Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 168 (2020) 123–128

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference with Theme:
Leveraging AI and Machine Learning for Societal Challenges, CAS 2019

Prediction of Likes and Retweets Using Text Information Retrieval

Ishita Daga[a]*, Anchal Gupta*[a] , Raj Vardhan[a], Partha Mukherjee[a]

[a]*Pennsylvania State University, Great Valley , 30 E. Swedesford Road, Malvern, PA – 19355, USA*

Abstract

Twitter is one of the major social media platforms today to study human behaviours by analysing their interactions. To ensure popularity of the tweet, the focus should be on the content of the tweet that results in numerous followings of that message with sufficient number of likes and retweets. The high quality of tweets, increases the online reputation of the users who post it. If a user can get the prediction of likes and retweets on his text before posting it on the internet, it would improve the popularity of the tweet from information sharing perspective. In this paper we employed different machine learning classifiers like SVM, Naïve Bayes, Logistic Regression, Random Forest, and Neural Network, on top of two different text processing approaches used in NLP (natural language processing), namely bag-of-words (TFIDF) and word embeddings (Doc2Vec), to check how many likes and retweets can a tweet generate. The results obtained indicate that all the models performed 10-15% better with the bag-of-word technique.

Keywords: TF-IDF; Doc2Vec; Text Mining; Twitter; Predictive Modeling of Words

Introduction

The connected society we live in today has allowed online users to willingly share opinions on an unprecedented scale. Motivated by the advent of mass opinion sharing, it is then crucial to devise algorithms that efficiently identify the emotions expressed within the opinionated content.

Twitter has received a lot of interest and attention from internet users across the globe to share their thoughts and views on multiple phenomenon. Such ease of use, coupled with the widespread use of connected portable devices, has made Twitter the primary channel for users to voluntarily share opinions, news, activities, interests, and other types of event-related information happening around them.

Corresponding author. Tel.: +1-484-320-0735
E-mail address: ixd84@psu.edu

10.1016/j.procs.2020.02.273

When a person writes a Tweet, it is expected that it's content will affect user and bring value to them. On Twitter, articles are posted including hashtags, URL's, titles etc. with a limit of 280 characters. Thus, to create good impression and enhance interest of a reader it is very important to focus on the content, adhering to the character limit. Users can access and express satisfaction in the form of 'likes' and 'retweets' of the original post. Providing a prediction of number of 'likes' and 'retweets' on a post can help users to improvise their subject matter and engage more users. We believe that the tweet gains popularity if it gets more retweets and likes within the user community related to the information the tweet has conveyed. In this research we use a family of classifiers to predict the popularity of tweet by considering number of likes and retweets as a popularity metric.

2. Literature Review

Previously, a lot of work has been done on specifically retweets, like analyzing the social behavior and interactions of people, [1] predicting if they will retweet a certain tweet [2], converting the prediction into a binary problem by dividing the tweets as highly tweeted versus rarely tweeted. Recently with the plethora of data, social media platforms like Twitter have become a hub for studying human behavior by tracking the content they post, the tweets they like, the accounts they follow [3]. These behaviors are used to analyze the recent trends in stock market [4] or world economy on global scale [5]. These researches have considered parameters like word count, followers count, mentions, favorites etc., which are mostly related to a user's profile, our analysis is purely based on the tweet itself, thus focusing on the content of the text rather than other attributes. Our work differs from the prior research from the perspective of usage of NLP based information retrieval for retweet and likes prediction.

3. Data

3.1. Data Collection

Two million tweets were collected from Twitter using 'tweepy' library in Python, for the Year 2015 to 2018 in the field Data Science. Since a tweet can be on any of the multifarious topics, it was necessary to choose a particular topic to perform our analysis. "Data science" being the most trending domain both in industry and academia that gained tremendous momentum in recent years, was chosen for our study, and tweets with tags such as #datascience, #machinelearning were chosen to extract data. We removed unwanted columns like username, geography, mentions, follower count, id, and permalink and kept the following attributes (see Table 1) in our study.

