Journal of Data Analysis and Information Processing, 2015, 3, 153-162
Published Online November 2015 in SciRes. http://www.scirp.org/journal/jdaip
http://dx.doi.org/10.4236/jdaip.2015.34015

Dimensionality Reduction of Distributed Vector Word Representations and Emoticon Stemming for Sentiment Analysis

Brian Dickinson, Michael Ganger, Wei Hu

Department of Computer Science, Houghton College, Houghton, NY, USA
Email: wei.hu@houghton.edu

Received 24 August 2015; accepted 20 November 2015; published 23 November 2015

Abstract

Social media platforms such as Twitter and the Internet Movie Database (IMDb) contain a vast amount of data which have applications in predictive sentiment analysis for movie sales, stock market fluctuations, brand opinion, or current events. Using a dataset taken from IMDb by Stanford, we identify some of the most significant phrases for identifying sentiment in a wide variety of movie reviews. Data from Twitter are especially attractive due to Twitter's real-time nature through its streaming API. Effectively analyzing this data in a streaming fashion requires efficient models, which may be improved by reducing the dimensionality of input vectors. One way this has been done in the past is by using emoticons; we propose a method for further reducing these features through identifying common structure in emoticons with similar sentiment. We also examine the gender distribution of emoticon usage, finding tendencies towards certain emoticons to be disproportionate between males and females. Despite the roughly equal gender distribution on Twitter, emoticon usage is predominately female. Furthermore, we find that distributed vector representations, such as those produced by Word2Vec, may be reduced through feature selection. This analysis was done on a manually labeled sample of 1000 tweets from a new dataset, the Large Emoticon Corpus, which consisted of about 8.5 million tweets containing emoticons and was collecting over a five day period in May 2015. Additionally, using the common structure of similar emoticons, we are able to characterize positive and negative emoticons using two regular expressions which account for over 90% of emoticon usage in the Large Emoticon Corpus.

Keywords

Natural Language, Emoticon, Twitter, Review

How to cite this paper: Dickinson, B., Ganger, M. and Hu, W. (2015) Dimensionality Reduction of Distributed Vector Word Representations and Emoticon Stemming for Sentiment Analysis. *Journal of Data Analysis and Information Processing*, **3**, 153-162. http://dx.doi.org/10.4236/jdaip.2015.34015

1. Introduction

In the past few years, the growth of social media platforms has brought with it a wealth of unprocessed data, containing interesting information that may be inferred from the interactions of users. The astronomical amount of information renders manual human analysis unfeasible, and has thus stimulated the development of machine learning algorithms to aid in gleaning information from this data. These algorithms may be running in either a batch processing environment, where a data set is stored and analyzed, or in a streaming environment, responding to changes in the source content. There are many different types of information that may be extracted from social media datasets; one type is that of sentiment. When a user makes a post, it often contains an opinion or feeling that he or she has for something; this opinion may be either explicit or implicit.

One of the most prominent social media platforms for sentiment analysis is Twitter. Over time, Twitter has become a vast source of data for different machine learning analyses, including natural language processing and sentiment classification. Enormous quantities of tweets are processed by Twitter every second, many of which contain users' feelings towards products, brands, events, media, and a wide array of other things. The main difficulty with using these tweets to extract sentiment is that they lack direct labels which convey the overall feeling or leaning of the text. This has led to many different ways to classify the tweets, involving both supervised and unsupervised machine learning. Analyzing Twitter using natural language processing also introduces interesting problems; each tweet is limited to a maximum of 140 characters, meaning that normal grammatical and syntactical conventions are mostly ignored or circumvented. The consequence of this is that models of vocabulary and sentence structure must be rebuilt according to the unique style which has developed on Twitter.

In addition to Twitter, another large source of sentiment information is the Internet Movie Database (IMDb). IMDb allows reviewers with an account to post reviews on different movies, which includes both text and a quantitative rating. The net result is a large data source for sentiment analysis, where each review has a manual label from the review's author according to the overall sentiment of the review; the advantage of such data is that the quality of these labels allows for isolation of prediction inaccuracy, as the labels themselves are already extremely accurate. The type of text in this data source is significantly different from the type in Twitter; the users are not limited to a certain character count, meaning that the text is more syntactically and contextually coherent. The natural consequence of this is that words with similar denotations or connotations generally occur in the same context, which oftentimes will improve the word model built from a dataset.

