Short Text Location Prediction - COMP30027 Report

Anonymous

1. Introduction

The aim of the report is to determine the most appropriate machine learning classifier that produces a high precision and low error rate when predicting the geolocation of short text posts from Twitter. Constructing and implementing a probabilistic learner and classifier using unstructured text demands a greater attention to filtering anomalous data – such as location neutral tweets – determining the best features in the labelled training dataset and lastly verifying of the accuracy of the classifier (Lee et al., 2019).

In order to achieve a classifier that has a low error rate and fine-grain location predictions, this task has been substantially simplified. We have reduced the geolocation spread to Australia, in the following cities exclusively: Brisbane, Perth, Melbourne and Sydney. By reducing the dataset, we are able to focus on building a classification-based algorithm that can accurately predict geolocations based on frequencies of keywords within a small region of the world. Although this approach suggests that the classifier may be limited in its applicability on a large scale – having a high bias, we will be able to determine key parameters and draw conclusions on how one can accurately predict the geolocability of text posts on a variety of social media platforms. Moreover, by studying the performance of a variety of machine learning algorithms, we will able to investigate the impact of textual features such as location indicative words (Hennig and Thomas, 2018).

To tackle location sparseness, we must not only apply external knowledge but also explore effective classifier combinations that consider the randomness in the tweet content.

2. Related Work

Extensive work on geolocability in unstructured text posts has reduced the scope of this issue to infer tweet text post locations through 1) tweet content and context and 2) location-centric or network-based methods (Zheng, Han and Sun, 2018). Previous research efforts have suggested that a reasonable classifier accuracy can only be achieved through classifier combinations and applying external knowledge to initialize classification rules *(Hennig and Thomas, 2018)* for the tweet content. Furthermore, by creating a probabilistic model that is built off of the training data alongside external knowledge such as slang found in the tweet, we will be able to generate a granular probabilistic model on user behavior to predict their tweet location. Due to the limitation of this project, we are unable to explore the realm location-centric methods such as stay point detection and GPS-generated spatial-temporal data to perform contextual location prediction (Xia, Huang and Wu, 2018).

3. Short Text Location Prediction

To address the key issue of geolocability of Twitter posts, we were limited to raw training, development and test datasets that were preprocessed into the top 10, 50 and 100 token word frequencies with binary features where 1 signifies the existence of the token word in the tweet and 0 signifies the inexistence of the token word.

3.1 Preprocessing

Prior to exploring supervised learning machines, pre-processing such as missing data handling, categorical data encoding and analysis of the data was performed in order to better understand the large amount of training and development instances in the data. Additionally, further preprocessing on the token words (features) such as stemming and lemmatization were also performed to remove sparse terms, stop words – which provide no value to the probabilistic model – and lastly inflectional forms to a common base word (Davydova, 2018).

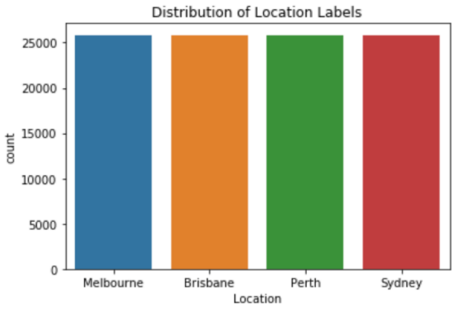

Figure 1 Distribution of location labels for training data

Figure 2 Distribution of location labels for development data

3.2 Implementation

In this section, we explored several classifier behaviors to determine the most appropriate and best fitting algorithm for the dataset in terms of tweet context and content. Due to the nature of the training data, the most logical approach to building the optimal learner is to work with simple classification learners as there is not enough data to develop more complex models such as neural networks.

A Multinomial Naïve Bayes (NB) learner was first tested – as it is recognized as the best learner for categorical text analysis. By using the token frequencies, the learner built a probabilistic model of discrete word counts. Although an appropriate learner for the task, the Gaussian NB was preferred over multinomial and Bernoulli methods as it outputted a greater accuracy (30.1%). Additionally, Gaussian NB assumes a normal distribution of token frequencies which are inherently random and cannot be accurately measured by frequency counts of each token word or binary features.

The support vector machine (SVM) learner was also tested as it considers the interactions between features – token frequencies. SVM learning adopts a spatial approach to classification whereby instances are mapped categorically, and class labels are divided by a clear line. The accuracy score for the SVM learner is not an ideal accuracy score (29.7%), however, is expected due to the randomness of the tweet content and locations. Ostensibly, SVM learner has a lower accuracy nevertheless, it predicts locations assuming a certain degree of interdependence between features.

Lastly, classifier combinations were also implemented such as random forest classification (RFC), bagging and stacking

4. Evaluation

We realized that the models are being "forced to predict geolocability on data that contains instances which may not even insinuate a location in its semantics" (Balwdin, Bo and Cook, n.d.). Henceforth, the average accuracies conjured from the exploration of classifier behaviors are not dismal as the models are essentially predicting geolocability on data that has low prediction capability.

We used the One-Rule classifier as a baseline to compare the performance of the other classifiers. One-Rule minimizes the error rate over the training data between classification rules by creating decision stumps for each attribute. We found that the best features (or token words) were location indicative words such as the names of the cities, thus the classification rule was based on this discovery (Figure 3). This discovery begs to question whether we should build probabilistic models including these location indicative words as they manipulate and essentially overfit our data to the model. For example, some tweets from one city can be referring to another depending on the context of the tweet thus creating high bias to the test dataset due to poor convergence criterion.

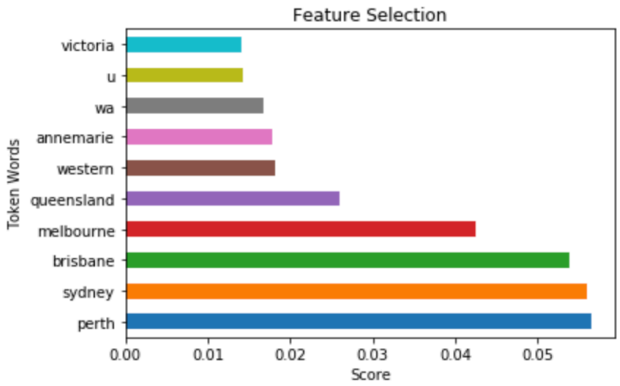

Figure 3 Best feature selection

After further analysis, we found that omitting these location indicative words when fitting models outputted a lower accuracy, thus were retained.

To compare the classifier performances, we evaluated on accuracy, f1-score and error rate.

$$Accuracy = \frac{Number\ of\ correctly\ labelled\ test\ instances}{Total\ number\ of\ test\ instances}$$

$$Precision_M = \frac{\sum_{i=1}^{c} Precision(i)}{c}$$

$$Error\ rate = 1 - Accuracy$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

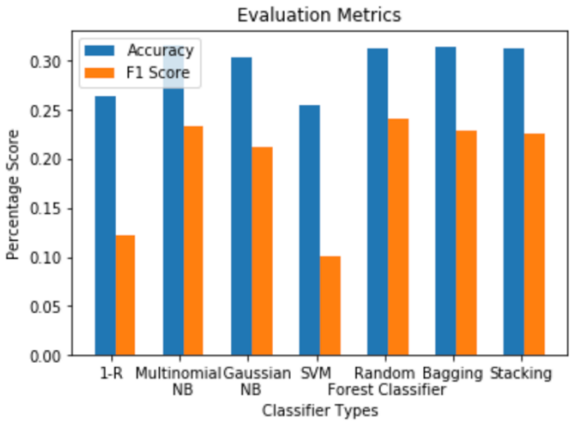

Figure 4 Classifier performance

As it appears, bagging is proven to be the most appropriate and best performing classifier out of the top 7 best supervised classification learning methods. The difference between RFC and the other learners is that bagging adopts a classifier combination method that tends to overfit and the learner is fitted using several random subset combinations of the training data. Additionally, bagging is highly effective over noisy data and minimizes model variance through its sampling. RFC has significant improvements when hyperparameters were optimally tuned for the number of trees and size of random subset at each node are adjusted. Moreover, we were able to reduce variance through hyperparameter tuning to minimize overfitting. Overall, we favour RFC as bagging is much more computationally expensive for what is already a small dataset.

