**COMP90024 Knowledge Technologies**

**Project 2: R ur tweeps as mad as u think ? #analysis**

*Table.1 Estimate Naïve Bayes and Decision Tree (J48)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P\_NB | R\_NB | P\_J48 | R\_J48 |
| Pos | 54.80% | 49.50% | 58.90% | 37.80% |
| Neg | 44.60% | 30.00% | 46.70% | 25.50% |
| Neu | 56.60% | 68.00% | 56.10% | 79.60% |
| Avg | 53.50% | 54.40% | 55.00% | 56.60% |

|  |  |  |
| --- | --- | --- |
| Accuracy | 54.4052% | 55.5826% |
| Runtime | 2.13s | 10+s |

1. **INTRODUCTION**

There are many researches about analysis tweets sentiment. For example, one direction is considered as finding new methods for effectively analysis, such as employ social relations for user level sentiment analysis [1] and another direction is concentrated on identifying and adding new features to the trained model for sentiment analysis, such as, the presence of intensifiers and character repetitions [2].

This report will discusses the observations, approaches and analysis by using machine learning method to classify different sentiment class (positive, negative and neutral) for each tweet instance. The primary objective of this report is using evaluation metrics to compare different classifiers performance on different features set. Naïve Bayes and Decision Tree (J48 in weka[[1]](#footnote-1)) classifiers are utilized in the machine learning process. Bigrams and Emoticons strategies are used to generate new features in feature engineering part. Data are provided by the 2017 SemEval conference [3] and seperated into three parts---training, development and testing. High accuracy for classifying test data set is the most anticipated desideratum for this project.

1. **FEATURE ENGINEERING**

**2.1.ORIGINAL FEATURES ANALYSIS**

The original data set has 46 features, which are generated by applying the methods of mutual information[[2]](#footnote-2) and Pearson’s X-Squared test[[3]](#footnote-3). Table.1 shows the accuracy, precision and recall from Naïve Bayes and J48 classifier on the original 46 features and default setting in weka.

From Table1, some useful information can be concluded as below:

The accuracy for J48 is higher than Naïve Bayes for this data set. One possible reason for this situation is because the assumption for Naïve Bayes is that attributes are conditionally independent, but this is too ideal for sentiment analysis among tweets, it will cause a bias and let the accuracy down.

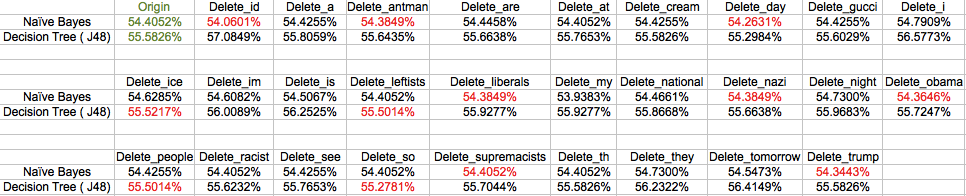
1. The running time for Naïve Bayes is much faster than J48, because J48 needs a long time to build a tree with 46 features. Also, the calculation of Naïve Bayes is simpler than J48.
2. Both Naïve Bayes and J48 classifier have higher recall for neutral class than others, which implies these two classifiers can’t classify well for positive and negative class with the training data set. There are many reasons for leading this phenomenon, first one can be that the original 46 features are recorded as the frequency of tokens within the tweets, to some extent, some of them can hardly represent sentiment and harder for analysis sentiment. For example, features like “a”, “are” … etc. they do not have strong correlation with sentiment classes. The second reason is possible because the number of feature is not enough, needing a sentiment tokens library for better classify positive and negative class. The third reason is sometimes the sentiment from tweets is not represented as a real “word“, but in another represent style. Such as, the presence of intensifiers and character repetitions.

The positive and negative recall to Naïve Bayes is higher than J48, but neutral is opposite. The reason can be assumed that Naive Bayes would perform better when the training data do not contain all possibilities so it can be very good with low amounts of data. But J48 works better with lots of data compared to Naive Bayes.

