COMP90049 Knowledge Technologies, Semester 1 2019

Project 2: tweets r mad, or r they!?

1. Introduction

Nowadays, people often use social media to share daily lives and their feelings. Based on that, some companies have a desire to collect comments and feedback from different platforms and gain some knowledge for improvement. Twitter is one of the social networking services that many companies have an account to know customers' feedback.

In this report, sentiment analysis is implemented to classify specific English short tweets by using some supervised Machine Learning methods. The main purpose of the report is to extract some knowledge from the tweets text and determine whether tweet texts can help us to identify people sentiment (in specific positive, negative, neutral).

2. Data Set

The dataset [1] of this project is a collection of about 33K tweets, which was split into two parts: a training set, an evaluation set, and a test set. For each part there are a .txt file and a .arff file.

The .text file contains the raw text of the tweets, one tweet per line, in a specific format: tweet-id TAB tweet-text NEWLINE. It was pre-processed (folding case and removing all characters that are not alphabetic ([a-z])) before the feature engineering/selection was performed.

The .arff file contain the structured information that transformed from raw text. And these files also contain each tweet's sentiment of either positive, negative or neutral which was manually assigned.

This report uses the train.arff to build models, verifies it by using eval.arff and then predicts the sentiments in the test.arff. During the process, some attributes in the training set and evaluation set are changed and new arff files are produced.

3. Weka

In this project, Weka is used for implementing the supervised Machine Learning methods on the dataset. It is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand [2]. It contains a collection of visualization tools and algorithms for data analysis and predictive modeling, also with graphical user interfaces.

4. Methodology & Results

In this report, we try to predict the tweets sentiments by two data mining methods: Naïve Bayes and Decision Tree. And try to improve the models based on the results.

4.1. Classifiers

4.1.1. Naïve Bayes

The Naïve Bayes classifier is a supervised Machine Learning method based on applying Bayes' theorem with strong independence assumptions between the features. It searches the highest probability value from all categories of documents tested [3].

4.1.2. Decision Tree

Decision Tree is a typical way to output classification methods in a logical model. In this report the J48 decision tree is used, it is the Weka implementation of the standard C4.5 algorithm [4].

4.2. Original Attributes Analysis

In the original data set, there are total 47 attributes including one id and one sentiment class. The feature selection was applied to generate the frequencies of 45 tokens in the training data and then summarised as a row for each tweet-id.

To get a simple baseline performance which can be compared with other classifiers, the ZeroR classifier was used. It predicts the class value that has the most observations in the training dataset for a classification predictive modelling problem [5].

After that, the Naïve Bayes and J48 were implemented on the train.arff with percentage split (66) and supplied test set eval.arff to get the accuracy of predictions (see Table 1).

	Accuracy
ZeroR on train.arff	49.83%
ZeroR on eval.arff	48.71%
Naïve Bayes on train.arff with percentage split	53.03%
Naïve Bayes on eval.arff	46.50%
J48 on train.arff with percentage split	54.23%
J48 on eval.arff	47.41%

Table 1: The Accuracies of Different Methods

From the table, it is obvious that although the ZeroR has lower accuracy on the training data set compare to Naïve Bayes and J48, however it gets the highest accuracy on the evaluation data set. This shows the original attributes which were selected by frequencies might not be good enough to define the classification. This shows that some attributes might have a little dependency on the sentiment expressions. So, some change is necessary for a better prediction.

4.3. Refine Attributes & Analysis

To get a better set of attributes, the frequencies of each attributes calculated by the Naïve Bayes classifier were checked. We found that some attributes are almost evenly distributed in sentiment class, for example, 'and' gets 0.3521 in negative, 0.3983 in positive and 0.3249 in neutral. Attributes like this are removed for better prediction results. So, 'and', 'do', 'is', 'like', 'smart', 'so', 'the', 'to' and 'we' are deleted for this reason. Although 'id' has the similar distribution like the attributes mentioned before, it is kept to avoiding the deletion of some data without any attributes.

Then the decision tree built by J48 was also checked. We found it is hard to gather knowledge from the massive decision tree. So, we tried to delete the attributes mentioned above first and then use three classifiers again to do the prediction. The results are shown in Table 2.

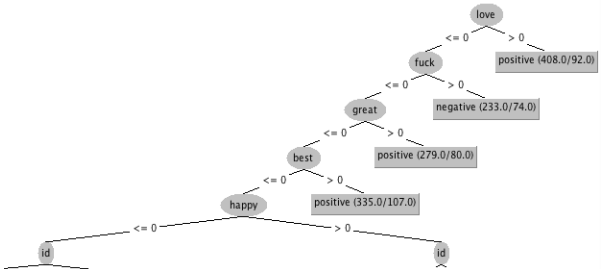
	Accuracy
ZeroR on train.arff	49.83%
ZeroR on eval.arff	48.71%
Naïve Bayes on train.arff with percentage split	53.47%
Naïve Bayes on eval.arff	47.47%
J48 on train.arff with percentage split	54.96%
J48 on eval.arff	48.35%

Table 2: The accuracies after deleting attributes

There are slightly improvements of the accuracy of Naïve Bayes and J48. However, more improvement still needs be done with the attributes. So, the decision tree built by J48 was checked. The root node and some top-level nodes turn out to be 'liberals', 'not', 'my' etc which some are rarely relevant with sentiments literally. So, we tried to delete these attributes: 'big', 'country', 'crap', 'ice', 'job', 'liberals', 'my', 'night', 'not', 'obama', 'people', 'their', 'they', 'today', 'trump', 'vote', watch' and use three classifiers to do the prediction again. The results are shown in Table 3. And the part of the decision tree built by J48 is show in Pic 1.