This paper starts with a discussion of relevant literature on Twitter analysis, and then moves into a discussion of the datasets used in this study, which are derived from data taken from either Twitter or IMDb. After that, the methods utilized to obtain an abstract representation of a body of text as a vector are discussed in detail, as well as operations on this representation. A section dictating the results of these operations is next, followed by concluding remarks and possible avenues of future research.

2. Related Works

There have been several studies demonstrating the value of data the can be obtained through Twitter. Sentiment analysis of Twitter data has interesting applications such as analyzing movie sales [1] or stock market fluctuations [2]. It may also be used to evaluate public opinion of a brand [3] or current events [4]. The prevalence of public accounts along with incredible volume of tweets and minute-by-minute time resolution through the streaming API has made Twitter a popular choice for mass opinion mining.

Another valuable feature of Twitter data is the popularity of user annotations on their posts. This includes both hashtags (e.g. #smile) and emoticons (e.g. =D). Previous studies have used emoticons as self-labels for labeling tweets based on sentiment [5] [6]. Additionally, many of these studies remove the emoticons after labeling the tweets, leaving the original tweet text without emoticons. However, without the emoticons, the text often does not contain definitive sentiment information. For example, compare the difference between the tweets "heading to be early :)" and "heading to bed early :(". These tweets are both very plausible and understandable and in full express a clear sentiment, however it would be impossible to determine which sentiment the author was feeling once the emoticon is stripped.

As previously stated, there are many benefits to be gained in treating Twitter data as a live stream, including real-time analysis of events, but there are also many challenges. Some of these include time restrictions, unbalanced classes, informal structure, and highly compressed meaning [5]. For these reasons streaming models have to be built to be robust against these challenges. First and most obvious of these is the restriction on processing

time from receipt of raw text to classification. This demands that algorithms perform only a single pass over incoming data. Severely unbalanced classes can cause certain algorithms such as SVMs to baseline their predictions with prior probability. This must be avoided to gain any useful insights from algorithms working in raw streams. Finally informal structure and condensed meaning which are common to all problems working with Twitter data can often be accounted for by using built in commonalities such as emoticons and hash tags.

3. Data

In this study, we used three different datasets, two of which contain tweets (Sentiment Analysis Dataset, and the Large Emoticon Corpus) and another which contains movie reviews (Stanford Movie Review Dataset). The Large Emoticon Corpus is a new dataset introduced with this study. For this reason, we discuss the collection and cleaning methods for this dataset in greater detail below.

3.1. Sentiment Analysis

The Twitter Sentiment Analysis Training Corpus consists of about 1.5 million labeled tweets, and is a compilation of the Sentiment 140 [6] dataset, the Sentiment Classification [7] dataset, and the Twitter Sentiment Corpus [8]. This dataset was labeled using emoticons contained within the tweets, classifying each tweet as either positive or negative. The emoticons were then stripped from the text to avoid circular reasoning (*i.e.* tweet is labeled positive due to emoticon, emoticon dominates classification). We later removed mentions from the tweets (e.g. @someusertag), as any classification enhancement they provided might be attributed to overfitting.

3.2. Large Emoticon Corpus

The Large Emoticon Corpus, introduced in this paper, is a dataset of tweets which contain one or more emoticons from a unique set of 115 emoticons. It contains roughly 8.5 million tweets, collected over the interval May 13 - 18, 2015 using the Twitter 4J interface to the Twitter Streaming API. For analysis, mentions and retweet tags were removed from the data. A sample of 1000 tweets from this dataset was manually labeled with binary sentiment corresponding to positive and negative emotion. Additionally, a subset of 500 tweets from this sample was manually labeled with user gender.

The tweets in the Large Emoticon Corpus have time and date information associated with them, allowing time-based analytics to be performed. One such inquiry, which we present later, is the frequency of a certain emoticon in tweet text over time. All times are in UTC, so they are not matched with local time, thus preventing time of day analysis. It does, however, open an opportunity for analysis of overall global emoticon usage over time. Additionally, a subset of about 38,000 tweets is tagged with geographic coordinates, which would enable time of day analysis on the subset.

