COMP90049 Knowledge Technologies
Project 2: Geolocation of Tweets with Machine Learning

Anonymous

1 Introduction

Every day, around 500 million tweets are tweeted on Twitter[1]. By using the location information of the tweets, plenty of location-based services can be promised to the users. However, only a few of tweeters will use the geospatial features in the Twitter, which limits the reach and impact of the location-based sensing system. To overcome this problem, in this report, we build a geolocation classifier for Tweets based on a simple machine learning algorithm. In our classifier, we only use the content of the user's tweets.

2 Ralated Work

In the last decade, many people have proposed methods for predicting the geographical scope of different contents.(Fink et al., 2009; Lin and Halavais, 2004; Amitay et al., 2004; Backstrom et al., 2008; Hurst et al., 2007) Most of their prediction is based on geographically related terms. Zhiyuan et al. in 2010 proposed a prediction method purely based on tweet content.(Cheng et al., 2010) Our method uses some of the theories of them.

3 Dataset

In this report, a tweet dataset (Eisenstein et al., 2010; Rahimi et al., 2018) is used for improving and evaluating the method. The features used in this report is contained in the dataset. There are two different kinds of it. The Most-N includes the term frequency for the top N terms and the Best-N consists of the term frequency for the terms with the greatest Mutual Information and Chi-Square values.

[1]Data from: https://www.internetlivestats.com/twitter-statistics

4 Methodology

4.1 Feature Engineering

4.1.1 Feature Selection

In our research, we use Chi-square and Mutual Information to help remove unrelated attributes.

4.1.2 Smoothing

Since a single tweet only contains a small set of terms, the sparsity problem among our features is severe (many words' term frequency equal to zero). To overcome this problem, we assume that the tweets of a given user are from the same location. We combine all the tweets of the same person to a corpus and extract features from this corpus.

4.2 Baseline

4.2.1 Zero-R

Zero-R classifies all instances according to the most common class in the training dataset. Because we have different sets of features and do not know which is better, it would be better to build a baseline without considering the features at the beginning.

4.3 Simple Probability-Based Location Estimation Method

In our report, we try to use the Simple Probability-Based Location Estimation method refer to a paper (Cheng et al., 2010). Given a tweet, we can simply calculate the probability of it coming from the city i based on maximum likelihood estimationas:

$$p\left(i|S_{tokens}(u)\right) = \sum_{w \in S_{tokens}(u)} p(i|w) * p(w)$$

where $S_{tokens}(u)$ is a set of tokens extracted from a given user u's tweets, $p(w)$ is the probability of the word w in the whole dataset and $p(i|w)$ identifies for each word w the likelihood that it was issued by a user located in city

i. Letting $count(w)$ be the number of occurrences of the word w, and t be the total number of tokens in the corpus, we replace $p(w)$ with $\frac{count(w)}{t}$. Also, to overcome the sparsity of words across different users and locations, we use the Laplace smoothing mathod to Smooth the words distributions, which is defined as:

$$p(i|w) = \frac{1 + count(w, i)}{V + N(w)}$$

where $count(w, i)$ is is the count of w in city i, V is the size of the vocabulary and $N(w)$ is the count of w in all the cities. We would call it SPLE in the rest of the report.

4.4 Machine Learning Method

Judging a city from which a tweet comes can be a classification problem. Since we have a fixed set of class, we can focus on comparing the effectiveness of supervised classifiers.

4.4.1 K-Nearest Neighbour

K-Nearest Neighbor classifiers classify the input accounding to the majority class of the k nearest training instances. We use $k = 20$ in our method.

4.4.2 Naive Bayes

Naive Bayesian classify the input based on Bayesian Theorem. Considering the probability of the occurrence of a word on Twitter may be related to the city, this method may work well in predicting the location of a given tweet. There are three kinds of Naive Bayesian: Gaussian Naive Bayes assumes that the prior probability of features is a normal distribution; Multinomial Naive Bayes assumes that the prior probability of features is a polynomial distribution; Bernoulli Naive Bayes assumes that the prior probability of features is a binary Bernoulli distribution;

4.4.3 Decision Tree

Decision tree is a basic classification and regression method. By building a tree-like model of decisions and their possible consequences, it can decide which city a tweet is from.

4.5 Evaluation Method

4.5.1 Evaluation Metrics

Since we think the effectiveness of our method depends on both precision and recall, we mainly consider the F1 Score of the methods we choose. The F1 Score is the harmonic mean of recall and precision. The formulas of them are list below:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

4.5.2 10-Fold Cross-validation

10-Fold cross-validation splits our dataset into ten partitions, and the evaluation metric is aggregated across ten iterative times.

5 Result and Analysis

5.1 Result of Baseline Method

The result of Zero-R is shown in Table 1. Since Zero-R only considers the primary class in the dataset, we can find that about 63% of the tweets in our dataset is from NewYork.

Method	F1 Score
Zero-R	62.960%

Table 1: The F1 Score of Zero-R

5.2 Result of SPLE

The result of SPLE is shown in Table 2. Although SPLE uses most of the content of the given tweets, the result of it is terrible. A lot of irrelevant words are contained in the prediction process. But in fact, some specific words contribute a lot to the judgment of the user's geographical location. In this method, we treat all words in the same way, which would not bring us a good result.

