Geolocation Classifier for Tweets

1 Introduction

People from different countries or cities use language differently. The spelling of some words in British English is different from in American English. Also, dialects and slangs are different from regions to regions. So, it is possible to infer a person's home geolocation by looking at what (s)he writes and how (s)he writes. This project aims to build a Geolocation Classifier Model, which able to classify a user's geolocation into one of the three cities (New York, California or Georgia) with decent accuracy. The dataset used in this project is the tweets dataset [1][2], which includes 96585 instances from the training set and 34028 instances from the development set. The software used in this project is Waikato Environment for Knowledge Analysis (Weka) [8].

2 Literature review

Some previous works have been done in this area. In 2010, Eisenstein Jacob [1] develop a model, which assumes the topic varies from regions to regions, to identify words that have "high regional affinity". In 2018, Afshin Rahimi [2] proposed three semi-supervised models (GCN, DCCA, and MLP-TXT+NET) to identify a user's location, which achieved some good results.

3 Method

First, we select a dataset for analysis. We use bestXX instead of mostXX since bestXX contains the features with the greatest Mutual Information and Chi-Square values, while mostXX contains the most frequent terms, which do not indicate a location.

Second, we select classifiers that are suitable for this project. Since the problem is a classification problem as we need to classify the tweet instances into one of the three cities (multiclass). Also, it is Supervised Learning as we have the desired output class in our training set and development set. ZeroR, OneR, J48, LMT, Random Forest, Naïve Bayes, Naïve Bayes Multinomial and Logistic Regression are selected for analysis. Although some of them are binary classifiers by definition, such as Logistic Regression, sometimes they work well with the multiclass problems.

Third, base on the previous result, we sum the vectors with the same user id in the dataset, which reduced the number of instances and run the selected classifiers again using 10-Fold Cross-validation.

3.1 ZeroR

ZeroR is a simple classifier, which always predicts the majority class. In this project, ZeroR is used as a baseline.

3.2 OneR

OneR uses one attribute (feature) which has the lowest error rate for prediction. In this project, OneR is also used as a baseline.

3.3 K-NN

K-Nearest Neighbors (k-NN) [3] compared one instance to other instances and classify the instance by looking at its K nearest neighbors. A common way of choosing K value is K = sqrt(n).

3.4 J48

J48 (C4.5) is a decision tree algorithm (by Ross Quinlan [4]) for classification.

3.5 Random Forest

Random Forest [5], which randomly selects features and construct different decision trees, uses the results (voting) from a collection of trees to make predictions.

3.6 Naïve Bayes and Naïve Bayes Multinomial

Naïve Bayes [6], which based on Bayes' theorem and assume each feature is independent of other features, is a way of classification based on the probabilities. Naïve Bayes Multinomial uses multinomial distribution for each feature.

3.7 Logistic Regression

Logistic regression [7], which uses the Logistic Model (a math function of the logarithm), has an 'S' shape logistic regression curve range from 0 to 1, which can be used to make predictions based on the probabilities.

3.8 LMT

logistic model tree (LMT) [12] is based on the decision tree (C4.5) and logistic regression.

4 Measurement

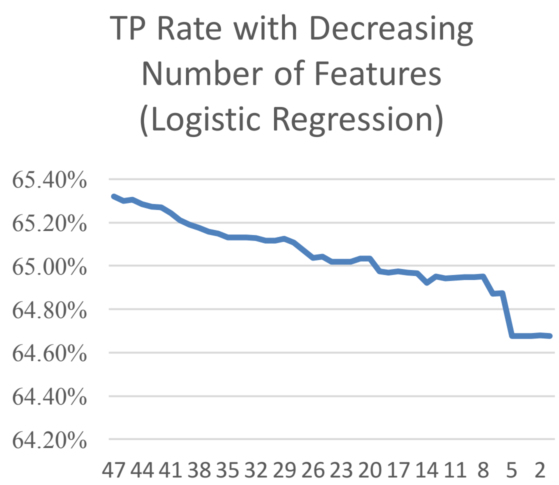
There are several measurements provided by Weka: TP Rate, FP Rate, Precision, Recall, F-Measure, MCC, ROC Area, and PRC Area. In this project, we mainly look at Weighted TP rate [9] (Percent of correctly classified instances), Receiver Operating Characteristic (ROC) [10] Area and Precision-Recall Curves (PRC) Area because TP rate indicates accuracy while ROC and PRC Area indicate benefits of the algorithm compared to the baseline.

