Twitter Geotagging Classifier Report

COMP30027 Assignment 2 Report

May 27, 2019

1 Introduction

Geotagging is one of the most complex problem that currently exist. While its use is quite controversial, it is still essential in today digital world for the purpose of providing great service to users in various applications and more. In solving this challenge, many has taken a look at using Machine Learning and the large amount of data that user generates every single day. The task in this project is to create a classifier that can identify user's location solely based on their tweets.

2 Related Literature

A very similar journal which was written by Pappas, Azab, and Mihalcea (2018) has made efforts in the solving the same problem. It focused on geotagging user solely based on posts both in Twitter and blogs. It managed to gain good results through their custom feature selection method and their choice of classifiers.

3 Methods

3.1 Preprocessing

The raw data for training that was provided have three features in total which are ID, user's location, and the tweet itself. As this is not very useful, preprocessing is necessary. The first step is to tokenize the tweets and extracting the important words.

3.1.1 Tokenization

The main idea of the tokenization is to extract the important words in a tweet. The goal is to identify a tweet's topic. Tweets are short texts in nature and users must pack a lot of information in a very short text to get their message across. Identifying topic can be nearly replicated by taking the keywords of each tweets. There are many factors as to why users use certain keywords, user's location can be one of them. By extracting keywords, and therefore topics, it is possible to find correlation between user's keywords and their location. This is done by looking through their choice of nouns and "hashtags". Nouns in sentences can give context and therefore determine topics. Hashtags is an important system that Twitter use to determine trends and it can be used to determine topic. The technologies that are used for the tokenization is NLTK (Natural Language Toolkit) and TextBlob. Below are the steps involved.

1. First, all URLs and unicode characters are removed through replacement using Python's regex library.
2. Then, the Tweet tokenizer from NLTK is used. This tokenizer has the feature to remove certain parts of a tweet such as name mentions and reduce longer words like "hellooooooo" to "hello". The extracted words are then concatenated again for further extraction by another tokenizer. Note that name mentions are preserved as they are important.
3. Hashtags which are crucial are preserved in extractions.
4. TextBlob is then used to tokenized the tweet even further. TextBlob has an important feature of extracting only noun phrases. This means it also removes all the unimportant tokens such as stop words.
5. Each noun phrases are also lemmatized to its base word so that it reduces features.
6. All the keywords are then returned and processed further

The main reason TweetTokenizer and TextBlob are used is because of their flexibility. Tweets can be very unpredictable and its usage of language can be wildly inconsistent. Tokenizing solely based on known words can result in loss of important tokens as new words can be trending and it is possible that the word is not part of the known English words. Due to its flexibility, unknown words are preserved and will be processed in the next step which is feature selections.

3.1.2 Feature Selection

After keywords are extracted, each word is then set as a feature for each instance. Then, the frequency for each word for each instance are calculated. This resulted in thousands of features for each instance. Therefore, to create a better performing classifier, feature selection is necessary. The selection method used is to remove words that do not appear at least a certain number of times, in this project, the selected threshold is 10. This method was also used by Pappas et al. (2018, p. 11). There are many benefits to this method as it reduces features which increases performance, removing mistakes in words as TextBlob can be very lenient, and only take trending words which are important. Finally, this data is then stored into a file for later processing.

3.2 Classifications

After the data is preprocessed, it can be used to train the classifiers. There are numerous options for classifiers. A Decision Tree was the first choice as whether or not a certain word appear in a tweet can decide user's location. Naive Bayes classifiers also have shown promise in classifications (Pappas et al., 2018, p. 12). All classifications uses the same training data and are evaluated against the dev-raw file. Accuracy is calculated by counting correct guesses and dividing it with total instances. Table 1 shows the accuracy.

Classifier	Accuracy
MultinomialNB	0.30338192732340014
DecisionTree	0.3034623217922607
GaussianNB	0.296387608532533

Table 1: Accuracy without counting how many users use word

There is another variable that is not included during preprocessing, which is how many users use the same words. This is because, interestingly, adding this variable makes the classifier scores slightly lower. By adding this to the preprocessing which is to remove words that are not used at least by a certain number of users, which in this case is 10, a new score against the same data is created. Table 2 shows the scores which also involve counting how many users use a word.
For Bernoulli Naive Bayes Classifier as it only accepts binary features, it requires different preprocessing. Preprocessing remains the same with only

Classifier	Accuracy
MultinomialNB	0.30316754207310537
DecisionTree	0.30322113838567905
GaussianNB	0.2960660306570908

Table 2: Accuracy with counting how many users use word and each word's usage frequency

difference being that it does not count frequency. Value is set to 1 if a word exists in the tweet and 0 when it does not. Table 3 shows the scores. Decision Tree also included for comparison.

Classifier	Accuracy
BernoulliNB	0.3074820452352878
Decision Tree	0.3034623217922607

Table 3: Setting each features to be binary

4 Evaluation

From all the classifiers, when tested against dev-raw.tsv, it is clear that preprocessing data into binary features and classifying using Bernoulli Naive Bayes has the highest accuracy. Interestingly however, this does not always work at other cases as when it is tested against other data (test-raw.tsv), it scores 0.29244 which is relatively low compared to other classifiers. Of all the classifiers, Multinomial Naive Bayes classifiers with the preprocessing method of just calculating word frequency (Table 1) has the highest score when tested against test-raw.tsv with score 0.30205. This might be due to the fact that, Multinomial Naive Bayes classifier performs well to this type of data which are discrete.

Another interesting finding is that when data is preprocessed without calculating how many users use each words (Table 1), classifiers tend to score slightly higher than using data that is preprocessed with removing words that is not used by at least a certain number of users and at least a certain number of times (Table 2). Counting the number of users that use each word is a condition for filtering features that is introduced by Pappas et al.

(2018, p. 11). The difference however is quite insignificant and it might be caused by the test and training data as there are benefits in including the number of users such as eliminating tweets that has the same word many times.

5 Error Analysis

These findings however are prone to errors. The training data is very large and it is near impossible to check and filter each tweet. Listed below are some of the possible mistakes.

1. As mentioned before, due to the preprocessing methods, tweets that have the same word used many times can skew the result
2. A worldwide trend can affect results highly and makes it really hard to locate user, due to the limited scope of the locations.
3. Multinomial Naive Bayes classifier assumes that features are independent and it will perform badly when features are not independent. As tweets tend to be short texts, the keywords in each tweet are likely to be dependent.

6 Conclusions

In conclusion, it is possible to geotag users solely based on tweets. There are various important variables that can determine a user's location, mainly the important words in a tweet which are nouns and hashtags. The frequency of these keywords is an important data that can be used to classify user's location. The frequency of users that use a certain keyword is also important and it can affect the results.

References

Pappas, K., Azab, M., & Mihalcea, R. (2018). A comparative analysis of content-based geolocation in blogs and tweets. *CoRR*, *abs/1811.07497*. Retrieved from `http://arxiv.org/abs/1811.07497`