Detect Geolocation from Twitters

Anonymous

1 Introduction

Can we find out the geolocation of a twitter user based on its contents? If so, how can we extract meaningful information related to location in an efficient way? The aim of this report is to implement a system that predicts users' location. The initial system was implemented based on the the words that have highest mutual information gain and chi square scores on different models. Later an attempt to combine text information based on TF-IDF with other meta features was further made.

2 Dataset

The datasets involves raw tweeter dataset and readily made ARFF format dataset. The twitter data set based on the data se(Eisenstein, Jacob, et al., 2010)[1] and (Rahimi, Afshin, Trevor Cohn, and Timothy Baldwin, 2018)[2]. The ARFF format datasets comprise of 20, 50, 200 best and most words. Best words mean the extracted best words from the twitter set that have the highest mutual information and chi square scores against each class. Most file means the words mostly used in the whole document. In this experiment in this paper, the most file was not used and best file was used for the initial experiment for baseline.

3 Evaluation Metrics

To evaluate the implemented system, the following terms will be used in this paper. Hold-out strategy was used by separating training dataset and test dataset for evaluation.

- Accuracy: Accuracy was calculated by counting number of the correctly returned results and number of data not returned by the system which is different from the label in the datasets.
- Precision: Precision is calculated as the number of correct positive results divided by the number of all positive results returned by the classifier.
- Recall: Recall was calculated as number of correct positive results divided by the number of all relevant samples
- F1 score: F1 score is was calculated as harmonic mean of the precision and recall.

4 Baseline

4.1 Methodology

Here the provided dataset(10, 20, 50, 200 best.txt) was used by using software Weka. An evaluation by using different models are seen in the table. Later, this text information was calculated in the system based on TF-IDF model on different models for better feature modeling.

4.2 Analysis

The result with best200.ARFF can be seen in the table 1. The best file held important information that can be a key to deciding the class. However, it also had some limitations. First, the provided words had limited information. It included stopwords such as his, my, etc. was included in the dataset. And it did not have several features that could be found in the original raw tweets such as user id, tagged users, hashtags, emoticons, etc. Therefore, more elaborate preprocessing was needed to check if these not included tokens may have significant information. Secondly, as seen in the table, it has a problem of imbalance. While it predicts the label for New York pretty well, the accuracy for the other classes were extremely low. Therefore, these are the two keys to tackling the low accuracy, which were considered in the latter system.

5 TF-IDF model

As discussed in the previous section, limited text information and imbalance are two huge

	Precision	Recall	F1 score
California	0.451	0.153	0.229
Georgia	0.511	0.072	0.126
NewYork	0.672	0.952	0.788
Accuracy			0.654
Weighted	0.603	0.654	0.572

Table 1: evaluation metrics

problems. In this section, to tackle the problems, the methodology based on TF-IDF models to obtain information on the texts and methodology in order to handle the imblance are introduced. TF-IDF model, which is term frequency and inverse document frequency model, is often considered to be a good model for text classification. It fits many different kinds of supervised machine learning method such as Naive bayes, Random forest, logistic regression, etc.

5.1 Preprocessing

First, preprocessing of the raw tweets were conducted. Here, TweetTokenizer was used for tokenizing sentences as this tokenizer was better fitted for this tweet text in that it reduces repeated characters to a certain length i.e. haaaaaaaa to haaa and can contain userids, hastags and emoticons that might be excluded by many other tokenizers. Swearwords were extracted from here[3] for the future use. After tokenization, stopwords, special characters, and punctuation were removed and lemmatized word was stored.

5.2 Feature engineering

In the previous section, the provided ARFF format data did not include user id that might have significant information to predict the class. However, user id might be weak in predicting unseen data so instead of user id, tagged user id was considered for feature. If a certain tagged id can be seen across a particular document, it may work as a good factor. Second, in terms of text features, TF-IDF was used to process the text into numerical format to feed the model. It is best to set threshold the maximum feature that can be extracted from TF-IDF. However, as seen in the previous section, the top best features might be concentrated in one class, i.e. NewYork, thus resulting in imbalance in prediction of accuracy. Therefore, feature selection based on chi square score in each class was considered. The steps to get chi square score to get top features for each class are as follows:

- TF-IDF score is calculated for all records.
- Sort TF-IDF score by each class. i.e Select top 20 features from Georgia
- combine the vocabulary that obtained top scores from each class and remove the duplication.
- Feed the combined vocabulary again to the TF-IDF vectorizer

Examples of the words that have the highest chi square scores from each class can be seen in the table below. Note that the word was lemmatized during preprocessing.