Method	F1 Score
SPLE	63.339%

Table 2: The F1 Score of SPLE

5.3 Smoothing

In Table 3, we use features in Best-200 to generate the result by Bernoulli Naive Bayes classifier. We find out that with smoothing, the F1 Score improves a lot. One reason is that we get more information to predict the location of a giving tweet. The other reason is that smoothing help solves the sparsity problem we talk about in section 4.1.2. However, the side

effect of this method is that it reduces the number of our samples since there is only one sample left for each user.

Method	F1 Score of Train	F1 Score of Test
Before	0.6533	0.6454
After	**0.7986**	**0.7649**

Table 3: The F1 Score Before and After Smoothing

5.4 Different Features and Classifiers

To find the best features set and the best classifer, we calculate the F1 Score of using three different classifiers to predict the **test set** based on different features sets. Classifiers include Decision Tree, Bernoulli Naive Bayes and KNN. The result is shown in Table 4, and all the data has been smoothed.

5.4.1 Classifiers Comparison

The result shows that the F1 Score of Naive Bayes Classifier is better than the other two. This is reasonable because it is said that there is a subset of words that have a more compact geographical scope compared to other words in tweets (Cheng et al., 2010). People from different cities have different preferences for different words. It is the probability of some words' frequency contributes a lot to where the tweet is from. Also, there is no hierarchical relationship between features, which means the Decision Tree can not work well. KNN has the worst effect because the task is not a Nearest-Neighbour problem. Besides, with a large number of features, the similarities are mostly meaningless.

5.4.2 Features Sets Comparison

The result also shows that using the Best-N feature sets can have a better outcome than using the Most-N feature sets. This is because the features selected by Mutual Information and Chi-Square values are more correlated with the class we want to predict. And by comparing features sets with different N, we can find that a bigger N can lead to a higher F1 Score. This is because some tiny features may remain important information about the tweet's location, and more features remain more information.

5.5 Different Naive Bayes Methods

To find the most suitable Bayesian method, we compare the F1 Score of tree different Bayesian methods. The result is based on Best-200 features set and all the data has been smoothed. The outcome is shown in Table 5. The result of Bernoulli Naive Bayes is better because it ignores the frequency of the occurrence of a word, only considering whether words appear or not. This is equivalent to smoothing the data, even though it lost some of the information.

Features	DF	NB	KNN
Most-10	0.4718	0.6003	0.5971
Most-20	0.4780	0.5956	0.5909
Most-50	0.5094	0.5564	**0.6003**
Most-200	0.5438	0.5831	0.5956
Best-10	0.6353	0.7487	0.6580
Best-20	0.6379	0.7539	0.6144
Best-50	0.6552	0.7654	0.6018
Best-200	0.6504	**0.7649**	0.5956

Table 4: The F1 Score of Different Features and Classifier

Distribution	F1 Score of Train	F1 Score of Test
Gaussian	0.7940	0.7605
Bernoulli	**0.7986**	**0.7649**
Multinomial	0.7798	0.7602

Table 5: The F1 Score of Different Naive Bayes

5.6 Inadequacies and Improvement

Through the overall analysis, we finally choose the Bernoulli Naive Bayes classifier. The overall accuracy is not too high because we assume that all of our features are independent, but they are not actually. Terms like *ahaha*, *ahahah* and *ahahaha* are relevant. Besides, we reduced the number of samples in the smoothing process and lost part of the frequency information during the process of using Bernoulli Naive Bayes. To improve our method, we can use some approximate matching methods like Jaro-Winkler Similarity to combine some of the features. We can also use Bagging to get more data and use ensemble learning to combine some classifiers and provide a better result.

6 Conclusions

In our report, we try to use a variety of machine learning methods and different feature sets to predict the location of tweets. The prediction is purely based on the content of tweets.

Through research, we find that combining all the tweets of the same person for the forecast can have higher accuracy. Besides, we find that using features selected by Mutual Information and Chi-Square values and a broader set of features can lead to better prediction performance. And through comparison, we find that using the Bernoulli Naive Bayes classifier can achieve a good classification effect in this task. We also proposed an improvement plan based on our shortcomings at the end.

References

E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280. ACM, 2004.

L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak. Spatial variation in search engine queries. In *Proceedings of the 17th international conference on World Wide Web*, pages 357–366. ACM, 2008.

Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768. ACM, 2010.

J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1277–1287. Association for Computational Linguistics, 2010.

C. Fink, C. D. Piatko, J. Mayfield, T. Finin, J. Martineau, et al. Geolocating blogs from their textual content. In *AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0*, pages 25–26, 2009.

M. Hurst, M. Siegler, and N. S. Glance. On estimating the geographic distribution of social media. In *ICWSM*, 2007.

J. Lin and A. Halavais. Mapping the blogosphere in america. In *Workshop on the weblogging ecosystem at the 13th international World Wide Web conference*, volume 18, pages 1–7, 2004.

A. Rahimi, T. Cohn, and T. Baldwin. Semi-supervised user geolocation via graph convolutional networks. *arXiv preprint arXiv:1804.08049*, 2018.