Neural Networks for Sentiment Analysis on Twitter

Brett Duncan and Yanqing Zhang
Department of Computer Science, Georgia State University
Atlanta, Georgia 30302-5060, USA
E-mail: bduncan7@student.gsu.edu, yzhang@gsu.edu

***Abstract*—The online medium has become a significant way that people express their opinions online. Sentiment analysis can be used to find out the polarity of an opinion, such as positive, negative, or neutral. Sentiment analysis has applications such as companies getting their customer's opinions on their products, political sentiment analysis, or opinions on movie reviews. Recent research has involved looking at text from online blogs, tweets, online movie reviews, etc. to try and classify the text as being positive, negative, or neutral. For this research, a feedforward neural network will be experimented with for sentiment analysis of tweets. The training set of tweets are collected using the Twitter API using positive and negative keywords. The testing set of tweets are collected using the same positive and negative keywords.**

***Keywords*—sentiment analysis, feedforward pattern network, text classification**

I. Introduction

The online medium has become a significant way for people to express their opinions [1], [2], and with social media, there is an abundance of opinion information available. Using sentiment analysis, the polarity of opinions can be found, such as positive, negative, or neutral by analyzing the text of the opinion. [3], [4]. Sentiment analysis has been useful for companies to get their customer's opinions on their products [4], [5], [2], predicting outcomes of elections [6], and getting opinions from movie reviews [3], [4]. The information gained from sentiment analysis is useful for companies making future decisions [2], [5].

Many traditional approaches in sentiment analysis uses the bag of words method [7], [3], [4]. The bag of words technique does not consider language morphology, and it could incorrectly classify two phrases of having the same meaning because it could have the same bag of words [3]. The relationship between the collection of words is considered instead of the relationship between individual words [4]. When determining the overall sentiment, the sentiment of each word is determined and combined using a function [4]. Bag of words also ignores word order, which leads to phrases with negation in them to be incorrectly classified [7].

Other techniques discussed in sentiment analysis include Naive Bayes, Maximum Entropy, and Support Vector Machines [4], [8]. In the related work section, approaches used for sentiment analysis and text classification are summarized.

II. Related Work

A. Related Text Classification Techniques

In [9], two strategies are discussed for using neural networks for text classification. In the traditional method, documents are encoded into numerical vectors, and in a novel method, string vectors are used. In the traditional method, documents are concatenated into a long string. The strings are tokenized by white space and punctuation. Then, each token is stemmed into its root form: verbs in past tense are converted to root form, and nouns in plural form are translated into singular form. Then, stop words are removed. The remaining words are called the feature candidates. The number of features are usually too large, so feature selection methods are used to select a subset of the features. Even after feature selection, the dimension of the numerical vectors can still be large. This leads to a high cost time for processing and a decrease in categorization performance. In the novel method of [9], documents are encoded into string vectors instead of numerical vectors. A d dimensional string vector would contain d words in descending order based on their frequencies in the text. This string vector is used as input into a novel neural network.

Tokenization, stemming, and stop word removal are common preprocessing steps used in text classification [9], [10], [11]. When using various classification algorithms for text classification, there are commonly a large amount of features due to there being many words in documents. Tokenization is the process of extracting words from the text to be classified. Stemming is reducing words that are not in their root form to their root form. Stop word removal is the removal of common functional words in order to improve performance of the classification. Even after these preprocessing steps, the dimensionality of the features can still be large.

Four feature selection techniques were used in [10]: the document frequency (DF) method, the category frequency-document frequency (CF-DF) method, the term occurrence frequency inverse document frequency (TFxIDF method), and the principal component analysis method. Through experiments, it was found that principal component analysis was most effective.

Besides feedforward neural networks, other kinds of neural networks called the self-organizing map (SOM) and the growing hierarchical self organizing map (GHSOM) were used in [11]. The reason for using the SOM is that it works well for mapping high-dimensional data into a two-dimensional representation space.

B. Sentiment Analysis Techniques

One approach to sentiment analysis is a recurrent neural network [3]. It has the advantage over traditional neural networks in that it gets better performance on structured data prediction on variable input. In the recurrent neural network architecture, the input layer contains the bag of words feature vector at time t. The input layer is connected to the hidden layer which contains the history of information. The hidden

Proc. 2015 IEEE 14th Int'l Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'15)
N. Ge, J. Lu, Y. Wang, N. Howard, P. Chen, X. Tao, B. Zhang, & L.A. Zadeh (Eds.)
978-1-4673-7290-9/15/$31.00 ©2015 IEEE

layer has recursive connections to itself, and also connections to the output layer. The hidden and output layers also feature neurons to store values at a time t. The recursion allows the network to be deeper than a traditional neural network [3]. Rong et al. [3] proposed a semi-supervised dual recurrent neural network for their sentiment analysis. It is similar to a traditional recurrent neural network in that it can use time to cover a longer history memory. It is different in that the output layer also has recursive connections back to itself for better sentimental analysis.