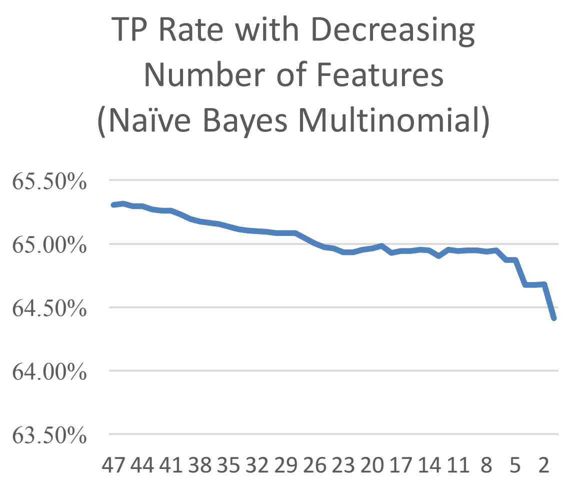
Experiment 1: Results of best 10, 20, 50 and 200 data with tweet-id and user-id removed (Hold-out)								
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
ZeroR	0.644	0.644	-	0.644	-	-	0.500	0.479
OneR	0.647	0.633	-	0.647	-	-	0.507	0.483
Best 10								
K-NN(185)	0.649	0.628	-	0.649	-	-	0.560	0.524
J48	**0.651**	0.620	0.658	0.651	0.529	0.114	0.517	0.489
Random Forest	**0.651**	0.620	0.645	0.651	0.529	0.113	**0.562**	**0.527**
Naïve Bayes	0.632	0.612	0.513	0.632	0.519	0.045	0.550	0.505
Naïve Bayes Multinomial	**0.651**	0.619	0.650	0.651	0.530	0.115	**0.561**	**0.527**
Logistic Regression	**0.651**	0.620	0.655	0.651	0.529	0.114	**0.563**	**0.528**
LMT	**0.651**	0.620	0.656	0.651	0.529	0.114	**0.563**	**0.528**
Best 20								
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
K-NN(185)	0.649	0.629	0.616	0.649	0.520	0.084	0.583	0.542
J48	**0.653**	0.615	0.657	0.653	0.533	0.130	0.521	0.493
Random Forest	0.650	0.609	0.603	0.650	0.535	0.115	0.583	0.542
Naïve Bayes	0.621	0.590	0.508	0.621	0.527	0.056	0.564	0.515
Naïve Bayes Multinomial	**0.653**	0.613	0.647	0.653	0.535	0.130	**0.588**	**0.550**
Logistic Regression	**0.653**	0.614	0.652	0.653	0.535	0.130	**0.589**	**0.551**
LMT	**0.653**	0.615	0.657	0.653	0.534	0.130	**0.588**	**0.550**
Best 50								
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
K-NN(185)	0.648	0.629	0.613	0.648	0.519	0.081	0.608	0.564
J48	**0.651**	0.598	0.601	0.651	0.544	0.133	0.549	0.511
Random Forest	0.636	0.576	0.549	0.636	0.547	0.107	0.593	0.551
Naïve Bayes	0.601	0.550	0.513	0.601	0.535	0.071	0.574	0.525
Naïve Bayes Multinomial	**0.654**	0.593	0.620	0.654	0.549	0.148	**0.623**	**0.583**
Logistic Regression	**0.656**	0.597	0.626	0.656	0.547	0.151	**0.624**	**0.584**
LMT	-	-	-	-	-	-	-	-
Best 200								
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
K-NN(185)	0.648	0.631	0.532	0.648	0.518	0.074	0.626	0.581
J48	-	-	-	-	-	-	-	-
Random Forest	-	-	-	-	-	-	-	-
Naïve Bayes	0.582	0.514	0.515	0.582	0.538	0.083	0.579	0.532
Naïve Bayes Multinomial	**0.654**	0.552	0.603	0.654	0.572	0.178	**0.662**	**0.618**
Logistic Regression	**0.658**	0.571	0.616	0.658	0.565	0.176	**0.661**	**0.618**
LMT	-	-	-	-	-	-	-	-