2.2. Improved Original Features

2.2.1 Delete Unrelated Features

As mentioned in 2.1.iii, many features don’t have a strong relationship with sentiment. Select some maybe unrelated features, delete them and try to find if they have an impact on the two classifiers. The test accuracy result as Figure.1, all the results are just delete one feature.



*Figure.1 Test to find unrelated features for each classifier*

Note: In Figure.1, the green value means reference value. The red value means this feature will affect that classifier. The black value means unrelated feature to that classifier.

So, the original features can be respectively deleted to fit naïve bayes and J48 classifier. Accuracy after deleting is as Figure.2.

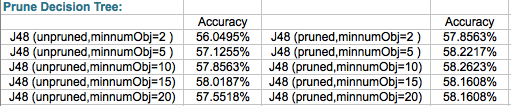
**

*Figure.2 Accuracy after delete unrelated features*

Compare to original 46 features, 28 features for naïve bayes and 23 features for J48 can gain higher accuracy.

2.2.2 Pruned Decision Tree

The statistic for J48 decision tree shown before is unpruned. But decision tree should be pruned to get better performance, because pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify in instance and pruning can reduce the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.



*Figure.3 Compare Pruned and Unpruned J48 Decision tree*

The parameter “minnumObj” in weka means the minimum number of instances per leaf (Default is 2). It should be increased for eliminating noisy data effect.

From Figure.3, some useful information have been shown as below:

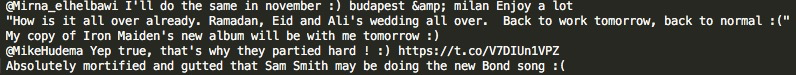
1. Obviously, J48 decision tree after pruning can always gain higher accuracy (comparing the left column with the right column in same row).
2. With the value of minnumObj increasing, the accuracy is firstly ascending then descending. Ascending means the reduction of overfitting, because it eliminates some noisy data influence in a certain extent. But later descending means the threshold value for stopping splitting the node is too large and leads to underfitting problem.

2.3. Applied New Features Analysis

2.3.1 Add New Features for Identifying Emoticons in Tweets

After observation, some tweets don’t contain a real word for expressing their sentiment but using emoticon. Emoticon is a pictorial representation of a facial expression using punctuation marks, numbers and letters, usually written to express a person’s feelings or mood[[4]](#footnote-4).

Manually select some of the example in the using data set:

**

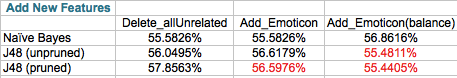
*Figure.4 Example of emoticon in tweets*

The sad and happy emoticons used in the report is as below Figure.5.

**

*Figure.5 Sets for sad and happy emoticons*

After adding new emoticon feature, the classify result is described in Firgue.6.

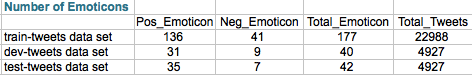
**

*Figure.6 Accuracy after adding emoticon feature*

The difference between third column and fourth column is that the fourth column use SMOTE Filter in weka for oversampling the instance so as to balance the instance in the majorclass and minorclass.

From Figure.5, some useful information have been shown as below:

1. To Naïve Bayes classifier, the accuracy is only increased after balancing. The reason for this situation maybe because the number of tweets that contains emoticon is too little comparing to the total tweets as shown in Figure.7, meaning the training data for emoticon feature are not enough, the testing data for emoticon are also not abundant and it’s only 0.8% of data in testing data set have emoticon. So it’s reasonable that there is not a big increase for accuracy.

