PREDICTION SYSTEM FOR CLASSIFYING GEOLOCATION OF TWEETS

Anonymous

1 Introduction

With large amount of data being generated everyday, it hasn't been very long since Twitter became a prominent platform for academic research on a wide variety of topics. This paper attempts to build a system, which aims to predict the location of a twitter user, based on the content of what the user has tweeted. The paper talks about various machine learning classification algorithms that will be inherited to solve the respective task, and will eventually assess the effectiveness of every method by evaluating the results using appropriate metrics.

2 Dataset

The dataset provided has been divided into two sections:

(i) the RAW tweets dataset, which contains the unprocessed tweets, along with the `tweetid` and `userid`, &

(ii) the ARFF format(compatible with WEKA) files, which contain pre-processed instances(tweets) of our data, along with some attributes that have been selected using appropriate methods. This has been discussed in detail as we move further in the report.

In both sets of files, the total tweets have been partitioned into a labelled training set, where the classifier- models get trained, an evaluation set, upon which the models get evaluated on their performance, and lastly a test set, which contains the unseen data, on which the preferred model will produce the required predictions. Of the total 1,63,590 tweets, the division has been made as follows:

Set	Number of tweets	Percentage (approx.)
training	96585	60%
evaluation	34028	20%
test	32977	20%

The dataset for this particular task has been restricted to three target classes only: NewYork, California & Georgia.

3 Relevant Literature

Research on this topic has attracted many data experts in the last decade, who have consequentially succeeded in giving an apt direction towards the given problem. Cheng et al. [1] discusses a pure content-based approach on geo-locating the location of a tweeter, based on a probabilistic framework. It provides a solid base by identifying local[1] words, selected using different attribute selection algorithms. In a nutshell, the classifier used "estimates k cities for each user with a descending order of probability".

Miyazaki et al. [2] uses a knowledge-base[2] for every selected feature, and accordingly assigns a class to the unseen data. This paper takes inspiration from the respective studies and constitutes a supervised learning approach in building a multi-class classifier for locating the tweeter.

4 Feature-Selection

Going by a survey in Forbes, 80% of any machine learning task is - data preparation. Following the similar lines, the features/attributes from the RAW tweets have already been engineered upon. As a result, ARFF format files have been provided which contain the top 10, 20, 50, 200 frequently occurring terms for each of the three target classes. Also, the best 10, 20, 50 & 200 terms have been determined based on the mutual information[3], for each of the three classes.

For this paper, the *bestXX* files have been used as the suitable attributes for creating the model. The rationale behind this lies in the fact that a mutual information based criteria has been adopted to subset the attributes. Previous studies have shown that feature-engineering based on mutual information has, for most of the time, been a good practice in case of high-dimensional datasets and for scattered relationships between features and target classes. Thus, the models will be built using these best[4] features.

---

[1] local words here means the words that have a concrete geographical scope in context to a particular city.

[2] Knowledge-base is a semantic-relation database that contains every possible information about an attribute(word) relevant to the research.

[3] MI gives a number which tells how good a term is in predicting the respective class.

[4] Based on mutual information: high to low

5 Methods

For the classification problem here, 3 supervised learning approaches were used, along with a baseline method, hypothesizing that the other 3 algorithms will be better than at least the baseline method.

Zero-R, serves as the naïve baseline method, which simply classifies according to the most common class in the training set.

Naïve Bayes(NB) classifying method, because of the probabilistic framework it is based on, is used as the first model. A NB model assumes that each of the features it uses are conditionally independent of one another, given some class.

J48 Decision Tree algorithm is a statistical classifier, which builds decision trees using the training set, based on entropy. Information gain, or entropy, is the crux of the J48 decision tree which decides, at each tree node, which feature fits better in terms of target class prediction.

Lastly, the Multinomial Naïve Bayes(MNB) method shall be implemented, and later in the report, it becomes clear how this method overshadows the other two in giving good results. The MNB method is a specific instance of NB classifier, using multinomial distribution for each of the users. Russel et al. [3] proves that MNB works well with data that can easily turn into counts, such as the word counts here.