In Yao et al. [12], a recursive neural network is used for language understanding. In language understanding, the main goal is to label words that have semantic meaning [12]. The network is trained using semantic labels rather than the words themselves. The language is modeled by using the output as the input shifted in time so that the next word is predicted.

In [4], sentiment analysis techniques are discussed specific to Twitter. Focus was placed on the electronic products domain. Twitter sentiment analysis is different from traditional sentiment analysis in that more slang and misspelled words are used due to the 140 character limit. Because of this, preprocessing is required before extracting features. Preprocessing is done in two steps. First, Twitter features such as hashtags and emoticons are extracted. Emoticons are given positive or negative scores based on being positive or negative. Hashtags are also given positive or negative scores. After Twitter specific features were removed, a unigram approach is used and the plain text of the tweet is represented as a collection of words. Positive and negative tweets are used in the feature vector. Other classification techniques for text classification that are discussed in [4] include Naive Bayes, a Support Vector Machine, and a Maximum Entropy classifier. The Naive Bayes classifier considers all features in the feature vector as independent of each other. The Support Vector Machine uses a hyper plane to separate tweets into classes. The Maximum Entropy Classifier does not assume relationships between features, and in the feature vector, the relationships between the part of speech tag, emotional keyword, and negation can be utilized in the classification process, unlike in the Naive Bayes Classifier.

Anjaria et al. also did sentiment analysis on Twitter with a focus on trying to predict election outcomes [8]. It is mentioned that n-grams and Part-of-speech tagging techniques are used in Twitter sentiment analysis. As part of the prediction method, retweets that each party generates are factored in. In this paper, Naive Bayes, Support Vector Machine, Maximum Entropy, and a feed forward neural network trained by back-propagation were used for predicting elections with good accuracy [8].

An event driven neural network was used in [1] for evaluating internet user's response to given events. A Mood engine is built from multiple artificial neural networks, one for each mood, and different sets for different cultures. The ANNs are trained using words that describe the dictionary definition of moods. Messages are collected from blogs and social networks.

Socher and Perelygin [7] created a recursive neural tensor network that tries to better understand the semantics of a group of words. It was created to better understand short messages, commonly found on Twitter. Compared with standard recursive neural networks, matrix-vector recursive neural networks, Naive Bayes, and support vector machines, the recursive neural tensor network showed the most accuracy.

III. Method

In this paper, a focus will be placed on neural networks for sentiment analysis on Twitter. The neural network used will be the feedforward pattern network from the MATLAB Neural Network Toolbox. The tokenization, word stemming, and stop word removal preprocessing steps that are used for document categorization will be used before passing the data to the neural network.

A. Preprocessing

Before tweets are passed to the neural network for training, preprocessing is done before feeding it to the neural network. The following items from tweets are removed:

- Punctuation and symbols
- Mentions to other users, in which the @ symbol is followed by the user name
- Single characters
- Stop words like and, to, the, etc.

Next, the remaining words are stemmed using Porter's Stemming algorithm [13]. For example, the words waits, waiting, or waited would become wait. This will reduce the feature space. Words for each tweet are then added to a 2D list structure, with each column containing the words from each tweet. After the words go through the full preprocessing, the result is shown in Fig. 1.

da	tweet	tweetl
vinci	tee	happi
code	awesom	bdai
book	crowdsour	dadda
just	shirt	happi
awesom	threadless	mami
		dai

Fig. 1. Tweets in 2D List Structure

B. Creating Vocabulary List

As the words from each tweet are being added to the 2D list structure, each unique word is added to a vocabulary list. The vocabulary is then sorted in alphabetical order. The tweets from the 2D list in Fig. 1. would produce the vocabulary list shown in Fig. 2.