**

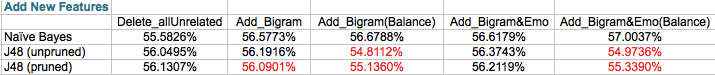
*Figure.7 Statistic for Emoticons in different data set*

1. To J48 decision tree classifier, it’s easy to find that the majority of the result is not good. To some extent, it indicates the same conclusion as in 2.1.iv, which is that decision trees work worse with low amounts of data compared to naïve bayes.

2.3.2 Add New Features for Identifying Bigram tokens in Tweets

Bigram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllables, or words. A bigram is an n-gram for n=2.[[5]](#footnote-5)

Bigram feature is implemented in the report. It uses the train-tweet.txt data set to obtain positive, negative and neutral bigram sets (100 highest frequency bigrams for each type of bigram sets). So, it can label each tweet with the sentiment bigram sets to some extent. After bigram feature engineer, the accuracy is shown as Figure.8.

**

*Figure.8 Accuracy after adding Bigram feature & adding Emoticon and Bigram two feature*

From Figure.3, some useful information have been shown as below:

1. It’s clearly to see that after adding “Bigram” feature, the accuracy has a slightly increase to both Naïve Bayes and J48 classifier (column2-3). And it can obtain higher accuracy after adding both “Emoticon” and “Bigram” feature (column2,3,4).
2. After observation for the given data set, the data distribution is very unbalance. Trying “SMOTE” filter (using default parameter setting) in weka to oversampling the data, which can use some near instances to build more instances for the minorclass. The statistic shows that Naïve Bayes is more sensitive to unbalance data when comparing to J48, because after balancing the data, the accuracy has increased.
3. After pruning J48 tree, the accuracy seems to decreasing. It can be assumed that the tree is not overfitting, so if prune the tree, it will lead to underfitting.
4. Conclusion

This report mainly uses Naïve Bayes and J48 decision tree classifier’s accuracy to analysis the given dataset from some features. Firstly modify the given 46 features (28 for Native Bayes and 23 for j48) and then add 2 more features, Emoticon and Bigram for comparing. Finally Naïve Bayes classifier gets about 57% and 56.4% accuracy for the given development data set. There are many reasons for that accuracy is still not very high, one is that not many tweets contain emoticons and just choose 100 highest frequent bigrams for each type of sentiment to create sentiment bigram set, it’s not enough. Besides, test the data set with vader sentiment analysis library[[6]](#footnote-6), which specially aims to analyze sentiment for tweet. Accuracy given by vader is just about 80%, which is not high enough as well. Also, vader consider many other features that the report didn’t implement so that makes a different accuracy between the report’s result and vader.k

1. Reference

[1] Guerra, P., Veloso, A., Meira Jr., W., Almeida, V.: From bias to opinion: A transfer-learning approach to real-time sentiment analysis. In: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD (2011)

[2] Gimpel, K., Schneider, N., O’Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.: Part-of-speech tagging for twitter: Annotation, features, and experiments. Tech. rep., DTIC Document (2010)

[3] Rosenthal, Sara, Noura Farra, and Preslav Nakov SemEval-2017 Task 4: Senti- ment Analysis in Twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval ’17). Vancouver, Canada. (2017)

Your analysis will be in terms of interpretting this data. For example, when we say that Naive Bayes has an Accuracy of 54.4%, what does that mean? What does Naive Bayes look like, and is that a sensible approach for this data? Is 54.4% a good number? If it isn't, what is making the problem hard for the method? And so on.

conclusion..

There are two conclusions reached: better features should be generated for higher performance in identifying geolocation of tweets; and Random Forest method performs the best with engineered features and an intermediate computation cost in classification.

1. Weka: <http://www.cs.waikato.ac.nz/ml/weka/> [↑](#footnote-ref-1)
2. http://en.wikipedia.org/wiki/Mutual\_information [↑](#footnote-ref-2)
3. http://en.wikipedia.org/wiki/Person\_chi-sequared\_test [↑](#footnote-ref-3)
4. Emoticon: https://en.wikipedia.org/wiki/Emoticon [↑](#footnote-ref-4)
5. