COMP90049 Knowledge Technologies Assignment 2

Can we use tweet text to help us to identify trolls on Twitter?

1. Introduction

Twitter is an American social networking site and one of the most popular websites in the world. User posts and interacts with messages on Twitter. In these short messages of less than 140 words known as "tweets", the user's personal thoughts will more or less be expressed. This article mainly discussed the possibility of inferring user's political tendency (trolls) from his/her tweets by implementing machine learning methods. There are three different classifiers used in the experiments which are zeroR, naive Bayes and decision tree classifier. By applying different classifiers, different experiments were performed and then analyzed these experimental results horizontally and vertically. The final section concludes.

2. Dataset

The dataset manipulated in experiments is constructed by 175 randomly chosen users and a total of 223K tweets [1]. This was then divided into three small files which are training (roughly 60%), development (20%), and test (20%) sets. In the dataset, "mostXX" and "bestXX" has recorded different terms as attributes by different methods, where "bestXX" files have more attributes than "mostXX"(more precise).

3. Classifier

Classification is an important method of data mining and machine learning. The data is mapped to a certain category by different classification methods, i.e. classifiers, and then predicted. The data in the training dataset is executed by different classifier algorithms, trained to obtain different models, and then the data in the validation dataset are separately executed by different models to generate prediction results. The performance of different classifiers is evaluated based on the predicted results. Three classifiers are used in this article, which are zeroR, naive Bayes and decision tree classifier, respectively.

- ZeroR

As the simplest classification method, zeroR just choices the majority class as predicted value on the basis of statistics of historical data. Although zeroR classifier is incapable of predicting, it is the most commonly used baseline in machine learning to evaluate other classifiers.

- Naive Bayes

The mathematical principle of the naive Bayes classifier is based on Bayes' theorem. Bayes' theorem is as follows,

$$\mathrm{P(A \mid B)} = \frac{\mathrm{P(B \mid A)P(A)}}{\mathrm{P(B)}}$$

The naive Bayes classifier is based on a simple assumption that given a target instance attributes are independent with each other. Through the above assumptions, we can draw the following formula:

$$\mathrm{P(category \mid instance)} = \frac{\mathrm{P(intance \mid category) * P(category)}}{\mathrm{P(instance)}}$$

$$= \frac{\mathrm{P(feature_1 \mid category) * \ldots * P(feature_n \mid category)}}{\mathrm{P(feature_1) * P(feature_2) * \ldots * P(feature_n)}}$$

where instance has n independent features. Naive Bayes classifier is insensitive to missing data and the principle of algorithm is simple.

Contrary to expectations, the independence assumption is not a good idea but it has good performance comparing to other sophisticated classifiers in practice [2].

- Decision tree

Decision tree is a model which represents the mapping from attributes to outcome. The node of decision tree represents different class and the path from root node to leaf node is the classification rule. As one of the most popular classifiers, complex dataset is unnecessary, also, it has high efficiency. Once a decision tree is constructed, it can be used repeatedly and get well-performing results in a relatively short time [3].

4. Evaluation metrics

The experimental results are evaluated by 4 major parameters throughout this paper: accuracy, precision, recall and F-score. Accuracy is the proportion of instances for which we have correctly predicted the label:

$$\text{Accuracy} = \frac{\text{correct prediction}}{\text{total count}}$$

Precision is the proportion of positive predictions that are correct:

$$\text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

Recall is the accuracy with respect to positive cases:

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

In large dataset, prediction and recall are often mutually constrained. In many cases, we need to consider these two parameters comprehensively, therefore, this leads to a new parameter F-score. F-score gives us an overall picture of system performance:

$$\text{F} - \text{score} = (1 + \beta^2)\frac{\text{PR}}{\text{R} + \beta^2\text{P}}$$

where R = recall and P = precision. Conventionally $\beta = 1$, called the F1-score:

$$\text{F1} - \text{score} = \frac{2\text{PR}}{\text{P} + \text{R}}$$

5. Experiments

5.1 Implementation

Several experiments are carried out by implementing different classifiers on different dataset.

The first part of experiments is cross-validation on the training data (train-bestXX) using zeroR, naive Bayes and decision tree classifiers and the validation sets are dev-bestXX corresponding to different train-bestXX. In this part, the “tweet-id” and “user-id” (two attributes) has been removed.

The second part of the experiments is cross-validation on the training data (train-bestXX) using naive Bayes and the validation sets are dev-bestXX corresponding to different train-bestXX. Unlike the first part, only “tweet-id” attribute has been removed.

5.2 Experimental results

	best10	best50	best200
Accuracy	35.785%	35.785%	35.785%

Tab 1. Performance of zeroR classifier on dataset bestXX (train-bestXX, dev-bestXX)

	best10	best50	best200
Accuracy	55.707 %	61.5511 %	62.6917 %
Precision	0.582	0.614	0.625
Recall	0.557	0.616	0.627
F-score	0.516	0.587	0.599

Tab 2. Performance of naive Bayes classifier on dataset bestXX (train-bestXX, dev-bestXX)

	best10	best50	best200
Accuracy	61.5457%	64.998%	67.226%
Precision	0.622	0.654	0.676
Recall	0.615	0.650	0.672
F-score	0.584	0.629	0.652