3.3. Stanford Large Movie Review Dataset

The Large Movie Review Dataset from Stanford consists of fifty thousand labeled movie reviews, and an equal number of unlabeled reviews, taken from IMDB. The labeled reviews combine the text of movie reviews with their associated numerical rating. Reviews with a rating of seven or more out of ten are considered positive, while those with a rating of four or lower are considered negative. No more than 30 reviews are allowed for a given movie; additionally, a majority of the reviews are unprofessional. During analysis, stop words (taken from the NLTK Stopwords Corpus [9]) were removed from the text to prevent overfitting.

4. Textual Representation

The problem of representing natural language text in vector form for machine learning analysis has been around for decades, with early models using the Bag-of-Words technique. Textual representation can have a significant impact on the effectiveness of machine learning models as it determines what information is extracted from the original text. The goal is to retain as much pertinent information as possible. For this reason Bag-of-Words models, which lose all of the contextual and structural information of the text, are being replaced by distributed vector models such as Word2Vec and Sent2Vec; these are trained to understand the multifaceted relationships that exist between various words. The following sections highlight the different techniques we used to convert text into vectors.

4.1. Distributed Vector

Word2Vec [10]-[13] is an algorithm which converts individual words to n-dimensional vector representations. For a given text (tweet, sentence, or paragraph), the vectors of each of the words may be combined to produce a single vector which represents the text as a whole. The cosine distance between two vector representations roughly corresponds to the contextual similarity of the two texts [11], allowing these vectors to be used in various clustering and classification algorithms. To maximize the usefulness of word representations, a model must be built using text similar in structure and context to the target data. To generate models, Word2Vec uses contextual cues to determine the similarity between all of the different words in the training sample, meaning that the size and content of the training sample have a significant influence on the resulting vectors. Another distributed vector model is Sent2Vec [14]-[16] which like Word2Vec creates a vector representation of a text in low dimensional space. It uses the Deep Structured Sematic Model (DSSM) or the DSSM with convolutional-pooling structure (CDSSM) to form its vector representations.

4.2. Bag of Words

The Bag-of-Words model of a text relates the frequency of a word or phrase in the text to an element in the resulting representative vector. This leads to sparse vectors in high dimensional space. For this study, phrases with anywhere from 2 to 5 words were used to represent the text in the Stanford Movie Review Dataset. Unlike Word2Vec and Sent2Vec, the Bag-of-Words model does not take into account the similarity in meaning of closely related words. Likewise the n-gram representation of a text is an alternate vector representation which is based on the number occurrences of a “gram” (a sequence of characters). We selected the most commonly occurring grams from the dataset; these were then used to convert each text to a vector by counting the number of occurrences of each gram in the given text. Each position in the resulting vector corresponds to a gram selected from the original text.

4.3. Feature Selection

Feature selection (a type of dimensionality reduction) is a way to reduce the dimensionality of a vector representation of a dataset, removing redundant features and reducing over fitting. Generally, the classification accuracy remains constant or is improved as a result of this process for sparse vectors. The Correlation Based Feature Selection [17] algorithm was used to reduce the size of each of the vectors; this algorithm finds features that are highly correlated to training labels but are not correlated with each other. Another method of feature selection is the Chi2 algorithm [18]. This algorithm takes continuous inputs, and discretizes to determine features. The CFS algorithm was mainly used to perform feature selection; the Chi2 algorithm was used to rank the relative importance of each feature.

4.4. Gender

In addition to manually labeling the sentiment of 1000 tweets, about 500 of the sentiment-labeled tweets were also manually labeled with user gender. A survey of this data is presented in **Figure 1** and **Figure 2**. The data indicate that emoticon usage on Twitter is dominated by female users, although surveys suggest total user gender distribution is much more balanced [19] [20]. However, this is a relatively imperfect metric as Twitter does not collect gender information on its users, requiring the use of gender predictive tools to estimate this balance [21] [22]. Additionally, in **Figure 2** it is apparent that proportionally, the emoticons utilized by male users are more positive than those utilized by female users.

5. Results

5.1. Stanford IMDB

Representative vectors of the Movie Review dataset were constructed using Word2Vec, Sent2Vec (both CDSSM and DSSM), and Bag-of-Words models. Features were then selected from each of the vector sets to create new, reduced vector sets. Naive Bayes, Random Forest, and SVM algorithms were run on both the full and reduced vectors sets for each model. The results are shown in **Figure 3**.