Table 1 - Results of best 10, 20, 50 and 200 data with tweet-id and user-id removed

Experiment 2: Information Gain Ranking for train-best20 Features								
No	IG	Attribute	No	IG	Attribute	No	IG	Attribute
1	0.005166	haha	17	0.001222	dead	33	0.000702	gw
2	0.002894	inhighschool	18	0.001187	dat	34	0.000669	flirty
3	0.002771	lmaoo	19	0.001135	atlanta	35	0.000643	san
4	0.002493	lml	20	0.001104	iight	36	0.000629	ahaha
5	0.002183	hahaha	21	0.000964	will	37	0.000619	coo
6	0.002007	da	22	0.000958	dis	38	0.000607	thatisall
7	0.001982	hella	23	0.000945	deadass	39	0.000596	lowkey
8	0.001932	lmaooo	24	0.000944	willies	40	0.000555	famu
9	0.001757	rt	25	0.000925	just	41	0.000517	frequency
10	0.001709	the	26	0.000911	finna	42	0.0005	juss
11	0.001542	and	27	0.00091	ga	43	0.000498	gsu
12	0.00154	ii	28	0.000832	la	44	0.000498	tinos
13	0.001421	are	29	0.000825	a	45	0.00049	parody
14	0.001287	atl	30	0.000799	know	46	0.000452	famusextape
15	0.001287	that	31	0.00072	bomb	47	0.000424	wet
16	0.001261	smh	32	0.00072	childplease			

Table 2 - Information Gain Ranking for Best 20 Features (tweet-id and user-id removed)





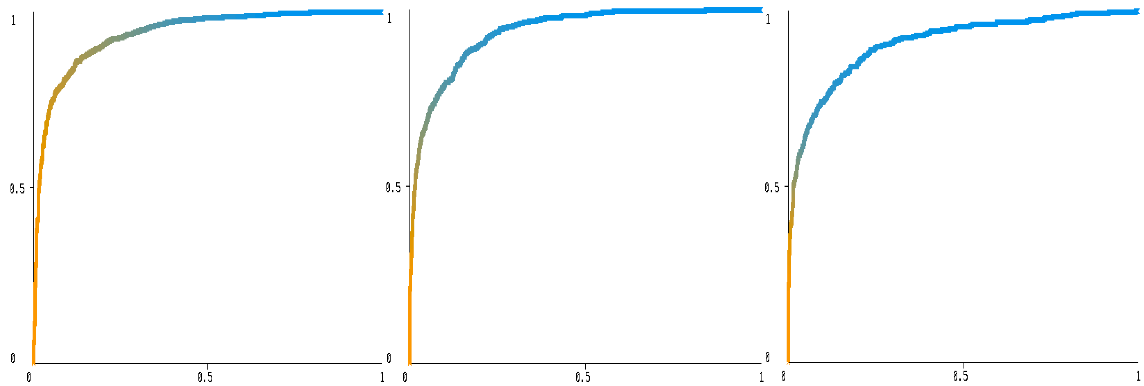
Graph 1 - TP Rate with Decreasing Number of Features (Logistic Regression and Naïve Bayes Multinomial)

Experiment 3: Results (best200 (train+dev), 10-Fold Cross-validation)								
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
ZeroR	0.633	0.633	-	0.633	-	-	0.500	0.470
OneR	0.636	0.622	-	0.636	-	-	0.507	0.474
Naïve Bayes Multinomial	0.653	0.533	0.617	0.653	0.574	0.206	0.678	0.627