Table 1. Data Description

Feature	Description	Scale
Created	When the Tweet was posted	Timestamp
Like	How many times that Twe was liked	Nominal
Retweet	How many times that Twe was shared	Nominal
Text	The content of the Tweet	String

3.2. Data Pre-processing

For the data pre-processing, we followed the steps listed below:

- Tweets data had many unwanted characters like @RT, www, emoticons encoded in utf-8 and links which had to be removed. Regular expressions in R were used to clean the column.
- Removed stop words (most frequent words with least significance for text procession like a, the, are etc.) using NLTK library.
- Removed Duplicate Tweets after cleaning.
- Removed all the rows which had counts of likes and retweets as 0.

4. Exploratory Data Analysis

The data was explored to analyse the possible metrics that can be used to understand the solution like identifying the relationship between the features with overall performance of the tweet analysing the high-level statistics of the tweets and understanding how many times on an average the tweets were retweeted or liked and the average length of a tweet.

The distribution of various features like, the average length of a Tweet is around 90 characters, were also studied. Likes and retweets are positive-skewed, i.e. they are concentrated on the left part of the graph. To avoid being biased by outliers, data points that don't fit Eq. (1) are removed.

$$Outlier < Q1 - 3 * IQR \quad (1)$$
$$Outlier > Q3 + 3 * IQR$$

Where Q1 and Q3 are the first and third quartile and IQR is the Interquartile Range (IQR = Q3 - Q1). While analysing the relationship between Likes, Retweets and Character length of tweets, it could be observed that there is a relation between Likes and Retweets as shown in Fig.1. . Fig. 2. Shows the most frequently occurring words, for most liked and retweeted tweets, in Data Science field.

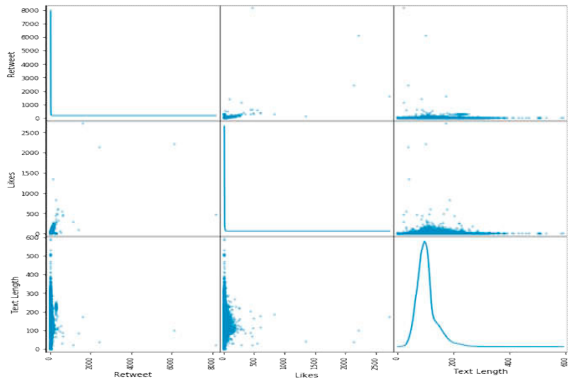

Fig. 1. Scatter Matrix and Correlations between Text Length, Likes and Retweets

Once the data was cleaned the target values, namely Likes and Retweets were binned into four classes: {bin1: [0-1], bin2: [1-2], bin3: [2-5], bin4: [5+]} where bin1 had likes/retweets in the range of 0-1, bin2 had in 1-2 range and so on. By binning the data, a regression problem was converted into a classification problem which is easier to deal with when working with predictions of like and retweets. This binning led to a highly imbalanced data in class 4. To adjust that we had combined the results from bin3 and bin4 but that 1) gave a highly imbalanced class again as the number of data points in bin3+bin4 were higher than the other two bins, 2) was generalizing our problem by keeping all the tweets with more than 2 likes/retweets in the same bin. To overcome this problem under sampling was implemented on the entire data without combining bin3 and bin4.

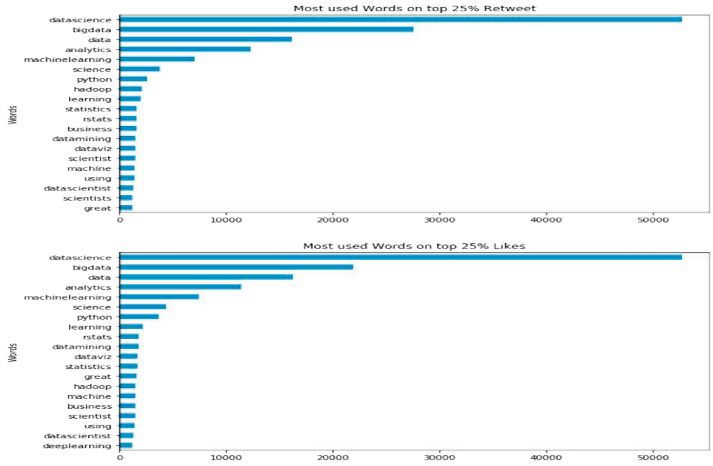

Fig. 2. Top words for top 25% performers in Likes and Retweets features.