Performing feature selection on the Bag-Of-Words model of yielded the phrases most correlated with review

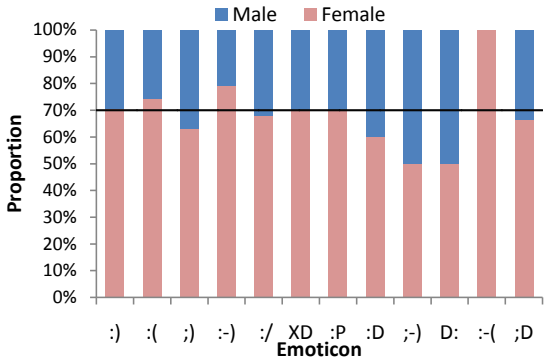

Figure 1. Emoticon usage by gender in a sample of about 500 tweets from the Large Emoticon Corpus manually labeled with user gender. The black line corresponds to the gender weighting of the sample. Note that the first few emoticons are significantly more frequent (**Figure 6**). While most of these emoticons are female-biased due to the sample, a few are used proportionally more often by males.

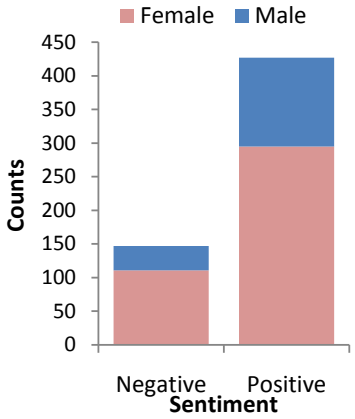

Figure 2. Positive and negative sentiment by user gender. This uses the same data as **Figure 1**. When males use emoticons, they use proportionally more positive emoticons than females.

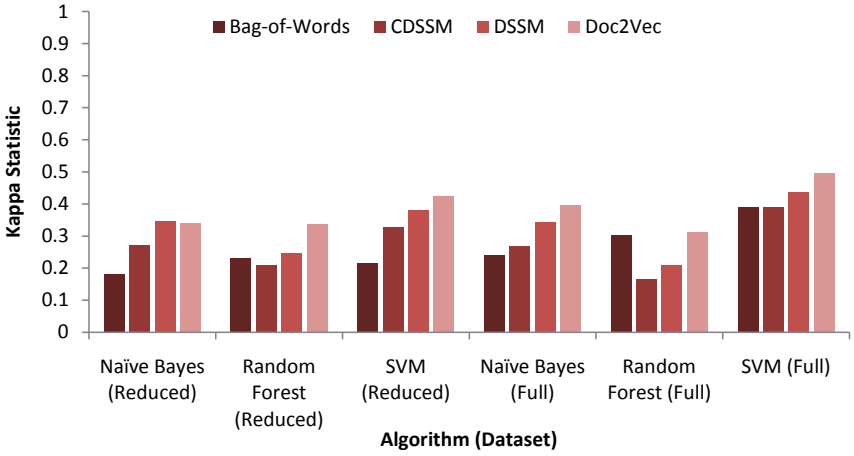

Figure 3. Prediction kappa statistic of different classification algorithms on movie review sentiment with different vector models. In most cases, the Word2Vec model outperformed the other models. Additionally, the Sent2Vec model outperformed the DSSM in all test cases. The Bag-of-Words model appears to be the least consistent of all the models.

sentiment. A selection of the top 20 of these is shown in **Table 1**. The majority of the phrases only contain 2 words, despite the fact that phrases with 2 to 5 words were used to build the Bag-of-Words; only 2 features from **Table 1** contain 3 words. This is likely due to the fact that many higher-complexity phrases (containing a structure of smaller phrases) may have the same meaning, dictated by at least one of these smaller phrases; thus, after feature reduction, the majority of selected phrases have 2 or 3 words.

5.2. Sentiment Analysis

The analysis of emoticon use over time shows small fluctuations over small periods of time (e.g. minutes or hours) but is relatively constant over longer periods of time (**Figure 4**). It also shows the frequency of use of the different emoticons. Emoticon use is dominated by the first few most frequent emoticons; the first four most frequent emoticons represent 86% of tweets with emoticons. This indicates that sentiment classification requires relatively few features to achieve reasonable accuracy when emoticons are included in the tweets. To show the significance of keeping emoticons within the tweets, classifying predictions were run on the vectors generated from emoticon-containing text as well as emoticon-absent text (**Figure 5**). Because of the potential for unbalanced class labels we use the kappa statistic rather than accuracy to evaluate the performance of each algorithm [5].