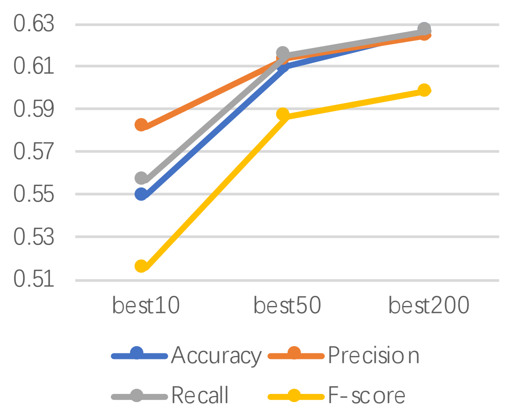
Tab 3. Performance of decision tree classifier on dataset bestXX (train-bestXX, dev-bestXX)

	best10	best50	best200
Accuracy	55.9437 %	60.1274 %	61.172 %
Precision	0.553	0.596	0.606
Recall	0.559	0.601	0.612
F-score	0.519	0.568	0.580

Tab 4. Performance of naive Bayes classifier on dataset bestXX (train-bestXX, dev-bestXX) with "user-id" attribute

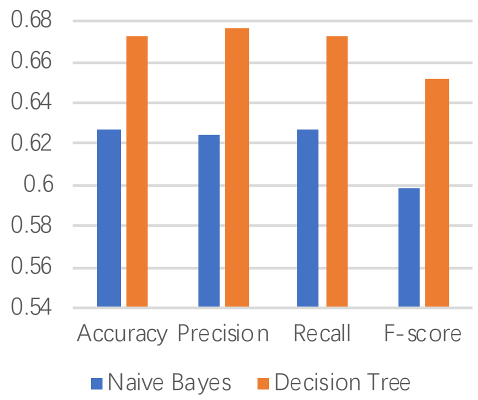
5.3 Analysis

In table 1, the accuracy of zeroR classifier on dataset bestXX is 35.785%, so the baseline is set with 35.785% accuracy. In the following experiments, no matter the "user-id" attribute was removed or not, all of the accuracy results are much higher than 35.785%, therefore, we can say naive Bayes and decision tree classifier is eligible.



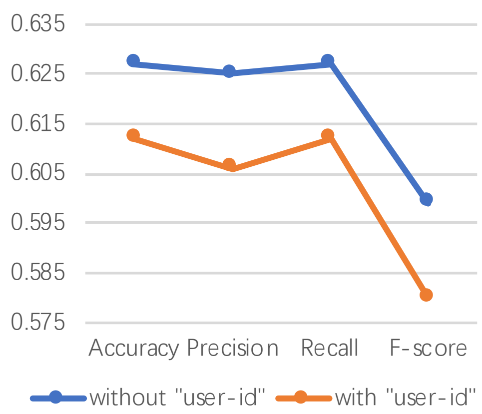
Pic 1. Performance of naive Bayes classifier on dataset best10, best50, best200

In picture 1, the classifier is the same, which is naive Bayes, so the only variable is the amounts of attributes. As we can see clearly in the picture, the performance is improved greatly by training data on the dataset with more attributes.



Pic 2. Performance of naive Bayes and decision tree classifier on dataset best200

In picture 2, the performance of decision tree classifier is much better than naive Bayes classifier. In this case, another parameter needs to be considered, which is running time. For naive Bayes classifier, the running time on training dataset best200 is 20 seconds but decision tree classifier takes 1450 seconds! Therefore, it's actually a tradeoff between performance and time.



Pic 3. Performance of naive Bayes classifier on dataset best200 with/without "user-id" attribute

In picture 3, the performance of naive Bayes classifier without "user-id" attribute is slightly better than performance with "user-id" attribute. This may be because not all tweets are related to personal political tendency. Even though a user is a staunch left troll, he/she may also post a normal tweet that has nothing to do with his/her political tendency, such as "Had an amazing time at XXX. Really glad I could hang out with some of my favorite people."

The above experimental results show that both naive Bayes and decision tree classifiers have very good performance (accuracy is much higher than baseline). If we remove the "user-id" attribute, the results show that the probability of correct speculation is around 60-70% which means we can predict the content of a tweet is belongs to left troll, right troll or others by implementing machine learning methods and the success rate is around 60-70%. If we add the "user-id" attribute, the probability will be slightly decreased. In this case, we can predict the user's political tendency by his/her tweets and the success rate is around 60%. As far as the current experimental results are, 60% accuracy/precision/recall is quite satisfactory. Admittedly, there are still many things need to be improved further, such as classification algorithm, attributes chosen.

6. Conclusion

In this study introduced three machine learning methods and implemented them on different datasets. The results are compared horizontally and vertically by several experiments. All in all, the possibility of inferring user's political tendency from his/her tweets using machine learning methods is high and it is still set to improve further in many respects.

(1294 words)

References

[1] Linvill, Darren, P. Warren, "Troll factories: The Internet Research Agency and state-sponsored agenda building (working paper)", Clemson University, 2018.

[2] I. Rish, "An empirical study of the naive Bayes classifier", *T.J. Watson Research Center*, 2001.

[3] L. Rokach, O. Maimon, "Top-down induction of decision trees classifiers – a survey", *IEEE Transactions On Systems, Man, And Cybernetics-Part C: Applications And Reviews*, Vol 35, No. 4, 2005