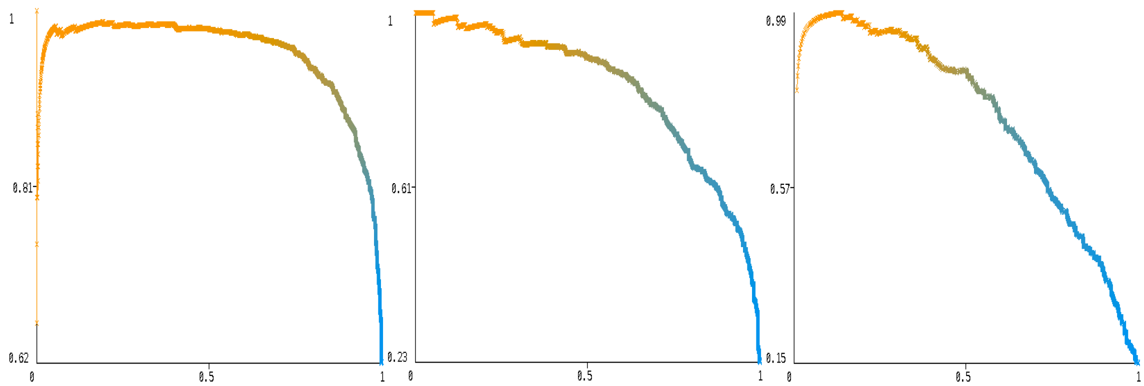
Table 3 - Results of Naïve Bayes Multinomial (10-Fold Cross-validations on all data)

Experiment 4: Results (best200 (train+dev), 10-Fold Cross-validation, summed Attribute Vectors by user_id)								
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
ZeroR	0.624	0.624	-	0.624	-	-	0.498	0.463
OneR	0.651	0.536	-	0.651	-	-	0.558	0.501
K-NN(17)	0.661	0.549	0.731	0.661	0.563	0.258	0.791	0.710
J48	0.716	0.236	0.713	0.716	0.715	0.483	0.735	0.635
Random Forest	0.715	0.431	0.737	0.715	0.664	0.411	0.895	0.840
Naïve Bayes	0.657	0.126	0.761	0.657	0.675	0.507	0.862	0.782
Naïve Bayes Multinomial	**0.766**	0.111	0.804	0.766	0.775	0.621	**0.897**	**0.857**
Logistic Regression	**0.789**	0.166	0.790	0.789	0.790	0.622	**0.891**	**0.841**
LMT	**0.828**	0.189	0.823	0.828	0.822	0.674	**0.930**	**0.893**

Table 4 - Results of 10-Fold Cross-validation on all data (summed attribute vectors by user_id)



Graph 2 - ROC Areas (New York: 0.935, California: 0.934, Georgia: 0.905)



Graph 3 - PRC Areas (New York: 0.953, California: 0.831, Georgia: 0.734)

5 Result

For the first experiment, we removed twee-id and user-id from the dataset as they would not reappear in the testing set and used 8 models for analysis. From the results in Table 1, we can see some classifiers perform better than the baseline. Logistic Regression, Naïve Bayes Multinomial and Logistic Model Tree have the best TP Rate, ROC Area and PRC Area. Random Forest, J48 and K-NN(185) are slightly better than the baseline while Naïve Bayes is worse than the baseline. So, for the second experiment, we mainly focus on Logistic Regression, Naïve Bayes Multinomial and Logistic Model Tree.

For the second experiment, we first analyzed the information gain (IG) [11] of each attribute, rankings can be seen from table 2. Then we remove the attribute with the lowest IG one by one (e.g. ‘wet’, ‘famusextape’, ‘parody’ etc.) and calculate TP Rate for each attributes set. The results are shown in graph 1. We can see TP Rate is positively correlated with the Number of Features and training instances. So, removing attributes with less information gain may improve Over-fitting models, but not helpful to improve accuracy since the attributes are selected using Mutual Information and Chi-Square values.

For the third experiment, we combined the train-best200 data with dev-best200 data, which contains 130613 instances in total. Then, we use the 10-Fold Cross-validation to train our model. We can see the results from table 3 that TP Rate improved by 2%, ROC Area improved by 17.8% and PRC Area improved by 15.7%.