Table 1. Ranking of the top 20 features from the Bag-of-Words model. Stop words from the NLTK Stopwords Corpus were removed from the dataset prior to analysis.

Top 1 - 10	Top 11 - 20
worst movie	worst movie ever
one best	worst movies
one worst	highly recommended
highly recommend	bad film
bad movie	well worth
worst film	looks like
bad acting	fast forward
movie bad	one worst movies
really bad	acting bad
must see	piece crap

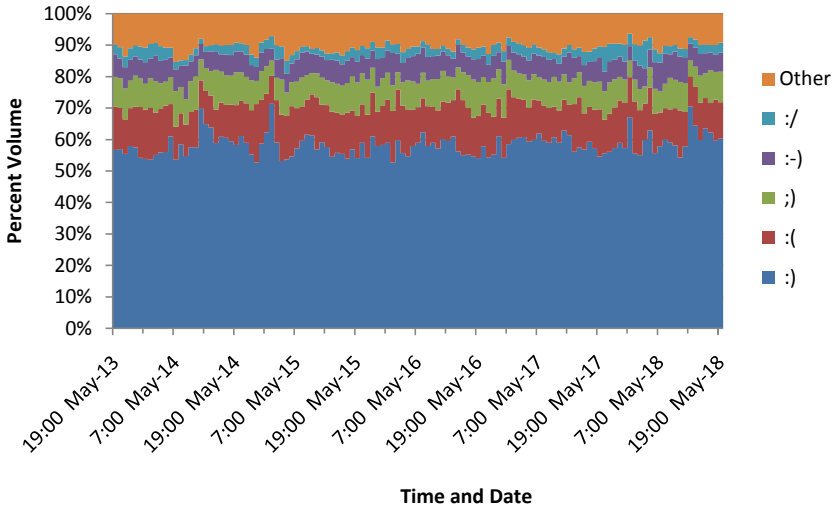

Figure 4. Emoticon usage by hour over a 5 day period from May 13, 2015 to May 18, 2015. The frequencies of each emoticon remain relatively constant over the collection period. The first five emoticons are the most dominate overall emoticon usage in the tweets.

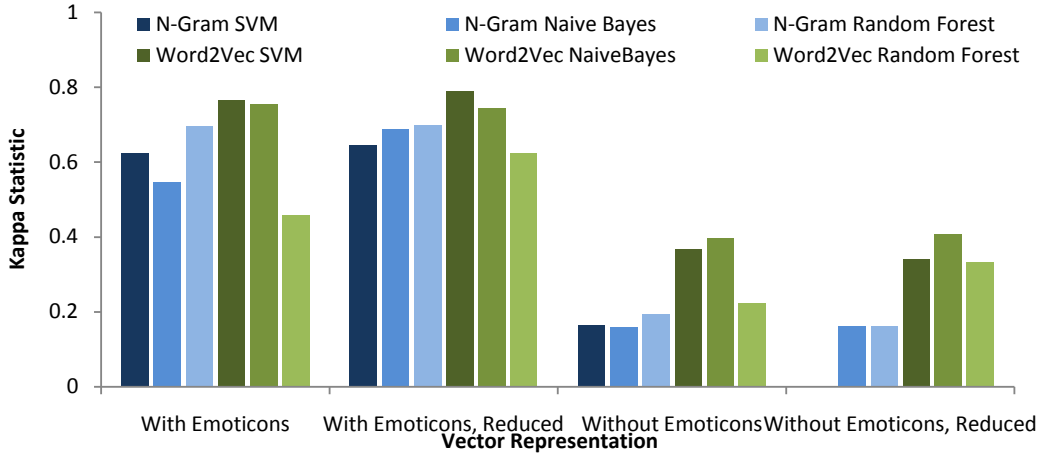

Figure 5. Chart showing comparison of classification algorithm kappa statistic on dataset with and without emoticons. The emoticons significantly affect the predictive ability of models built on the vectors produced from the tweets, especially in the case of N-Grams. Note that the SVM model prediction of the N-Gram reduced vectors had a Kappa Statistic of 0.

