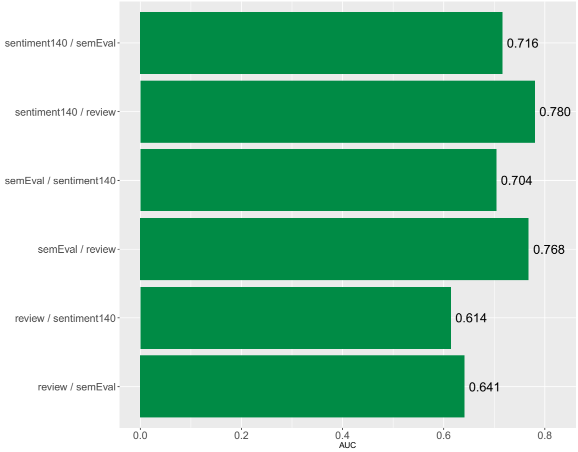

Fig. 2: AUC values using SVM for Cross-Domain Sentiment Analysis

Results are again presented in a bar graph. Figure 2 shows performance of the support vector machine classifier on cross-domain sentiment analysis. We observe similar trends as with NB, however, in general, the AUC values when using the SVM classifier are higher. We see that using tweets to train and reviews to evaluate leads to the best performance, with sentiment140 producing a higher AUC than semEval. Similar to the results of NB, using tweets to predict review sentiment results in a higher AUC than using tweets to predict tweet sentiment in a different tweet dataset. The worst performance is observed when using reviews to train a the sentiment140 data to evaluate.

C. *Multinomial Naïve Bayes*

Finally, the results for the Multinomial Naïve Bayes classifier are presented in Figure 3. From Figure 3, we observe the same trends found when using NB and SVM. Moreover, the trends are more exaggerated than with the previous classifiers. The best performer is still using sentiment140 to predict sentiment in reviews; however, the difference between using sentiment140 and semEval to predict review sentiment is 0.07. This is significantly larger than the 0.01 difference in NB and the 0.015 difference in SVM. When comparing the tweet datasets we observe better performance when using sentiment140 to train a model that classifies sentiment in semEval. The worst performer is again using reviews to train a model and sentiment140 to evaluate it.

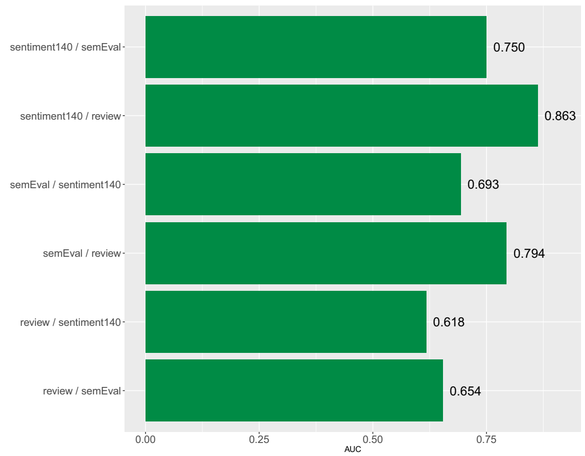

Fig. 3: AUC values using MNB for Cross-Domain Sentiment Analysis

D. Discussion

The results presented in the previous subsections show similar trends across all three learners. Table III presents the AUC values grouped by learner and train/test combination, the best performer for the dataset combination is shown in **boldface**. In terms of classifiers, we see Multinomial Naïve Bayes resulting in the highest AUC for five of the six combinations, whereas SVM produces the highest AUC for the final combination. From this table and the results in previous sections, we can say that when performing cross-domain sentiment analysis between tweets and reviews, MNB serves as the best classifier.

	SVM	NB	MNB
sentiment140 / semEval	0.72	0.65	**0.75**
sentiment140 / review	0.78	0.76	**0.86**
review / sentiment140	0.61	0.59	**0.62**
review / semEval	0.64	0.62	**0.65**
semEval / sentiment140	**0.70**	0.65	0.69
semEval / review	0.77	0.75	**0.79**

TABLE III: AUC values by learner and dataset combination

When examining Table III and the previous results, some interesting trends appear that warrant further examination. The first of these trends is that using sentiment140 to predict sentiment in reviews is the top performing combination across all classifiers. We see that the highest AUC for this combination was achieved using the Multinomial Naïve Bayes classifier. Keeping with the trend, the second top performing combination for all classifiers is using the semEval data to predict sentiment in reviews. These results confirm that tweets are a viable option for training a classifier that predicts sentiment in reviews. However, we see that the worst performing combination is using reviews to predict sentiment in tweets. The worst performer is using reviews to predict sentiment in the sentiment140 data, followed by using reviews to predict sentiment in the semEval data.

It is interesting to note that the best performer and the worst performer, both use the same datasets, however, tweets serve as better data for predicting results in reviews. This may be due to the difference in length between tweets and reviews. As tweets are composed of no more than 140 characters, patterns found in tweets are smaller than patterns found in reviews. Smaller patterns are applicable to both long and short text instances, however, longer patterns are not applicable to shorter bodies of text. Thus, the patterns learned from tweets are directly applicable to reviews, but longer patterns learned from reviews cannot be applied to tweets. Additionally, reviews are easier to classify as the reviews are written by users with the intent of conveying an opinion and users give a rating to their review that can be directly converted into a sentiment label.