In juxtaposition to RFC, multinomial NB outputs a similar accuracy however a lower F1 score thus lesser ability to balance precision and recall. For comparison, a good F1 score is closer to 1 whereby the model outputs low false positives and false negatives. A limitation to our error analysis is the ability to minimize one score over another. Consequently, we used the stacking method of the best classifiers, applied feature engineering and found a good accuracy that did not overfit the training data.

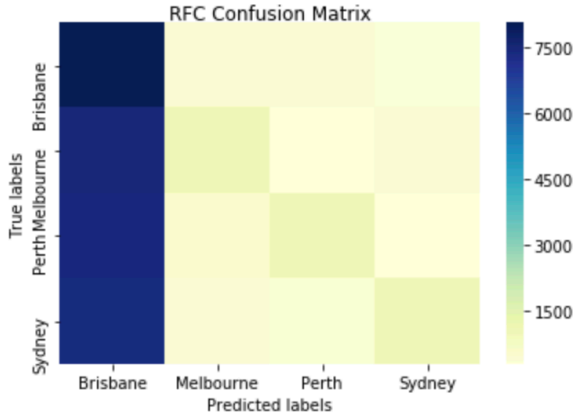

Figure 5.1 Confusion Matrix for Random Forest Classifier

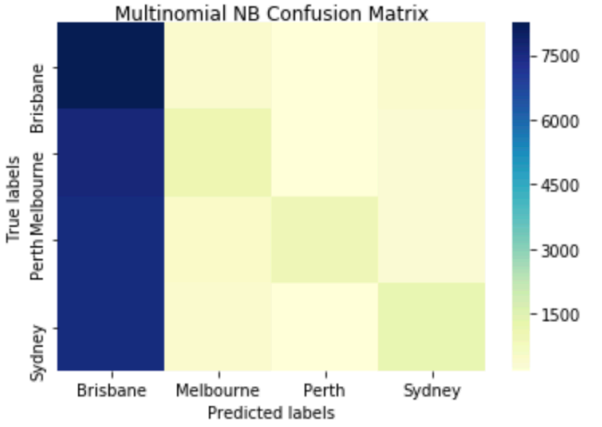

Figure 5.2 Confusion matrix for Multinomial Naive Bayes

A surprising discovery was the low accuracy and precision scores for SVM which was expected to be one of the better performing classifiers as it "acknowledge[s] the particular properties of text: (a) high dimensional feature spaces, (b) few irrelevant features (dense concept vector), and (c) sparse instance vectors" (Joachims, 2019) (Figure 6). However, the accuracy for SVM may have been lower than the baseline as the dataset is a complex multiclass problem with a large number of features. Furthermore, SVM is developed for binary classification problems thus a "1-against-all strategy or hyperparameter tuning" prior to implementing the classifier could be further explored (A. Lawala and A. Abdulkarim, 2017).