As shown in **Figure 5**, building models with emoticon-containing text and doing predictive analysis with these models is significantly more effective than using emoticon-absent text. In both representations (N-Gram and Word2Vec), the classifying algorithms had superior performance when emoticons were included. Additionally, the utility of using reduced vectors is apparent, as it stabilizes differences in algorithms. Also, the graph indicates that Word2Vec has a higher kappa statistic than N-Grams on the dataset without emoticons, indicating that the predictive ability of the N-Gram representation is highly dependent on the inclusion of emoticons.

In natural language processing, it is common to perform stemming on the input text. This operation usually involves removing the suffixes from different tenses of a word, leaving just the root [23] (e.g. “connect”, “connected”, and “connecting” all have the root “connect”). A similar process may be applicable to emoticons as well, due to structural similarities in emoticons which convey the same sentiment. For example, the emoticons:), :-), =), ;), and ;) all exhibit the root characteristic “)”, while :(, :-(, and = (all have the root characteristic “(” (see **Table 2**). While it is possible for positive emoticons to be backward (*i.e.* they contain “()”, the frequency of this occurring is orders of magnitude less than the forward emoticons (see **Figure 6**)).

The structural similarity between emoticons may be used to reduce the amount of processing required to predict the sentiment of a given tweet. For example, the input text may be preprocessed by replacing emoticons with their base element. This reduces the number of unique features which correlate to sentiment, further reducing the computational expense of classifying a given text. An advantage of using this approach is that a regular expression may be used to match positive or negative tweets; this may be computationally more efficient than storing a set of emoticons and comparing each word each word in the set.

5.3. Dimensionality Reduction of Distributed Vectors

In many cases, vectors produced using Word2Vec contain many irrelevant or redundant features, which may increase the computational expense of classification or clustering. They may additionally add noise, which decreases accuracy. One common way to reduce this is by performing feature reduction after calculating representative vectors for text. However, it may be inefficient to generate vectors with excessive dimensionality, only to reduce or discard features.

One way to remedy this is to build a model which minimizes dimensionality without sacrificing predictive quality. To test this theory, we built multiple models based on the Large Emoticon Corpus that cover the spectrum of dimensionality, and tested the predictive ability for sentiment given two datasets. Vectors were obtained using multiple Word2Vec models; labels for this dataset were obtained by using a Random Forest model [24] built on the manually labeled subset of 1000 tweets. One of the datasets contained about 100,000 tweets, with about 90% of the tweets being positive and the other 10% being negative; this was considered the unbalanced set. The balanced set contained a subset of about 20,000 of these tweets, but was chosen such that there was an

equal split of positive and negative tweets. Classifying was performed using an SVM model on the unbalanced and balanced datasets; additionally, a Naïve Bayes model was used to classify the unbalanced dataset for comparison. The results are shown in **Figure 7**.

Table 2. Positive and negative emoticons by root character. The regular expressions matching the positive and negative emoticons are shown in the final row.

Base	Positive Emoticons	Negative Emoticons
)	:) ;) :-) ;-) =)	
(	(:	:(:-(=(
/		:/ =/
D	XD :D ;D =D	
]	:] =]	
Regular Expression	**([;:\-=X]{1,2}[)\]D])|([(/D][;:\-=X]{1,2})**	**[;:\-=X]{1,2}[(/o]**