5. Methodology

5.1. Text Embedding

When dealing with text data, it is imperative to convert the text data to some numeric matrix or vector. This is necessary because machine learning models normally don't process raw text, but numerical values. For our study, two different NLP methods were approached, one was Term Frequency and Inverse Term Frequency (TF-IDF) [6] and other was Doc2Vec [7].

- The TF takes the term frequency, i.e. the number of times a word (term) has occurred in that particular sentence (tweet) and IDF counts the number of times the word (term) has occurred in the entire document (corpus of tweets). The IDF helps in eliminating the most frequently occurring words, which do not have much contribution while analysing the data.
- Doc2Vec is a method in which the entire document (each tweet) is converted into a vector of a specified size (usually in the power of 2, like 64, 128). The underline process implements a neural network model, which takes the entire document (each tweet) as input and outputs a weight matrix of the specified size.

5.2. Models

Classification is a common task of machine learning which involves predicting a target variable taking into consideration the previous data. Predicting the number of retweets and likes of an article can be treated as a classification problem, because the output will be discrete values (range of numbers). The following classification models were used in this case study:

- Logistic Regression: The Logistic regression [8] model was chosen, as it provides probabilities for outcomes and a convenient probability scores for observations.
- SVM with linear kernel: SVM model [9] was chosen, because it works well with linear and non-linear datasets. Due to the fact we have more samples than number of features, SVM can generate a good prediction.
- Random Forest: Random forest [10] is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.
- Neural Networks: We use neural networks [11] as it works really well with huge amount of data, and also text data.
- Multinomial Naïve Bayes: The multinomial Naive Bayes classifier [12] is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as TD-IDF may also work.

6. Results

For each model tested, the accuracy with the default parameters (e.g., the default parameters for an SVM model are: C=1.0, kernel='rbf', degree=3, gamma='auto_deprecated') was calculated, and then the parameters were changed and tested again to improve the accuracy. To test the combination of the new parameters and fine tune the model, grid search (GridSearchCV) method was used. The grid search method takes different values for each parameter and chooses the combination of parameters which gives

the best accuracy. In our case, 2-3 values were passed for each parameter with a 5-fold cross validation to estimate the model's accuracy. Table 2, and Table 3 shows the result of accuracy achieved for Retweets and Likes with the TF-IDF and Doc2Vec input features. (Note:- we are using train accuracies instead of validation accuracies for all cross-validation results.

Table 2. Accuracy for Retweets in (%)

Model Name	TF-IDF		Doc2Vec	
	Train	Test	Train	Test
Logistic Regression	70.1	47.6	37.03	35.68
Support Vector Machine	62.9	46.9	32.2	35.4
Random Forest	49.8	41.6	80.02	62.67
Neural Network	39.10	44.6	54.16	44.3
Multinomial Naïve Bayes	75.9	51.4	34.02	35.68

Table 3. Accuracy for Likes in (%)

Model Name	TF-IDF		Doc2Vec	
	Train	Test	Train	Test
Logistic Regression	76.9	50.7	36.20	35.06
Support Vector Machine	77.5	51.1	35.9	34.8
Random Forest	52.3	40.8	90.28	60.08
Neural Network	42.77	49.18	47.35	40.65
Multinomial Naïve Bayes	77.7	50.1	33.38	35.08

7. Discussion of Results

It is observed that though TF-IDF gives a better accuracy in all the test cases, there is a lot of overfitting, mostly in less complex models like Logistic regression and Naïve Bayes. Even after implementing 'l2' regularization, we were not able to improve the accuracy much. This can be because, TF-IDF being a bag-of-word vectorizer, is not able to take the word order in account, and hence is not able to generalize on unseen data. On the other hand, Doc2Vec comparatively has a low-test accuracy for all the models, but does not overfit (except in Random Forest). The overall low accuracy of all our models can be because the Doc2Vec technique is built for large texts i.e. 50+ words. A better approach would be to use a short-text embedding technique [13,14] which does not lose the sequential information and is also capable of capturing the context/topic in short text data such as Tweets.

