IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 7, No. 3, September 2018, pp. 119~124
ISSN: 2252-8938, DOI: 10.11591/ijai.v7.i3.pp119-124

Analysing Event-Related Sentiments on Social Media with Neural Networks

P. Santhi Priya[1], T. Venkate swara Rao[2]
[1]Department of Computer Science and Engineering, Sri Sairam College of Engineering, Benguluru, India
[2]Computer Science and Engineering, PVP Siddhartha Engineering College, India

Article Info

Article history:

Received Mar 11, 2018
Revised May 10, 2018
Accepted Jun 20, 2018

Keyword:

Doc2Vec
EU referendum
Multi-layer perceptrons
Neural networks
Paragraph vectors
Sentiment analysis

ABSTRACT

Sentiment analysis is performed to determine the polarity of opinion on a subject. It has been applied to text corpora such as movie reviews, financial documents to glean information about overall-sentiment anc produce actionable data. Recent events have demonstrated that polling can be sometimes unreliable. People can be difficult to access through conventional polling methods and less than frank in polls. In the era of social media, voters are likely to more freely express their opinion on social media forums about divisive events especially in media where anonymity exists. Analyzing the prevailing opinion on these forums can indicate if there are any deficiencies in polling and can be a valuable addition to conventional polling. We analyzed text corpora from Reddit forums discussing the recent referendum in Britain to exit from the EU (known as Brexit). Brexit was an important world event and was very divisive in the run-up and post vote. We analyzed sentiment in two ways: Initially we tried to gauge positive, negative, and neutral sentiments. In the second analysis, we further split these sentiments into six different polarities based on the directionality of the positive and negative sentiments (for or against Brexit). Our technique utlilized paragraph vectors (Doc2Vec) to construct feature vectors for sentiment analysis with a Multilayer Perceptron classifier. We found that the second analysis yielded overall better results; although, our classifier didn't perform as well in classifying positive sentiments. We demonstrate that it is possible glean valuable information from complicated and diverse corpora such as multi-paragraph comments from reddit with sentiment analysis.

Corresponding Author:

P. Santhi Priya,
Department of Computer Science and Engineering,
Sri Sairam College of Engineering, Benguluru, India.
Email: shantipriya.p@gmail.com

1. INTRODUCTION

Sentiment analysis is a quantitative automated opinion mining technique which attempts to quantify the sentiment (positive/negative/neutral) attached to text content. Machine learning approaches [1] look for words or phrases that denote the sentiment polarity of the text. A training set of examples is used to assign a polarity and weight to words which can later be used to predict the sentiment of a wider set not included in the training set. Training can be supervised by explicit labelling of polarity manually or by implicity determining polarity from features such as emojis. Sentiment analysis and prediction have been employed in various domains successfully. Sentiment prediction has been used to generate stock recommendations based on reactions to news and blogs [5], assign emotional polarity to news articles [6] and predict the reaction of specific audiences/communities to news [11].

Markets all over the world respond to major events not only post the event but also in the run-up to events. Prices in stock markets, real-estate markets and currency markets all fluctuate according to perceived

outcome of events. The ability to predict how these events will pan out by gauging sentiment would be an immensely useful tool for taking profitable positions in these markets. Polls don't always reflect the sentiment accurately since people can be difficult to access or less than frank in polls [7]; therefore, it is worthwhile investigating if social media and especially, anonymous social media can be a good predictor of sentiment. Anonymity affords a degree of protection to the commenter so that they can express their opinion more freely. This is especially relevant when the event is highly divisive and a high social cost is attached to expressing one's opinion frankly.

Reddit is an entertainment, news and general purpose social networking site where members of the community submit either direct link to websites or text posts. Reddit is organised by topics of interest into sub-communities called subreddits.The members posting in a sub-reddit are only identified by their usernames therefore affording a degree of anonymity on Reddit. Other members of the community can comment on these posts. These comments are then up-voted or down-voted reflecting the views or the sentiments of the members of the community. The comment votes can be used to measure the sentiment in that particular sub-community which may differ from sentiments in other subreddits. Reddit comments can be multi-paragraph and can also contain links, emojis and images. The complex nature of these comments containing different media, rhetorical flourishes and sarcasm can be challenging for sentiment analysis.

