Who Tweeted That?

Saransh Srivastava, Hardik Sawhney and Nandini Sengupta

The University of Melbourne

1 Introduction

The number of twitter users has shown a huge increase in recent years and is expected to reach more than 500M monthly active users in the year of 2019 [1]. In this project, we are identifying the authors from a twitter dataset consisting of nearly 3M tweets. Many previous studies in author attribution have involved only few hundred authors [1], however, our project classifies tweets of nearly 10K authors. The twitter data is a unique combination of words, numbers and special characters including hyperlinks. Additionally, tweets may have special tokens representing, if it is a retweet and/or if any other user is mentioned in it. From the literature, we have seen the general features that are used for discrimination among texts are *style-based features* such as, style markers (SM) [1,2] and *frequency-based features* such as, bag of words (BoW) [2] and term frequency-inverse document frequency (tf-idf) [3]. Where SMs are indications of the writing style of a twitter user, BoW and tf-idf reveal the tendency of word use among users. These features were used to train classifiers, such as k-nearest neighbour (k-NN) [4, 5], regularized least-squre classifier (RLSC) [4], support vector machines (SVM) [2,5], neural network (DNN) [6] etc. In this work, we have used both SM and choice of words to make our feature space. We train a simple statistical k-NN learner and a deep neural network for class predictions.

In section 2, we note down about the methodology including preprocessing, extracted features and classification model selection. Next, in section 4, we discuss about the results. In section 5, we conclude with some future directions.

2 Data Analysis & Feature Engineering

2.1 Data Imbalance

During investigation of the data, we realised that there are authors who have just tweeted once and then there are authors who have more than 250 tweets. On an average there are nearly 30 tweets per user. Figure 1 shows the histogram of the number of tweets with respect to number of twitter user. 14.7% of the users have either authored more than 100 or less than 16 tweets. We removed these tweets before feature creation.

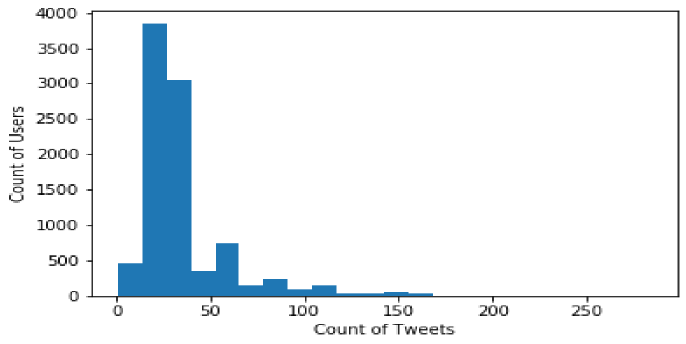

Figure 1: Histogram representation of data distribution.

2.2 Preprocessing & Feature Extraction

We started by removing all tokens marked as ‘RT’, ‘@handle’, and links. We also cleaned the data of any numericals.

Figure 2 represents the feature extraction technique used for classification. We computed SM features per tweet, i.e., (i) number of links, (ii) number of hashtags, (iii) word count, (iv) character count, (v) number of capital letters, (vi) special character count and (vii) if it is a retweet. We observed large variations in word count, character count and special character count values and thus normalized the parameters using min-max scaling.

[1] `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/`

To obtain more meaningful feature projection of the texts, we tokenised each tweet, removed stop words and then stemmed them. We did not lemmatize in order to preserve the original form of the words. This process helped us in reducing the vocabulary size while creating BoW and tf-idf vectors. In order to remove sparsity in the feature space, we applied singular value decomposition (SVD). SVD reduces the correlation among feature vectors, which helped us in classification. It reduced the dimensions of BoW and tf-idf vector from nearly 1M to 1000. We empirically, tried 100 to 2000 reduced vector space and realised 1000 gave both good accuracy and required less computational resource. We thus, made 2007 feature vector size including seven SMs, 1000 BoW representation and 1000 tf-idf representation.

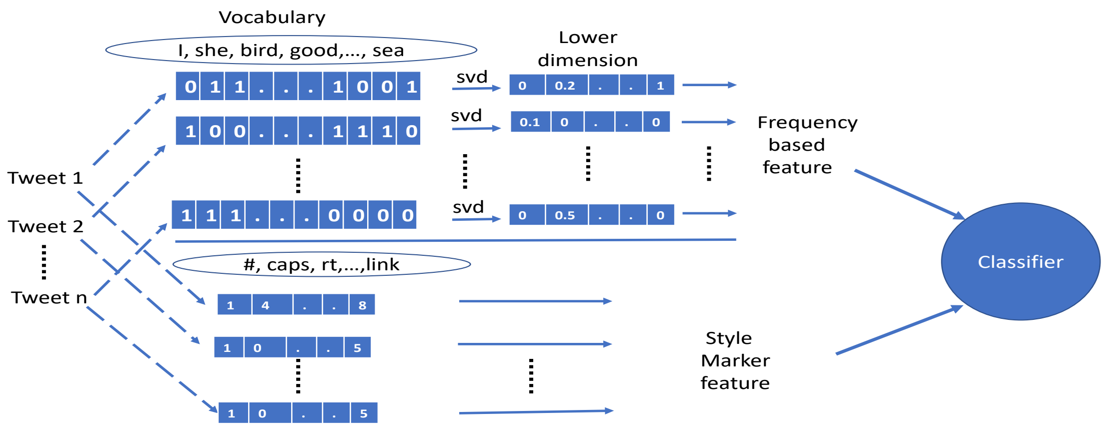

Figure 2: Block diagram of features utilized for classification.