For the fourth experiment, we sum the attribute vectors by the same user id in the dataset because there are too many ‘0’s in our dataset, which is not good for building models. After we sum the vectors, the number of instances get reduced to 3190 (train+dev), so I chose to use 10-Fold Cross-validations for analysis. Since we sum the attribute vectors, most attributes would be non-zero, so we choose to use the best200 dataset. The result can be seen from table 4 and ROC/PRC Arear can be seen in graphs 2 and 3. We can see the result got improved a lot for most classifiers. LMT has the highest TP rate (82.8%), which is 20% more than the baseline. Also, LMT has 93.0% ROC Area and 89.3% PRC Area, which is surprisingly good. Logistic Regression and Naïve Bayes Multinomial have solid performance as well.

Apart from that, we did some experiments with Naïve Bayes Multinomial and J48. We combined NBM with other models using ensembling, boosting and bagging, but the results are not ideal. Furthermore, we used J48 for feature selection. We removed ‘wet’, ‘gsu’, ‘ii’, ‘lmao’, ‘lmaooo’, ‘lml’ and ‘parody’ from the best10 dataset and the weighted TP Rate got improved a little bit, but at the cost of ROC Area and PRC Area. So, this approach might not be suitable as well.

6 Discussion

There are several things I learned from this project. First, Machine Learning is not only about using Machine Learning Algorithms. Pre-process the data properly is crucial. My first 3 experiments did not show much improvement on TP Rate because of the training data are mostly filled with ‘0’. After we pre-process the data in experiment 4, we can see improvements in almost every classifier.

Second, trying out different classifiers are necessary. Before doing the project, I think K-NN might be the best classifier because of the “closeness” of language habits of people in the same region, but it turns out NBM, LR, and LMT are much better.

From the results of this project, I would recommend using Naïve Bayes Multinomial, Logistic Model Tree and Logistic Regression.

For Naïve Bayes Multinomial, it is a linear classifier, fast, and highly scalable. We can put more instances in our dataset and train our model in a relatively short time. Also, unlike decision trees, it is not likely to have an over-fitting problem with massive training data because of its feature conditional independence hypothesis.

For Logistic Regression, I think one of the reasons LR did well in this project is because of good Feature Engineering. Since the attributes in bestXX are meaningful, they work well with LR’s decision boundary even though LR was mainly a binary classifier.

For Logistic Model Tree, it combined benefits of LR and the decision tree. The decision tree uses IG to select the best features for the LR model. I think that is the reason it has the highest accuracy in this project.

7 Conclusion

The model developed for are project Naïve Bayes Multinomial, Logistic Model Tree and Logistic Regression.

Naïve Bayes Multinomial is fast, scalable and unlikely to get over-fitting problems. A large number of instances can be used to train the model to improve accuracy. Logistic Regression works well if the features are properly selected. Logistic Model Tree can select the best features for its LR model, which performed best in this project.

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 preprint arXiv:1804.08049* (2018).

[3] Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. 46 (3): 175-185. doi:10.1080/00031305.1992.10475879. hdl:1813/31637.

[4] Quinlan, J. R. C4.5: Programs for *Machine Learning.* Morgan Kaufmann Publishers, 1993.

[5] Ho, Tin Kam (1995). *Random Decision Forests* (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original(PDF) on 17 April 2016. Retrieved 5 June 2016.

[6] Maron, M. E. (1961). "Automatic Indexing: An Experimental Inquiry" (PDF). *Journal of the ACM*. 8 (3): 404–417. doi:10.1145/321075.321084.

[7] Walker, SH; Duncan, DB (1967). "Estimation of the probability of an event as a function of several independent variables". *Biometrika*. 54 (1/2): 167–178. doi:10.2307/2333860. JSTOR 2333860.

[8] Ian H. Witten; Eibe Frank; Mark A. Hall (2011). "Data Mining: Practical machine learning tools and techniques, 3rd Edition". Morgan Kaufmann, San Francisco. Retrieved 2011-01-19.

[9] Powers, David M W (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" (PDF). *Journal of Machine Learning Technologies*. 2 (1): 37–63.

[10] Fawcett, Tom (2006). "An Introduction to ROC Analysis" (PDF). Pattern Recognition Letters. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010.

[11] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106

[12] Niels Landwehr, Mark Hall, and Eibe Frank (2003). Logistic model trees (PDF). ECML PKDD.