California	Georgia	NewYork
mor	famusextape	lml
gw	willies	lmaooo
hella	atlanta	lmaoo
hahaha	thatisall	inhighschool
haha	atl	haha

Table 2: Examples top best words in each class

5.3 Model Selection

Multinomial Naive Bayes and Random forest were considered in this experiment. Due to its imbalanced text classification problem, random forest did not show good performance. Naive bayes model showed better performance with more features[4], but it might have resulted in overfitting in the evaluation of unseen data (test-tweet.txt). In terms of implementation, sklearn's package (RandomForestClassifier and MultinomialNB) was used.

5.4 parameter scaling

In an effort to prevent overfitting, parameter adjustment in RandomForestClassifier was conducted by increasing min-df and decreasing max feature gradually. Furthermore, to tackle imbalance, parameter classweight='balanced' was set. For TF-IDF vectorizer, min-df was set to 10 and max-df was set to 0.5 so that frequently used across all documents are disregarded. sublinear parameter was useful for feature scaling. [4]

5.5 Analysis

The results of applying this methodology can be seen in Table 3 and 4. This methodology showed a better performance compared to baseline. However, the evaluation metrics for minority, which is California and Georgia, was poor.

The imbalance problem still remained. In this section, in an attempt to tackle this issue, sampling method is introduced.

	Precision	Recall	F1 score
California	0.97	0.64	0.77
Georgia	1.00	0.58	0.74
NewYork	0.83	1.00	0.91
Accuracy			0.86
Weighted	0.88	0.86	0.85

Table 3: evaluation metrics tested on Naive Bayes model

	Precision	Recall	F1 score
California	0.97	0.64	0.77
Georgia	0.37	0.90	0.52
NewYork	0.95	0.69	0.80
Accuracy			0.71
Weighted	0.85	0.71	0.75

Table 4: evaluation metrics tested on Random Forest model

6 Sampling

In order to tackle the imbalance, sampling method was considered. There are many sampling methods, but among them, particularly, SMOTE technique was used. SMOTE stands for Synthetic Minority Oversampling Technique.. It generates new samples in by interpoloation but samples used to generate new Synthetic samples differ.[1], For the implementation, SMOTE from imblearn library was used.

6.1 Analysis

With sampling method, it was expected to see greatly improved result. However, as seen in the Table, the accuracy was decreased when applied on Naive Bayes model. There might have been a problem in the method of plugging in these features on models. On the other hand, evaluation metrics for random forest did not change. Possibly, its inherent algorithm along with the parameter(class weight = 'balanced') may be the reason for this. Many other sampling library was used but did not result in a better performance. Therefore, the use of sampling method was not considered in the final evaluation.

	Precision	Recall	F1 score
California	0.41	0.94	0.57
Georgia	0.99	0.58	0.74
NewYork	0.95	0.69	0.80
Accuracy			0.72
Weighted	0.85	0.72	0.74

Table 5: evaluation metrics tested on Naive Bayes model

	Precision	Recall	F1 score
California	0.97	0.64	0.77
Georgia	0.37	0.90	0.52
NewYork	0.95	0.69	0.80
Accuracy			0.71
Weighted	0.85	0.71	0.75

Table 6: evaluation metrics tested on Random Forest model

7 Conclusions

Overall, through careful analysis of the original data along with feature engineering and accordingly adjustment of parameter, acceptable result was achieved. An attempt to tackle the imbalance with sampling technique was made but did not lead to a better performance in accuracy. It turned out that the originally provided ARFF format was a good information. However, through processing the data and feature engineering, more knowledge was gained such as top N features that have high chi square scores in each class. Furthermore, it was observed that slight adjustment of the parameter of the model also leads to a different result. However, meta features other than text features did not show an improvement. There might have been a problem in the method of plugging in these features on models. Last but not least, sentiment analysis of the tweet might be worth trying as the sentiment analysis indirectly indicates the inclination of a user to a certain direction, which may be used in prediction geolocation.

References

[1] 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.

[2] Rahimi, Afshin, Trevor Cohn, and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. arXiv

[1] https://imbalanced-learn.readthedocs.io/index.html

preprint arXiv:1804.08049 (2018).

[3] http://jultika.oulu.fi/files/nbnfioulu-201801111060.pdf

[4] https://datascience.stackexchange.com/questions/12607/tf-idf-vectorizer-doesnt-work-better-than-countvectorizer