COMP30027 Report - Short Text Location Identification

Individual Anonymous

1 Introduction

Nowadays, millions of users share information and communicate through social media, which makes it widely regarded as a resource for data extraction in data science studies. Twitter, for example, is one of the largest information sharing platforms. Various valuable information can be interpreted from the content of tweets.

This report describes the machine learning approaches to classify users geographic location, as one for four cities (Brisbane, Perth, Sydney, Melbourne) based on the content of their tweet. To achieve reasonable high accuracy in test data prediction (36.3% in the Kaggle), feature engineering and error analysis applied, together with different implemented classifiers used. They all described in the following sections.

2 Data Representation and Learner Choice Prediction

The datasets provided are divided into three parts:

- "train": to build a classifier model.
- "dev" (development): to test and perform the evaluation of the built model.
- "test": hand in the final prediction of this dataset as a part of final project score.

The ratio of "train" to "dev" is approximately 3:1, which is an acceptable Hold-out strategy ratio. Each of the datasets has further split into two categories ("raw" and "top-n"), which utilised different approaches to processing.

The datasets provided have been extracted from different years of Twitter contents. Each dataset class label has a uniform distribution with four classes, which means there is roughly 25% accuracy if use 0-R method to assign a class label. For a multi-class classification problem, there are some direct classifier (Naive Bayes, K-NN), and other classifiers are binary, which need to apply One-vs.-all Strategy. Besides, with a large number of features and instances, the time costs of training and testing, together with accuracy, are aspects that need to consider. Accordingly, classifiers are narrowed down to Naive Bayes. Some other classifiers, like Random decision forest (RF) and Stochastic gradient descent classifier (SGDClassifier), will be explored.

2.1 Multinomial Naive Bayes

Naive Bayes was considered since it works fast for text samples with simplified probability calculations (McCallum and Nigam, 1999). Also, with the data representation of training datasets, which are all integer frequency counts, Naive Bayes with joint probability is a possible approach. For hyperparameters, multinomial distribution in frequency counts was determined rather than Bernoulli distribution, as well as add-one smoothing method was chosen. However, this model assumes sample features independent and performs poorly on samples with strong correlation. Thus it was chosen as an adequate benchmark.

2.2 Stochastic gradient descent classifier

Stochastic gradient descent classifier (SGDClassifier) is a collection of algorithms that use gradient descent to find the optimal model for classification and reach results. The significant advantage of this model is its efficiency in classify text documents using sparse features.[1] However, SGDClassifier requires many hyperparameters, and it is sensitive to feature scaling. GridSearch was applied to adjustment and find out the best hyperparameter, which is

[1] Classification of text documents using sparse features: https://scikit-learn.org/stable/modules/sgd.html

figure 1 indicated.

```
Performing grid search...
pipeline: ['tfidf', 'clf']
parameters:
{'clf__alpha': (1e-05, 1e-06, 0.0001),
 'clf__max_iter': (1000, 2000, 3000, 500),
 'clf__penalty': ('l2', 'elasticnet', 'l1'),
 'tfidf__max_features': (1000, 5000, 10000, 30000, 50000, None),
 'tfidf__norm': ('l1', 'l2', None),
 'tfidf__stop_words': ('english', None),
 'tfidf__use_idf': (True, False)}
Fitting 3 folds for each of 2592 candidates, totalling 7776 fits
done in 31244.328s

Best score: 0.504
Best parameters set:
        clf__alpha: 1e-05
        clf__max_iter: 2000
        clf__penalty: 'l2'
        tfidf__max_features: None
        tfidf__norm: 'l2'
        tfidf__stop_words: 'english'
        tfidf__use_idf: True
```

Figure 1: Output: Grid Search for SGDClassifier and TfidfVectoriser

3 Top-N Results

3.1 Data

The datasets provided in this part contain top n words as features, extracted using Mutual Information and Chi-Square from the raw dataset. After further inspect with top-n datasets, it shows that there are a considerable amount of instances labelled as zero, 77.6% in top-100 dataset for example. Moreover, in all zero instances, four location classes have a uniform distribution, which adds difficulties to the chosen classifiers.

3.2 Evaluations and Error Analysis

After applying Hold-out (3:1) evaluation using "dev" datasets (Table 1) and 10-fold Cross-validation evaluation using "train" datasets (Table 3), surprisingly, the "benchmark" Multinomial Naive Bayes (MNB) has the best performance. Even so, only 30.80% accuracy can be found in maximum, and if look at Table 2, class Sydney has high recall while relatively low precision. It can be interpreted as MNB classifier took the other three cities as Sydney, even though it also labelled Sydney instances correctly. Those findings indicated that with a large number of instances with zero value, apply any useful information for class labelling can be unrealistic. Classifiers just randomly assigned labels to those instances. This assumption is further confirmed in the training of top-50 and top-100: the prediction accuracy is slowly rising with the increasing of features' number. This fact gives an inspiration, with more features given in training datasets, the performance of prediction accuracy will be better, which leads to preprocessing of raw data in later sections.