6 Evaluation of results

The idea here is to apply each of the above 3 methods, on the 3 sets, viz., training, evaluation and test set, according to best 10, best 20 and best 50 terms. Using WEKA, the train sets for each of the 3 *bestXX* files is given as an input, and a method is applied and then accuracy[5] is recorded. The model is then evaluated using the evaluation set, which gives an overview as to how good the model is. Finally, the unseen test data is supplied to the respective model for getting a label predicted for each instance. The results in the form of accuracy for the best10, best20 and best50 are calculated after running various models. The results for the Best50 terms are given as:

Best50:

Method	Dataset	Accuracy
Naïve Bayes	Train Set	59.13%
	Evaluation Set	60.81%
J48 Decision Tree	Train Set	99.53%
	Evaluation Set	49.48%
Multinomial Naïve Bayes	Train Set	64.77%
	Evaluation Set	65.56%

The rationale behind using accuracy as our evaluation metric is that it works the best when there are equal number of samples belonging to each class.

7 Critical Analysis

What can be inferred from the results is that the decision trees' method gets evicted from the task because it gives out a very high accuracy on the train set, but on the evaluation set, the accuracy drops at a very sharp rate. Thus it tends to overfit the data and doesn't seem a suitable approach.

On the other hand, both NB and MNB methods maintain their consistency in building much better models for our datasets. It becomes rather difficult to choose between the two, but eventually building the models on the best50 and more larger feature sets, we realise that MNB tends to achieve higher accuracy as the features get added.

Therefore, MNB turns out to be an appropriate method for building the classifier when the features are selected using mutual information criterion. Working further down on the model, yet another feature selection method was applied on the best50 file, which consisted of 120 features in total for all 3 classes. On adopting the PCA[6] approach, the features were reduced to 114, and there was a slight increase in the accuracy when the MNB model was run using this feature set. The results below show the slight betterment over the previous models in classifying the unseen data.

Precision	0.644
Recall	0.647
F-measure	0.54

Therefore, adding more features and applying appropriate dimension reduction methods, might provide us with a better classifying model.

[5] accuracy = $\frac{\text{No. of correctly classified instances}}{\text{Total number of instances}}$

[6] Principle Component Analysis is a traditional statistical method for dimensionality reduction of data.

8 Conclusion

Using Multinomial Naïve Bayes method, a 3-class classifying model has successfully been created, but this might not be the most efficient algorithm for predicting the classes. Using the features based on mutual information may not always be the correct approach for feature-engineering. It might be possible that combining the frequently occurring terms(mostXX) along with the bestXX, and applying some filtering method like wrapper subset, give us better results as compared to the above.

In the end, it is pretty evident that it is not(just) the algorithm that makes a classifier perform well, but the set of features that give it the strength to achieve greater accuracy. There's still a lot of research going on in this direction, and probably there would be more effective algorithms in future that would overcome the shortcomings in this procedure.

9 References

[1] Cheng Z., Caverlee J., and Lee K. "*You are where you tweet: A content-based approach to geo- locating twitter users*". In CIKM'10, Toronto, Ontario, Canada, 2010. ACM

[2] Taro M., Afshin R., Trevor C., and Timothy B. "*Twitter Geolocation using knowledge-based methods.*" NHK Science and Technology Research Laboratories, University of Melbourne. (2018)

[3] Stuart J. Russell and Peter Norvig. 2003. Artificial Intelligence: A Modern Approach (2 ed.). Pearson Education. *See p. 499 for reference to "idiot Bayes" as well as the general definition of the Naive Bayes model and its independence assumptions*

[4] Eisenstein, Jacob, et al. A latent variable model for geographic lexical variation. Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2010.

[5] Rahimi, Afshin, Trevor Cohn, and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. arXiv preprint arXiv:1804.08049 (2018).