In this paper, we performed fine-grained sentiment analysis to determine if we could classify sentiment related to a particular event. We tried to determine if commenters viewed a certain event positively or negatively and sampled from multiple sub-reddits. For the event, we chose the referendum conducted in Britain as to whether Britain should leave the European union (known as Brexit) as this was a major historical event with ample discussion before and after the event. This afforded us the possibility to collect and analyze large amounts of data

2. DATASET

In this paper, we extracted comments from posts in two specific subreddits concentrated on the United Kingom- r/unitedkingdom and r/ukpolitics. These two subreddits have a large number of subscribers- 134,917 and 56,493 respectively. We first filtered posts that had the keywords "brexit" or "referendum" in the title. Posts with timestamps ranging from June 1, 2016-July 31, 2016 were included in the study. This time period covered the pre and post referendum (conducted on June 23, 2016) period. All posts and comments were extracted using the Reddit API platform implemented in Python 2.7.

124,788 comments were extracted totally from the posts across all epoch of the two subreddits. We picked a subset of approximately 30% of the comments for the training and test set (24,423 comments). We divided the comments into five roughly 7-15 day epochs: 1-17 June, 17-23 June, 24 June-1 July, 1-16 July and 16-31 July. The comments were selected randomly from each epoch in proportion to the total number of comments using stratified random sampling. This was done to ensure that we didn't accidently overly sample from a certain time period. We also selected posts from other subreddits such as r/worldnews and r/politics that had "Brexit" or "EU referendum" posted within the timespan under consideration (1545 comments) The comments from these posts were extracted and added to the subset. These comments were then annotated by two human annotaters. We only kept those annotations on which both the annotaters agreed.

The data was preprocessed by removing extra whitespaces, removing all punctuation and all text was converted to lower case.

We then performed analysis on two levels. The training set was annotated on one level as positive, negative and neutral. These polarities were then further split into six different sentiment polarities: positive towards leaving the EU (PL), positive towards remaining in the EU (PR), negative towards leaving the EU (NR), negative towards staying in the EU (NL), negative towards remain supporters (NLR) and neutral.

After removing blank paragraphs and so on, we ended up with 20750 comments in the training set and 5187 comments in the test set for the first analysis after a 80:20 split of the original set and 20774 comments in the training set and 5162 comments in test set for the second analysis. The difference in the number of comments is due to the fact that the analysis was run separately on the two sets because the picks were randomized and the ratio splits varied slightly.

3. ALGORITHM

Machine learning algorithms for applications like sentiment analysis require text to be transformed into a fixed-length vector suitable for processing. The common bag-of-words (BoW) approach which represents text/documents in terms of a vector of word/token frequency has some drawbacks-word order is lost and some of the semantics associated with the word in text is also lost. Vector representations of words

have attempted to remedy some of these drawbacks. In Word2Vec, vector representations are computed for each word. Its input is a text corpus and its output is a set of feature vectors for words in that corpus.

Word2vec is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus. In Word2Vec, a distributed representation of a word is used. Given a feature vector with several hundred dimensions, each word is represented by a distribution of weights across those elements and each element in the vector contributes to the definition of many words. The neural network based word vectors are usually trained using stochastic gradient descent where the gradient is obtained via back propagation.

A more refined approach involves paragraph vectors which work at the level of sentences or documents. Training word vectors occurs as normal, except that an additional vector representing the paragraph is added to the task whenever the sampled window comes from that paragraph. In the Paragraph

Vector(Doc2Vec) [9] framework, every paragraph is mapped to a unique vector and every word is also mapped to a unique vector like in Word2Vec. Both the paragraph vectors and word vectors are trained using stochastic gradient descent and the gradient is obtained via backpropagation. The obtained feature vectors can be used as inputs to conventional machine learning algorithms such as classifiers, etc. Paragraph vectors inherit an important property of the word vectors: the semantics of the words.i.e.it groups the vectors of similar words together in vectorspace. The second advantage of the paragraph vectors is that they take into consideration the word order and retain some information about the context. We considered this approach to constructing feature vectors most suitable for our analysis and transformed our corpus into feature vectors using the gensim implementation of the Doc2Vec algorithm [4]. After transformation, each comment was represented by a fixed-length feature vector with a dimensionality of 100.

4. METHOD

The feature vectors obtained from the Doc2Vec process were standardized by removing the mean and scaling to unit variance. A Multi Layer Perceptron Classifier (MLP) [10] was then trained on these scaled and centered feature vectors of the training set. We found the best parameters through a gridsearch algorithm which performs exhaustive searches over specified parameters. A k-fold (5-fold) cross validation approach was used to tune the classifier and pick parameters which yielded the best accuracy.