The differences between the two table's accuracy imply the models are overfitting to the training dataset. It may due to the data extracted from different years, and Twitter users' language behaviour might change from time to time. The model for "train" may not have strongly fit in "dev".

Classifer	Top-10	Top-50	Top-100
MNB	29.44%	30.13%	30.80%
RF	29.44%	30.11%	30.77%
SGD	29.46%	30.13%	30.78%

Table 1: Accuracy using Top-N dev datasets

Class	Precision	Recall
Sydney	0.27	0.91
Melbourne	0.57	0.10
Perth	0.63	0.09
Brisbane	0.49	0.31

Table 2: Classification report for MNB Top-100

Classifer	Top-10	Top-50	Top-100
MNB	34.44%	37.13%	42.50%
RF	30.44%	32.11%	33.21%
SGD	33.46%	37.13%	40.12%

Table 3: Accuracy using Top-N train dataset with 10-fold cross validation

4 Feature Engineering

4.1 Data Preprocessing

The text of the tweets includes emojis, websites address, and other contents that ineffective for location identification, to improve the efficiency of feature selection, preprocessing of raw data must be done as follow steps:

- Using regular expressions to remove websites, Unicode (including emoji and other language's special characters), usernames (which followed by "@"), numbers and punctuations.
- Replace camel case words and abbreviation words with regular words.
- Remove English stopwords by using "nltk" library.
- Perform tokenisation and Lammitization.

4.2 Feature Extraction and Selection

After the above preprocessing the number of features reduced from 16w (without preprocessing, directly tokenise and get feature numbers) to 6w. Print out feature's name, and there is still some feature that cannot provide valuable information for classification, such as "aaaaaaaa". TF-IDF (Term Frequency-Inverse Document Frequency) vectoriser was introduced to further select useful features. After this using TF-IDF vectoriser, the number of features reduces to 4.8w (around 80% before).

The hyperparameters including smooth by adding one to document frequencies, default is True, and when turning it off, zero divisions will occur. Another mentionable hyperparameter is the number of n-grams to be extracted, since the nature of language is that increased vocabulary store more information, an initial assumption is that two grams perform better than unigrams, however, after several trials, it shows that unigrams produce higher accuracy than two-grams (approximately 2% higher using SDGClassifer). One reason can explain that is the number of features growth exponential when choosing more than two grams, which may produce useless features. GridSearchCV applied to find the rest of the best hyperparameters, seeing in figure 1 above.

5 Final Results

5.1 Evaluation

This turn SGDClassifier has the best performance of three classifiers, with the improvement of 3.82% compared with Top-100 result. Other two classifier's accuracies arose around 3% as well, which confirmed the inspiration in initial error analysis that increasing the number of features will give those classifiers better performance.

The potential reason for MNB has moderate performance in this round is that MNB's nature assumes all features are independent. In the final dataset with a large number of features like 4.8w, the influence of dependent feature will be far higher than the dataset with just 100 features like top-100 dataset, which can be the significant reason to cause moderate performance.

Classifier	Accuracy
MNB	34.95%
RF	32.83%
SDGClassifier	35.34%

Table 4: Accuracy with final dataset

Class	Precision	Recall
Sydney	0.34	0.34
Melbourne	0.36	0.34
Perth	0.35	0.33
Brisbane	0.34	0.37

Table 5: Classification report for SDG final dataset

5.2 Error Analysis

There are 4.8w features (top 80% of original preprocessing features)used to reach the final results, which is the most optimal choice after running several numbers of features like top 10%, 30%, 50%, 80%, 100% features. Despite this, accuracy still cannot reach a satisfactory level. The low macro averaging precision and recall show the unreliable of the original dataset. This result may be explained by the nature that tweets, the users may talk about one place while they actually in other places. For example, a tourist came back to Melbourne after he travelled around Sydney, he posted a tweet that "I love Sydney, that was a pleasant journey." with his geotag in Melbourne. After extract information from tweet content, this tweet is labelled as Sydney class while the actual location in Melbourne.

6 Conclusions and Potential Improvements[2]

As the above sections discussed, MNB has the best performance in a dataset which has relatively small features, while when features number rise, SDGClassifier has the highest performance. The neural network and word embeddings might be introduced in future studies to address the grammatical problems (Moon and Liu, 2016) mentioned in final error analysis, as well as consider combining "train" and "dev" datasets and perform k-fold cross-validation or raised the training data's size to improve the efficiency of the training process.

[2]This report's words count is 1325 exclude references

7 References

McCallum, A. and Nigam, K. (1999). A Comparison of Event Models for Naive Bayes Text Classification. [online] Available at: https://www.cs.cmu.edu/ knigam/papers/multinomial-aaaiws98.pdf [Accessed 20 May 2019].

Tim Moon and Eric Liu. (2016). Using feedforward and recurrent neural networks to predict a bloggers age.