Another trend worth noting is that using tweet sentiment datasets for training and testing results in lower AUC scores than training on tweets and testing on reviews. This likely indicates that determining tweet sentiment is a more difficult task than determining review sentiment. It is possible that predicting sentiment in tweets is more difficult, as tweets are not necessarily written to convey opinion and they are shorter, therefore less data is available for determining sentiment. Additionally, tweet sentiment data is labeled with no knowledge the users actual sentiment, while reviews sentiment labels are generated directly from user created scores. This likely leads to reviews having more accurate labels than tweets.

V. Conclusion

In this study, we set out to determine the performance of cross-domain sentiment analysis using either tweets or reviews to train the model. To this aim, we conducted an empirical study consisting of three datasets and and three different classifiers. A total of 18 experiments were conducted across 6 combinations of training/testing datasets and three classifiers. Two datasets were from the tweet sentiment domain, while the third dataset was from the review domain. Support vector machine, Multinomial Naïve Bayes and Naïve Bayes were used to determine sentiment.

Our results show we can train a cross-domain classifier to classify sentiment in a different text domain than the one the classifier was trained in. Regardless of what classifier is used, the best performing combination is using the sentiment140 data to classify sentiment in reviews. We found that classifiers trained on tweets perform well when classifying reviews; however, the converse, training on reviews to classify tweets, does not yield high AUC values. Thus, a relationship between tweets and reviews exists, where tweets can be used to train an effective classifier for reviews. We note that this result may be due to patterns found in tweets that are directly applicable to reviews, while those found in reviews may not be as applicable to tweets. Since tweets are smaller, the patterns found are applicable to larger text documents, such as reviews; however, as reviews are longer than tweets, the patterns learned from reviews may not fit tweets. It is also noteworthy to mention that reviews are written with opinions in mind and have a rating which directly correlates with sentiment, making sentiment more apparent in reviews. Our results show that tweets are a valid training model for determining sentiment in reviews; however, reviews are not suited to classify sentiment in tweets.

Future work can involve testing cross-domain sentiment analysis on other datasets to see if results generalize. Using datasets from other domains in addition to tweets and reviews is also an avenue for future work.

VI. Acknowledgement

The authors gratefully acknowledge partial support by the National Science Foundation, under grant number CNS-1427536. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] (2015, Jun.) Twitter usage statistics. [Online]. Available: http://www.internetlivestats.com/twitter-statistics//

[2] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: A case study," in *Proceedings of recent advances in natural language processing (RANLP)*, 2005.

[3] M. Crawford, T. M. Khoshgoftaar, and J. D. Prusa, "Reducing feature set explosion to facilitate real-world review spam detection," in *Proceedings of the 29th International FLAIRS conference*, May 2016.

[4] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *Journal Of Big Data*, vol. 2, no. 1, pp. 1–24, Dec 2015. [Online]. Available: http://link.springer.com/article/10.1186/s40537-015-0029-9

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

[6] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[7] A. Hannak, E. Anderson, L. F. Barrett, S. Lehmann, A. Mislove, and M. Riedewald, "Tweetin'in the rain: Exploring societal-scale effects of weather on mood." in *ICWSM*, 2012.

[8] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" *ICWSM*, vol. 11, pp. 538–541, 2011.

[9] J. Li, O. Myle, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, June 2014, pp. 1556–1576. [Online]. Available: http://anthology.aclweb.org/P/P14/P14-1147.pdf

[10] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, pp. 1–167, 2012.

[11] A. McCallum and K. Ni-gam, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, Jul 1998.

[12] C. Meador and J. Gluck, "Analyzing the relationship between tweets, box-office performance and stocks," *Methods*, 2009.

[13] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[14] J. Prusa, T. Khoshgoftaar, D. Dittman, and A. Napolitano, "Using random undersampling to alleviate class imbalance on tweet sentiment data," in *IEEE International Conference on Information Reuse and Integration (IRI)*, 2015.

[15] J. D. Prusa, T. M. Khoshgoftaar, and D. J. Dittman, "Impact of feature selection techniques for tweet sentiment classification," in *Proceedings of the 28th International FLAIRS conference*, May 2015, pp. 299–304.

[16] J. D. Prusa, T. M. Khoshgoftaar, and A. Napolitano, "Using feature selection in combination with ensemble learning techniques to improve tweet sentiment classification performance," in *Proceedings of the 27th International Conference on Tools with Artificial Intelligence*, Nov 2015, pp. 186–193.

[17] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Nov 2001.

[18] S. Rosenthal, P. Nakov, A. Ritter, and V. Stoyanov, "Semeval-2014 task 9: Sentiment analysis in twitter," *Proc. SemEval*, 2014.

[19] H. Saif, Y. He, and H. Alani, "Alleviating data sparsity for twitter sentiment analysis," in *2nd Workshop on Making Sense of Microposts*. CEUR Workshop Proceedings (CEUR-WS. org), 2012.

[20] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012, pp. 115–120.

[21] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.

[22] H. Zhang, "The optimality of naive bayes," in *Proceedings of the FLAIRS Conference*, 2004.