	Accuracy
ZeroR on train.arff	49.83%
ZeroR on eval.arff	48.71%
Naïve Bayes on train.arff with percentage split	52.87%
Naïve Bayes on eval.arff	49.32%
J48 on train.arff with percentage split	53.89%
J48 on eval.arff	48.26%

Table 3: The accuracies after deleting attributes again



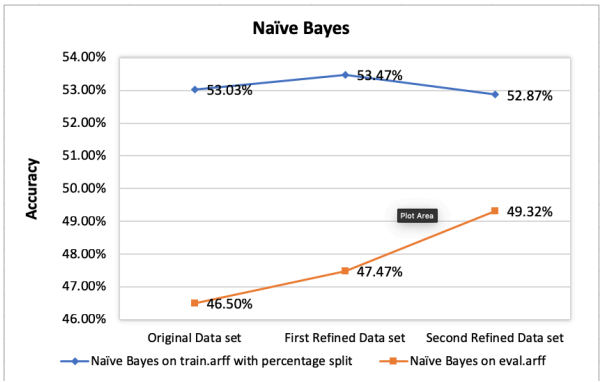
Pic 1: Part of The Refined Decision Tree

The rules are much clear and understandable compare to the tree built before. For the decision trees algorithm, the higher nodes are better attributes for prediction. From the prcture, we can see the root node 'love' is the word with high potential to express positive emotions. And pruned decision tree is also tried during the process, however the result is not ideal.

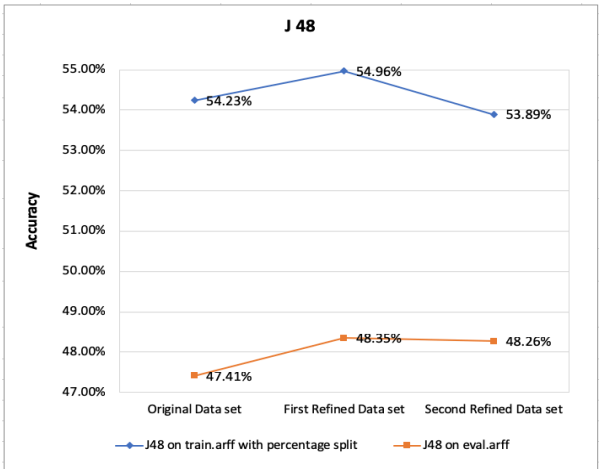
5. Discussion

When analysing the results of Naïve Bayes on the dataset (see Pic 2), it is obvious that the accuracy of prediction grows during the process from being lower to higher than the baseline. For Decision Tree (see Pic 3), the first process helps to achieve higher accuracy, but the latter process makes the accuracy slightly decreased. The reason

of change is complex, when delete some attributes which someone supposed to be irrelevant, the accuracy might increase for reduce obstruction, but it also might decrease for losing some information.



Pic 2: Results of Naïve Bayes



Pic 3: Results of J48

During the process, there are some interesting findings. First, there are many words in tweets with high frequency but have no sentiments at all such as 'to', 'and', and 'the' etc. Second, some words somehow seem literally related to sentiments but have little contribution to the model, such as 'like'. It is because the words might be used in different ways, such as 'like' can be a preposition, conjunction and adjective etc. Third, some words literally have no relation with emotions but do have bias on sentiment class. For example, the word 'liberals' has a severe bias toward negative sentiment.

To achieve better prediction results, it is necessary to define good attributes as well as suitable models. Many factors should be considered such as latest news, time-based hot topics and celebrities etc. Some words might be quite popular during a special period but came to be low frequency word later. During collecting the training data set, it might be good to take the factors into account depends on different situations. Also, some phrases and sentences can be really helpful in sentiment analysis. And the prediction accuracy can be high enough to help people identify the sentiment by some suitable methods by weighted attributes or different algorithms.

5. Conclusions

For this dataset, the Naïve Bayes achieves better results finally, but it does not mean Naïve Bayes are better method than Decision Tree. And much more methods might achieve better results for this dataset. But we choose the Naïve Bayes as the final model for predict the refined test.arff file.

In conclusion, we believe the tweet text can help us to identify people sentiment on Twitter, however, there are many factors need to be considered for better prediction during the process.

6. References

[1] Rosenthal, Sara, Noura Farra, and Preslav akov (2017). SemEval-2017 Task 4: Sentiment Analysis in Twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval '17). Vancouver, Canada.

[2] Weka (machine learning). (2019). Retrieved from https://en.wikipedia.org/wiki/Weka_(machine_learning)#cite_note-1

[3] Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition. (2011). Database and Network Journal, (2), 9. Retrieved from https://search-ebscohost-com.ezp.lib.unimelb.edu.au/login.aspx?direct=true&db=edsgao&AN=edsgcl.254556764&site=eds-live&scope=site

[4] The Research and Analysis in Decision Tree Algorithm Based on C4.5 Algorithm. (2018). 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018 IEEE 3rd, 1882. https://doi-org.ezp.lib.unimelb.edu.au/10.1109/IAEAC.2018.8577527

[5] Brownlee, J. (2019). How To Estimate A Baseline Performance For Your Machine Learning Models in Weka. Retrieved from https://machinelearningmastery.com/estimate-baseline-performance-machine-learning-models-weka/