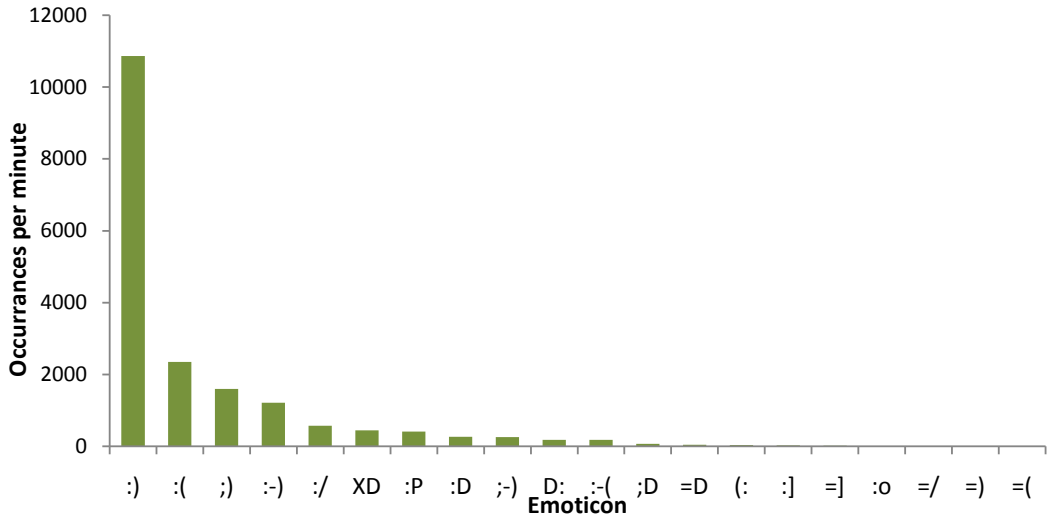

Figure 6. Emoticon usage in the large emoticon corpus. The counts are divided by total collection time in minutes to yield an approximate usage rate. The first few emoticons dominate total emoticon usage; the first emoticon occurs about 16,000 times more frequently than the last emoticon on the chart.

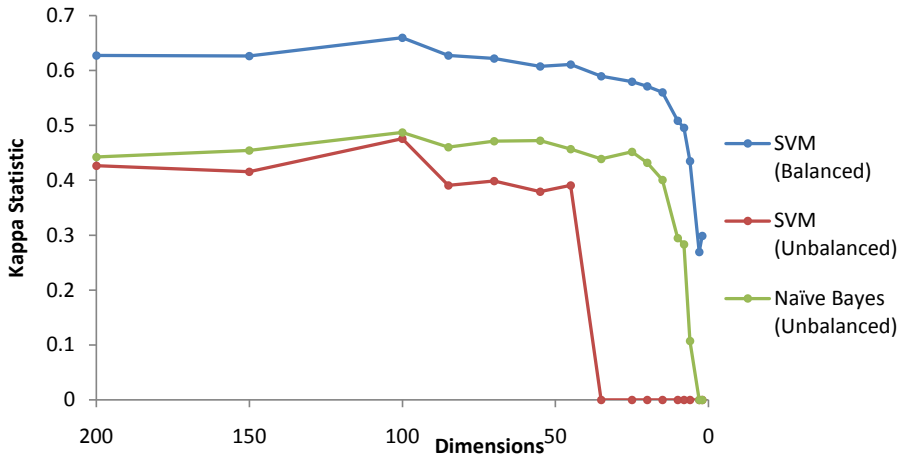

Figure 7. Kappa statistic of sentiment prediction on two samples of the large emotion datasets. The balanced sample has an equal proportion of positive and negative labels, while the unbalanced sample has 90% positive labels.

Depending on the data and model chosen, dimensionality can be significantly reduced without adversely affecting the predictive quality. If the dataset is balanced, it may be possible to reduce the dimensions further than if the dataset is biased toward one label. If the dataset is biased, an SVM model, which is designed to optimize accuracy, may be unstable with too few dimensions; it may begin to base its predictions entirely on prior probability. In this case, more features may be necessary. This depends on model type, though, as the Naïve Bayes model appears to be much more stable at high bias and low dimensions, for metrics other than accuracy.

6. Conclusions

This study introduced the Large Emoticon Corpus, which had provided insights into emoticon usage on Twitter, both in the frequency of individual emoticon use and usage based on user gender. We had also shown that positive and negative emoticons might be deconstructed into more basic forms and potentially used to classify positive and negative sentiment by simple regular expressions. Additionally, we found a bias in emoticon usage toward female Twitter users, despite the relatively balanced gender partitioning of the platform as a whole. Also, we ranked the most influential phrases in the Large Movie Review Dataset, finding that the smaller phrases could better represent a set of longer phrases with similar meaning. Finally, we found dimension reduction to be an effective technique on distributed vectors; feature selection might be used to reduce dimensionality of models such as Word2Vec in order to improve computational efficiency when performing real-time analysis with data streams.

A possible area for future research was deeper investigation into gender bias on Twitter with a larger dataset; by using a larger set of labeled gender data, it would be possible to determine to what extent male and female Twitter users utilized different emoticons. Another possible research topic was the optimization of dimension reduction for other labeled datasets, including sets with continuous inputs, compared to the binary inputs used in this study.

Acknowledgements

We would like to thank the Summer Research Institute at Houghton College for providing financial support for this study.

References

[1] Rui, H., Liu, Y. and Whinston, A. (2013) Whose and What Chatter Matters? The Effect of Tweets on Movie Sales. *Decision Support Systems*, **55**, 863-870.