We then ran an evaluation (test) set through the optimized classifier to evaluate the performance of the classifier on hitherto unseen data. We tested different algorithms and activation functions for the MLP classifier, for example, L-BFGS, Adam and Stochastic gradient descent (SGD). For the first case of assigning three sentiment polarities, we obtained the best results with the L-BFGS algorithm [2] and a rectified linear unit activation function. For the second case, we obtained the best results with the Adam algorithm and a hyperbolic tan activation function [8].

Upon annotation into the different polarities, we discovered that our dataset had more examples of negative and neutral sentiments than positive sentiments. For the first analysis type, we obtained the training and test set by picking proportionately from different classes using stratified sampling. [10]. In the second type of analysis where we were classifying the data into six different classes, the NR and negative polarity was undersampled to 70% of the class in order to balance the training set and not bias the classifier towards negative and neutral sentiments.

5. RESULTS

For our first analysis, we attempted classification into three sentiment polarities: positive, negative and neutral. We tuned our parameters by conducting a grid search within a cross-validation loop using the nested cross-validation paradigm [3]. The hyperparameters were tuned in the inner cross-validation loop (stratified K-fold with 3 splits) while the generalization error was measured over several dataset splits in the outer cross-validation loop (stratified K-fold with 3 splits). The parameters that were varied were the number of neurons in the hidden layer and the L2-regularization parameter (alpha) and we used F1- weighted score in order to evaluate the performance. Our hidden layer parameters were: 50, 100 or 200 neurons in a single hidden layer; and our alpha parameters were: 0.0001, 0.001, 0.01, 0.1. The cross validation process yielded a mean F1-score of 0.518 (this corresponded to a mean classification accuracy of 60% estimated in a separate run with the same training set) and a combination of hidden layer parameter of 50 neurons and an alpha of 0.0001 yielded the best mean F1-score of 0.527 as shown in Figure 1.

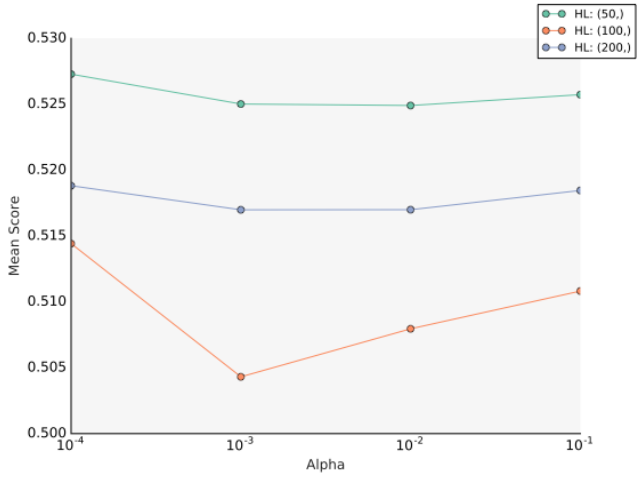

Figure 1. Results of Grid-Search Algorithm for Analysis 1. Mean F1-score vs L2-Regularization; Legend: Hidden Layer (HL) Number of Neurons