Since simple Machine Learning models were not able to give us a very good accuracy, we experimented with Random Forest model (ensemble learning) as well as a fully connected feed forward neural network (Deep Learning model) to see if the accuracy can be increased. Even though Deep Learning is a highly sort after approach when dealing with unstructured data (such as textual data) or when trying to increase the accuracy, we can say that it is not always the best. Something peculiar in the case of our Neural Net model on TF-IDF is that the accuracy of test data is greater than train data. This can be because we were training both Doc2Vec and TF-IDF features on same Neural Net model, so even though it works well for Doc2Vec, the model might be regularizing a lot on TF-IDF features, which gives us lower accuracy on Train. In our experiments Random Forest on top of TD-IDF gives the best accuracy, and least overfitting and can be considered the best model, compared to the rest.

8. Conclusion

From the results it is seen that even though Doc2Vec would seem to be a more powerful technique when dealing with text data, (as it takes the entire context in consideration for document embedding, while TF-IDF being a bag-of-word approach, loses the context), in our case TF-IDF performed better. This can be because Doc2Vec needs thousands of words in each document to gather context but in our dataset the average word count was 10-12 per document(tweet), which makes it difficult to infer the context. It can be inferred from our findings that when dealing with documents with less word count, like Twitter, it's better to fall back on traditional bag of words approach like TF-IDF or experiment with short-text embeddings.

References

[1] Lee, Kyumin, et al. (2015) "Who will retweet this? detecting strangers from twitter to retweet information." 6(3): p. 31.
[2] Nesi, Paolo, et al. (2018) "Assessing the reTweet proneness of tweets: predictive models for retweeting." Multimedia Tools and Applications. 77(20): p. 26371-26396.
[3] Tsugawa, Sho, and Kito, Kosuke (2017) "Retweets as a predictor of relationships among users on social media." Plos one. 12(1): p. e0170279.
[4] Zhang, Xue, Fuehres, Hauke, and Gloor, Peter A (2011) "Predicting stock market indicators through twitter "I hope it is not as bad as I fear"." Procedia-Social and Behavioral Sciences. 26: p. 55-62.
[5] Dodds, Peter Sheridan, et al. (2011) "Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter." Plos one. 6(12): p. e26752.
[6] White, Howard D (2018) "Bag of works retrieval: TF* IDF weighting of works co-cited with a seed." International Journal on Digital Libraries. 19(2-3): p. 139-149.
[7] Mijangos, Víctor, Sierra, Gerardo, and Montes, Azucena (2017) "Sentence level matrix representation for document spectral clustering." Pattern Recognition Letters. 85: p. 29-34.
[8] Peng, Chao-Ying Joanne, Lee, Kuk Lida, and Ingersoll, Gary M (2002) "An introduction to logistic regression analysis and reporting." The journal of educational research. 96(1): p. 3-14.
[9] Cortes, Corinna, and Vapnik, Vladimir (1995) "Support-vector networks." Machine learning. 20(3): p. 273-297.
[10] Liaw, Andy, and Wiener, Matthew (2002) "Classification and regression by randomForest." R news. 2(3): p. 18-22.
[11] Haykin, Simon S, et al., Neural networks and learning machines. Vol. 3. 2009: Prentice Hall, USA.
[12] Lewis, David D (1998) "Naive (Bayes) at forty: The independence assumption in information retrieval." in European conference on machine learning. Chemnitz, Germany: Springer.
[13] Huang, Heyan, et al. (2017) "Leveraging Conceptualization for Short-Text Embedding." lEEE Transactions on Knowledge Data Engineering, 30(7): p. 1282-1295.
[14] De Boom, Cedric, et al. (2016) "Representation learning for very short texts using weighted word embedding aggregation." Pattern Recognition Letters, 80: p. 150-156.