C. Loading Training Labels

Training labels for positive or negative are stored in a 2 $\times$ (number of tweets) matrix. A 1 on the first row represents a negative tweet, and a 1 on the second row represents a positive tweet. Fig. 3 shows an example of this structure.

awesom
bdai
book
code
crowdsourc
da
dadda
dai
happi
just
mami
shirt
tee
threadless
tweet
tweetl
vinci

Fig. 2. Tweets in 2D List Structure

0	0	0	1	1	1	1
1	1	1	0	0	0	0

Fig. 3. Training labels; Each column represents a tweet. A 1 on the top row represents a negative tweet, and a 1 on the bottom row represents a positive tweet.

D. Map Variable

A map variable is created that maps a word from the vocabulary (key) to the index it is located in the vocabulary (value). For example, if the vocabulary from Fig. 2 were used, the string "awesom" would map to the first location, and the string "happi" would map to the ninth location. This map variable will be used when creating the numerical training vectors for the neural network. When creating the numerical vector, a 1 is placed for words that appeared in the tweet, and a 0 for words that did not appear.

E. Numerical Training Vector

A numerical vector of size (number of vocabulary words) $\times$ (number of tweets) is created that will be fed to the neural network for training. The number of rows is the same as the amount of unique vocabulary words collected during preprocessing, so if that word was present in a tweet, a 1 is placed. If not, a 0 is placed. The map variable is used to determine which row a 1 will be placed. The tweets from Fig. 1 would produce the input vector in Fig. 4, for example.

F. Training the Neural Network

The neural network is now trained using the numerical input vector shown in Fig. 4, and the training labels shown in Fig. 3. The neural network used is the pattern recognition network provided by MATLAB, which is a feedforward network that uses the scaled conjugate gradient method for adjusting the weights and biases. The number of inputs to the neural network varies depending on the number of vocabulary words. One hidden layer of 10 neurons was used for all experiments. The

da	tweet	tweetl
vinci	tee	happi
code	awesom	bdai
book	crowdsour	dadda
just	shirt	happi
awesom	threadless	mami
		dai

1	1	0
0	0	1
1	0	0
1	0	0
0	1	0
1	0	0
0	0	1
0	0	1
0	0	1
1	0	0
0	0	1
0	1	0
0	1	0
0	1	0
0	1	0
0	0	1
1	0	0

Fig. 4. Tweets in 2D List Structure & the Numerical Vector Equivalent

output layer consists of two neurons, one neuron representing the probability of the tweet being positive, and the other the probability of the tweet being negative.

G. Collecting Test Tweets

Next, tweets for testing are loaded into a 2D list structure similar to in Fig. 1. Like the training tweets, the test tweets also go through some preprocessing. The following are removed from the tweets:

- Punctuation and symbols
- Mentions to other users, in which the @ symbol is followed by the user name
- Single characters
- Stop words like and, to, the, etc.

Each word is also stemmed using Porter's stemming algorithm.

H. Numerical Test Vector

A numerical vector of size (number of vocabulary words) $\times$ (number of test tweets) is created similar to the numerical training vector shown in Fig. 4. The map variable is used to determine which row a 1 will be placed.

IV. Results

Tweets were collected from the University of Michigan's Sentiment Analysis competition [14]. A total of 200 tweets were used from the set, 100 being positive, 100 being negative. Those 200 tweets created a vocabulary of 288, making the number of inputs to the neural network 288. In the experiment, a test set of 100 tweets that has a similar vocabulary to the original training set is used. The tweets in the test set were labeled as postive or negative manually. One hundred feedforward neural networks were created and trained on the training set and tested on the training set. The average accuracy (number of correctly classified tweets divided by the number of incorrectly classified tweets) was 74.15%.

In another experiment, positive and negative tweets are collected using the Twitter API [15]. Positive tweets were collected containing the keywords *amazing*, *thankful*, *love*, and *outstanding*. Negative tweets were collected containing the keywords *disappointed*, *terrible*, *horrible*, and *disaster*. A

training set and a testing set of the tweets were collected. The majority of the tweets in the testing set were classified correctly. The output neurons would output results close to 100% positive or 100% negative if a tweet only contained positive or only negative words. If a tweet contained a positive and a negative word, the output neurons would be closer to 50% indicating uncertainty, and the tweet could be classified incorrectly.

Limited memory became an issue if significantly more than 200 tweets are used for training. More tweets result in more vocabulary words, and if there were too many vocabulary words, there was not enough memory to create the data structures needed for training the neural network. A second experiment was done on the same 200 tweets, but this time using principal component analysis to reduce the number of features for the training set and the test set. The number of features was reduced to 50, but experiments showed that the accuracy was not as good as using all the original features. The average accuracy on the test set of 100 tweets was only 31.04%.