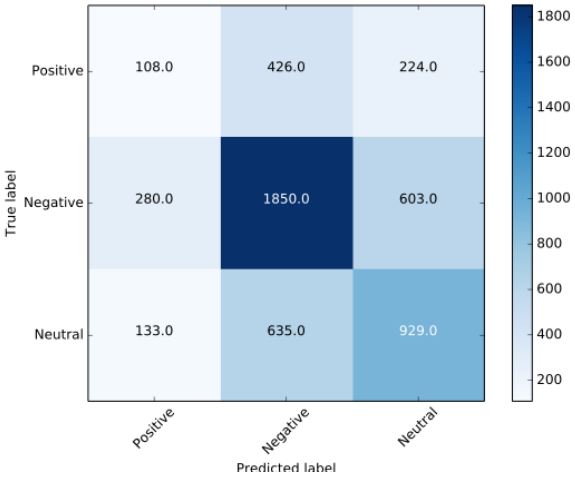

Figure 2.a) Confusion Matrix for Analysis

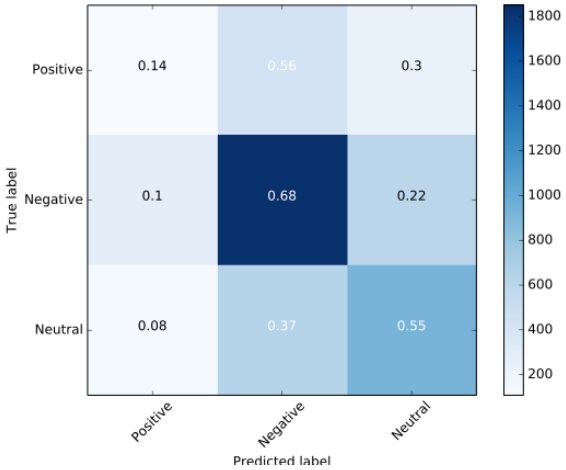

Figure 2.b) Normalized Confusion Matrix for Analysis

Figure 2. Confusion Matrices for Analysis 1

We then ran the test set through the optimized, tuned classifer and obtained an accuracy of 55.6%. A confusion matrix showed that our classifier was especially prone to misclassifying positive samples as shown in Figure 2. We then sought to improve our model by further splitting the sentiment polarities. The sentiment polarities were split into: six different sentiment polarities: positive towards leaving the EU (PL), positive towards remaining in the EU (PR), negative towards leaving the EU (NR), negative towards staying in the EU (NL), negative towards remain supporters (NLR) and neutral as described previously. We optimized the classifer using the same parameter space described above. The cross validation process yielded a mean F1-score of 0.549 (this corresponded to a mean classification accuracy of 61.6% estimated in a separate run with the same training set) and a combination of hidden layer parameter of 100 neurons and an alpha of 0.001 yielded the best mean F1-score of 0.559 as shown in Figure 3. The optimized classifier showed improved classification accuracy of 60.2% on the test set as compared to the first analysis as shown in Figure 4.

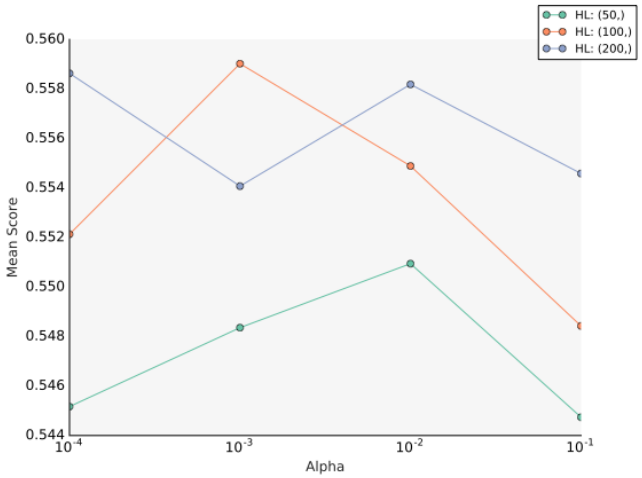

Figure 3. Results of Grid-Search Algorithm for Analysis 2. Mean F1 score vs L2-regularization; Legend: Hidden Layer (HL) Number of Neurons

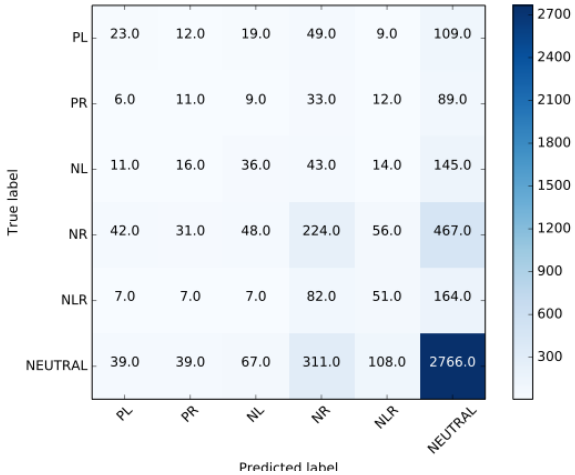

Figure 4.a) Confusion Matrix for Analysis 2

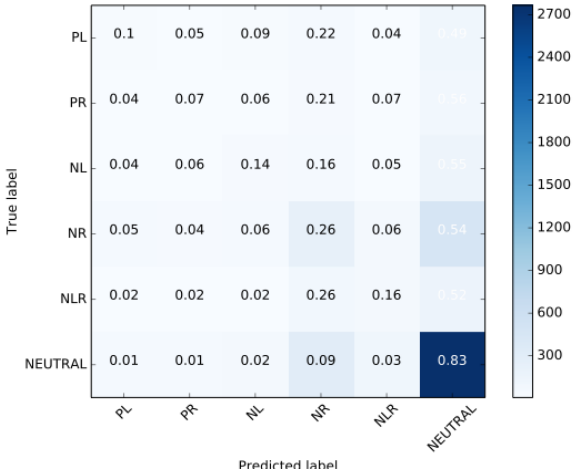

Figure 4.b) Normalized Confusion Matrix for Analysi

Figure 4. Confusion Matrices for Analysis 2

We found that the classifier still struggled with accurately classifying positive sentiments as before although the overall performance of the classifier improved with the second approach. We believe that this is due to the nature of the data itself and that overall positive statements might display a high degree of variability and include nested negative statements as we observed while annotating the data.