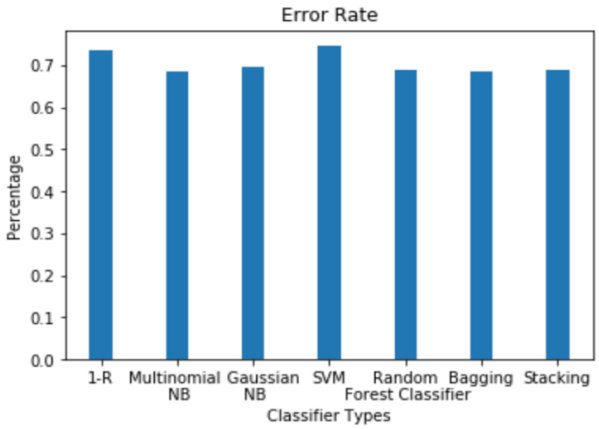

Figure 6 Error rate of classifiers

5. Conclusions

Tweet geolocability has been a trending problem on Twitter due to the incompleteness and inaccuracy of pre-existing data. In this exploration, it can be deduced that the performance of classifiers is greatly affected by the initial nature of the training data provided. Fortunately, the preprocessed training, development and test datasets provide a good starting point to understand the best performing classification learners for this task. We learnt that using contextual features such as location words enhances prediction accuracy significantly but creates high bias in the model. Moreover, RFC appears to be promising in the sense that with minimal data manipulation, its underlying assumptions assists the learner to predict locations naturally using hard voting – a behavior that is reflective of users and tweets alike. However, we can improve on this learner by utilizing nested ensemble learners or stacking, where "class-probabilities of the first-level classifiers can be used to train the meta-classifier" for better accuracy.

6. References

- Lawala, I. and A. Abdulkarim, S. (2017). Adaptive SVM for Data Stream Classification. [online] Scielo. Available at: http://www.scielo.org.za/pdf/sacj/v29n1/04.pdf [Accessed 20 May 2019].
- Arun, V. (2018). Predict Product Success using NLP models. [online] Towards Data Science. Available at: https://towardsdatascience.com/predict-product-success-using-nlp-models-b3e87295d97 [Accessed 20 May 2019].
- Balwdin, T., Bo, H. and Cook, P. (n.d.). Geolocation Prediction in Social Media Data by Finding Location Indicative Words. [online] Melbourne: University of Melbourne. Available at: https://www.aclweb.org/anthology/C12-1064 [Accessed 19 May 2019].
- Davydova, O. (2018). Text Preprocessing in Python: Steps, Tools, and Examples. [online] Medium. Available at: https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908 [Accessed 27 May 2019].
- Hennig, L. and Thomas, P. (2018). Language Technologies for the Challenges of the Digital Age. Twitter Geolocation Prediction Using Neural Networks. [online] Springer, Cham, pp.248-255. Available at: https://link.springer.com/chapter/10.1007/978-3-319-73706-5_21#aboutcontent [Accessed 13 May 2019].
- Joachims, T. (2019). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. [online] Dortmund, Germany: Universitat Dortmund. Available at: https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf [Accessed 19 May 2019].
- Lee, K., K. Ganti, R., Srivatsa, M. and Liu, L. (2019). When Twitter meets Foursquare: Tweet Location Prediction using Foursquare. [online] Atlanta, GA, USA: College of Computing, Georgia Institute of Technology. Available at: http://www.istc-cc.cmu.edu/publications/papers/2014/twitter_www2014.pdf [Accessed 14 May 2019].
- Raschka, S. (2019). StackingClassifier - mlxtend. [online] Rasbt.github.io. Available at: http://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/#example-2-using-probabilities-as- meta-features [Accessed 21 May 2019].
- Xia, L., Huang, Q. and Wu, D. (2018). Decision Tree-Based Contextual Location Prediction from Mobile Device Logs. Mobile Information Systems, [online] 2018, pp.1-11. Available at: https://www.hindawi.com/journals/misy/2018/1852861/ [Accessed 18 May 2019]
- Zheng, X., Han, J. and Sun, A. (2018). Correction to "A Survey of Location Prediction on Twitter." IEEE Transactions on

Knowledge and Data Engineering, [online] 30(11), pp.2227-2227. Available at: https://www.researchgate.net/publication/316820996_A_Survey_of_Location_Prediction_on_Twitter[Accessed 14 May 2019].