3 Classification Model Selection

We developed a baseline model to compare our other solution with it. We looked for suitable learners used in previous literatures and decided to run them for our problem.

To begin with, we ran a simple k-NN model on our features, which had already been used for text classification [4]. k-NN works by calculating the minimum distance from the current input to the training samples to determine the k nearest neighbours. The majority class of these k- neighbours is taken as the final prediction. We ran the algorithm several times with different values of k and recorded the accuracy for each. We realised that the poor performance of k-NN is because the dataset was huge [7]. k-NN fails to perform well with large datasets as it's computation depends primarily on distance between classes. As the volume of data increases, the distance becomes less representative.

SVM, random forest and other linear classifiers are less efficient when handling huge multiclass dataset [8, 9]. We wanted to train our features on a model which can handle large datasets with high dimensions and high classifications. Deep neural networks have performed well in past [8] on such problem sets. Our classification model was trained using 80% of the labelled data and 20% was set aside for validation. Input layer had 2007 nodes corresponding to feature vector dimension and output layer had 8286 nodes representing the number of classes. We experimented with our NN by both deepening and widening it, and realized that 4 layers with 30-100 varying neurons width performed optimal. To reduce independent learning among nodes, we employed dropout at the visible layer. Rectified linear unit (ReLu) was used for the hidden layers [10], and softmax function was chosen for classification layer.

4 Results & Discussion

The k-NN classifier using only SM features gave us poor results as shown in fig. 3. We augmented the feature set with more information and trained DNN using all three types of features. Because the author ids were not sequential, we encoded them and then passed it to our learner. After varying number of epochs and batch size, the model performed best when batch size was high and epochs were less than 100. Additionally, a high epoch value for low batch size was over fitting the model. This was due to the fact that with small batches after a certain number of epochs, the classifier started learning noise of the training data. A large batch size ensures variability in author ids ensuring generalization during each epoch. The corresponding values are in Table 1. Overall, the prediction accuracy of our DNN was 9.763% on the unlabelled dataset on the Kaggle platform.

Table 1: Results using DNN classifier changing epochs and batch size.

Layers	Batch Size	Epoch	Training Loss	Training Acc	Validation Loss	Validation Acc
4	20	200	4.319	0.320	8.015	0.074
4	1000	200	7.410	0.095	7.190	0.090
4	20	50	6.156	0.091	7.918	0.076
4	1000	50	9.152	0.073	9.520	0.064

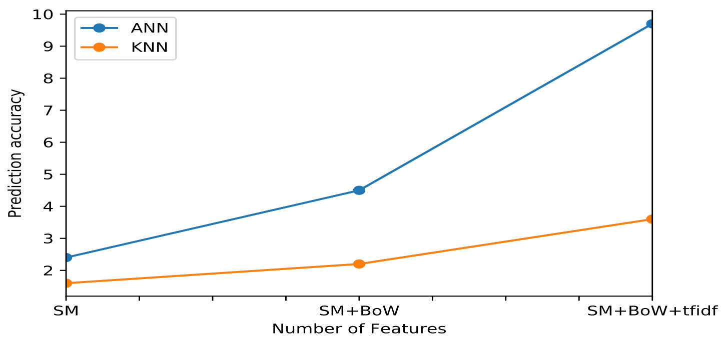

Figure 3: Accuracy comparison of KNN and DNN using three features separately.

We found from our results that the accuracies in both the cases i.e., with a balanced dataset and with unbalanced dataset were comparable. This might be because the minority classes in training set have high tweet occurrence in unlabelled data or the balancing criterion of our model needs improvement.

5 Conclusion with Future Direction

With our present implementation, we were able to predict a subset of authors accurately. We realised that in order to further improve performance, we will need to augment our feature vector space with sentence semantics such as POS and N-gram. In future, we will also like to use ensemble bagging and boosting techniques, by training weak learners like decision tree or random forest on subsets of authors and train a strong learner like an ANN or SVM on the predictions of those outcomes.

References

[1] İ. Mayda and M. Yeşiltepe, "N-gram based approach to recognize the twitter accounts of turkish daily newspapers," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, 2017, pp. 1–5.

[2] R. M. Green and J. W. Sheppard, "Comparing frequency-and style-based features for twitter author identification," in *The Twenty-Sixth International FLAIRS Conference*, 2013.

[3] X. Yang, C. Macdonald, and I. Ounis, "Using word embeddings in twitter election classification," *Information Retrieval Journal*, vol. 21, no. 2-3, pp. 183–207, 2018.

[4] A. Castro and B. Lindauer, "Author identification on twitter," 2012.

[5] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 300–314.

[6] R. Moraes, J. F. Valiati, and W. P. G. Neto, "Document-level sentiment classification: An empirical comparison between svm and ann," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621–633, 2013.

[7] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?" in *European conference on computer vision*. Springer, 2010, pp. 71–84.

[8] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 364–371.

[9] X. Solé, A. Ramisa, and C. Torras, "Evaluation of random forests on large-scale classification problems using a bag-of-visual-words representation." in *CCIA*, 2014, pp. 273–276.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.