V. Conclusion

The classification of positive and negative tweets saw somewhat satisfactory results if the number of tweets used for training was not too large. If 200 training tweets were used, the vocabulary was not too large and the average accuaracy was 74.15%. Memory was an issue when training the feedforward pattern network if the vocabulary became too large. There was not enough memory to hold the data structures needed for training the feedforward pattern network once the vocabulary became too large from using more training tweets. Although it was shown in [10] that principal component analysis was effective in a text categorization task, the principal component analysis applied for this experiment did not see effective results.

VI. Future Work

As memory was an issue when training the feedforward pattern network, more feature reduction techniques can be experimented with in the future to reduce the size of the vocabulary and input vector to the neural network.

Besides feature reduction on the feedforward pattern network, other types of neural networks could be experimented with to solve the memory issue. Recurrent neural networks [3], [12], self organizing maps [11], and recursive deep neural networks [7] are some other neural networks used.

Emoticons can be considered as having an effect on the sentiment value. The method used in the experiments currently removes single characters and symbols, discarding of emoticons. Text based regular emoticons such as :) or :(can be extracted, as well as dedicated emoticon symbols.

Tweets can have repeated letters in words, such as "awesomeeeee", which would cause this word to be considered different from just "awesome". Another preprocessing step could be to reduce words containing repeated letters to their root word.

Twitter hashtags tend to have multiple words not separated by spaces. A preprocessing step could be added to separate words contained in hastags and add them to the vocabulary.

The sentiment analysis experiments were performed on limited tweets of no more than 300, due to memory limitations. Another work that can be performed in the future would be to use a high performance computer allowing higher scaled experiments.

Acknowledgements. This research is supported by the NSF REU grant # 1156733.

References

[1] S. Fong, S. Deb, I.-W. Chan, and P. Vijayakumar, "An Event Driven Neural Network System for Evaluating Public Moods From Online Users' Comments," in *Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the*, Feb 2014, pp. 239–243.

[2] W. Wang, "Sentiment Analysis of Online Product Reviews With Semi-supervised Topic Sentiment Mixture Model," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, vol. 5, Aug 2010, pp. 2385–2389.

[3] W. Rong, B. Peng, Y. Ouyang, C. Li, and Z. Xiong, "Semi-supervised Dual Recurrent Neural Network for Sentiment Analysis," in *Dependable, Autonomic and Secure Computing (DASC), 2013 IEEE 11th International Conference on*, Dec 2013, pp. 438–445.

[4] M. Neethu and R. Rajasree, "Sentiment Analysis in Twitter Using Machine Learning Techniques," in *Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on*, July 2013, pp. 1–5.

[5] X. Zhou, X. Tao, J. Yong, and Z. Yang, "Sentiment Analysis on Tweets for Social Events," in *Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on*, June 2013, pp. 557–562.

[6] A. Bakliwal, J. Foster, J. van der Puil, R. O'Brien, L. Tounsi, and M. Hughes, "Sentiment Analysis of Political Tweets: Towards an Accurate Classifier," in *Proceedings of the Workshop on Language Analysis in Social Media*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 49–58. [Online]. Available: http://www.aclweb.org/anthology/W13-1106

[7] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," Aug 2013.

[8] M. Anjaria and R. Guddeti, "Influence Factor Based Opinion Mining of Twitter Data Using Supervised Learning," in *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*, Jan 2014, pp. 1–8.

[9] T. Jo, "NTC (Neural Text Categorizer): Neural Network for Text Categorization," Ontario, Canada, April 2010.

[10] S. Lam and D. Lee, "Feature reduction for neural network based text categorization," in *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, 1999, pp. 195–202.

[11] N. N. Pise, "Document Classification using Neural Networks," Pune, India, 2005. [Online]. Available: http://www.niitcrcs.com/iccs/papers/2005_99.pdf

[12] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent Neural Networks for Language Understanding." Interspeech, August 2013. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=200236

[13] M. F. Porter, "Readings in Information Retrieval," K. Sparck Jones and P. Willett, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, ch. An Algorithm for Suffix Stripping, pp. 313–316. [Online]. Available: http://dl.acm.org/citation.cfm?id=275537.275705

[14] "UMICH SI650 - Sentiment Classification," March 2011. [Online]. Available: http://inclass.kaggle.com/c/si650winter11

[15] "API Overview | Twitter Developers." [Online]. Available: https://dev.twitter.com/overview/api