6. CONCLUSION

Our study was conducted to assess if fine-grained sentiment analysis could be performed with a dataset extracted from a complex ecosystem like Reddit. We performed sentiment analysis and assessed overall sentiment contained in multi-sentence and multi-paragraph blocks of comments. For this reason, we used new approaches to constructing feature vectors such as paragraph vectors that can be applied to variable-length pieces of text. We attempted to determine if we could accurately classify if commenters were overall positively or negatively viewing a particular event. We then attempted to determine the directionality of the comment's positive or negative sentiment with more fine-grained analysis, for example, if a commenter was positively viewing leaving or remaining in the EU. We found that our model was able to classify overall positive and negative statements but also be able to distinguish the directionality of the

statements. We found that there were manifest differences in our classifier's ability to classify sentiments. Our classifer was in both modes of analysis better at classifying negative and neutral sentiments than positive sentiments and showed a high degree of confusion between positive and negative statements. We believe that this effect might be due to a few issues: the preponderance of negative statements in the overall dataset and the interleaving of negative statements within overall positive statements as observed anecdotally while annotating the dataset. We attempted to offset the first issue by undersampling the majority classes and this showed some improvement in the classification accuracy. To address the second issue, in future research, we will attempt further graded classification, for example, somewhat positive, somewhat negative, very positive, very negative, to assess if this improves our classification accuracy. We have established that satisfactory classification is indeed possible with a complex dataset such as ours and this model can be used to social media to assess sentiments as an addition to polling data.

REFERENCES

[1] L. Lee B. Pang., S. Vaithyanathan. "*Thumbs up? Sentiment classification using machine learning techniques*." ACL-02 conference on Empirical methods in natural language processing, Volume 10, 2002.

[2] Richard H. Byrd, Peihuang Lu, Jorge Nocedal., Ciyou Zhu. "A limited memory algorithm for bound constrained optimization." *SIAM Journal on Scientific Computing*, 16(5):1190-1208, 1995.

[3] G.C. Cawley., N.L.C. Talbot. "On over-fitting in model selection and subsequent selection bias in performance evaluation." *J. Mach. Learn. Res*, 11:2079-2107., 2010.

[4] Rehurek, R., Sojka, P. "*Software Framework for Topic Modelling with Large Corpora*." Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45-50, May 2010.

[5] J. X. Yu G. P. C. Fung., W. Lam. "Stock Prediction: Integrating Text Mining Approach using Realtime News." *Computational Intelligence for Financial Engineering*, Hong Kong, 2003.

[6] M. Gamon, S. Basu, D. Belenko, D. Fisher, M. Hurst., A. C. K¨onig. "*Blews: Using Blogs to Provide Context for News Articles*". 2nd AAAI Conference on Weblogs and Social Media, Seattle, Washington, USA, 2008.

[7] R. Jowell., B. Hedges., P. Lynn., G. Farrant., A. Heath. "Who misled whom? The polls and the voters in the 1992 british election". *St Charles, Illinois*, 1993a.

[8] D. Kingma., J. Ba. "Adam: A method for stochastic optimization". *arXiv*, 1412.6980, 2014.

BIOGRAPHIES OF AUTHORS

Santhi Priya P. She received a B.Tech (Computer Science and Engineering) degree from Jawaharlal Nehru Technological University, Hyderabad (JNTUH), India in 2003 and M.Tech (computer science) degree from JNTUH in the year 2009. She is doing a part-time research in University College of Engineering and Technology, Acharya Nagarjuna University, Guntur, A.P, India. She is working as an Assistant Professor in the Department of CSE in Sri Sairam College of Engineering, Benguluru, India.

Venkateswara Rao T. He received a Bachelor of Engineering degree in Electronics and Computer Engineering and a Masters of Engineering in Computer Science, and a Ph.D in computer science and engineering from Wayne State University, Detroit, USA. He is currently working as Professor in KL University, Vaddeswaram, Guntur Dt, India. He has more than 32 years of experience and has published many papers in national and international conferences. His areas of interest are multicore and parallel programming.