[2] Bollen, J., Mao, H. and Zeng, X.-J. (2011) Twitter Mood Predicts. *Journal of Computer Science*, **2**, 1-8.

[3] Jansen, B.J., Zhang, M., Sobel, K. and Chowdury, A. (2009) Twitter Power: Tweets as Electronic Word of Mouth. *Journal of the American Society for Information Science and Technology*, **60**, 2169-2188.

[4] 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. *Proceedings of the* 50*th Annual Meeting of the Association for Computational Linguistics*, Jeju, 115-120.

[5] Bifet, A. and Frank, E. (2010) Sentiment Knowledge Discovery in Twitter Streaming Data. *Discovery Science*, 1-15.

[6] Go, A., Bhayani, R. and Huang, L. (2009) Twitter Sentiment. Stanford Digital Library Technologies Project.

[7] University of Michigan (2011) UMICH SI650—Sentiment Classification. https://inclass.kaggle.com/c/si650winter11/data

[8] Sanders, N.J. (2011) Sanders-Twitter Sentiment Corpus. *Sanders Analytics LLC*.

[9] Bird, S., Loper, E. and Klein, E. (2009) Natural Language Processing with Python. O'Reilly Media Inc.

[10] Le, Q. and Mikolov, T. (2014) Distributed Representations of Sentences and Documents. *CoRR*, vol. abs/1405.4053.

[11] Miklov, T., Chen, K., Corrado, G. and Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space. http://arxiv.org/abs/1301.3781

[12] Miklov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. (2013) Distributed Representations of Words and Phrases and Their Compositionality. In: *Advances in Neural Information Processing Systems*, Morgan Kaufmann Publishers Inc., San Francisco, 3111-3119.

[13] Mikolov, T., Yih, W.-T. and Zweig, G. (2013) Linguistic Regularities in Continuous Space Word Representations. HLT-NAACL, 746-751.

[14] Huang, P.S., He, X., Gao, J., Deng, L., Acero, A. and Heck, L. (2013) Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. *Proceedings of the* 22*nd ACM International Conference on Information & Knowledge Management*, San Francisco, 27 October-1 November 2013, 2333-2338. http://dx.doi.org/10.1145/2505515.2505665

[15] Shen, Y., He, X., Gao, J., Deng, L. and Mesnil, G. (2014) A Latent Semantic Model with Convolutional-Pooling. *Proceedings of the* 23*rd ACM International Conference on Conference on Information and Knowledge Management*, Shanghai, 3-7 November 2014, 101-110.

[16] Gao, J., Pantel, P., Gamon, M., He, X., Deng, L. and Shen, Y. (2014) Modeling Interestingness with Deep Neural Networks. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing*, Doha, 25-29 October 2014. http://dx.doi.org/10.3115/v1/D14-1002

[17] Hall, M.A. (1999) Correlation-Based Feature Selection for Machine Learning. PhD Dissertation, University of Waikato, Waikato.

[18] Liu, H. and Setiono, R. (1995) Chi2: Feature Selection and Discretization of Numeric Attributes. *Proceedings of the IEEE 7th International Conference on Tools with Artificial Intelligence*, Herndon, 5-8 November 1995, 388-391.

[19] Statistica (2015) Number of Active Twitter Users in the United States from 2010 to 2014, by Gender. http://www.statista.com/statistics/238715/number-of-active-twitter-users-in-the-united-states-by-gender/

[20] Beevolve (2012) An Exhaustive Study of Twitter Users across the World. http://www.beevolve.com/twitter-statistics/#a1

[21] Miller, Z., Dickinson, B. and Hu, W. (2012) Gender Prediction on Twitter Using Stream Algorithms with N-Gram Character Features. *International Journal of Intelligence Science*, **2**, 143-148. http://dx.doi.org/10.4236/ijis.2012.224019

[22] Dietrick, W., Miller, Z., Valyou, B., Dickinson, B., Munson, T. and Hu, W. (2012) Gender Identification on Twitter Using the Modified Balanced Winnow. *Communications and Network*, **4**, 189-195. http://dx.doi.org/10.4236/cn.2012.43023

[23] Porter, M.F. (1980) An Algorithm for Suffix Stripping. *Program*, **14**, 130-137. http://dx.doi.org/10.1108/eb046814

[24] Pedregosa, F., *et al.* (2011) Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830.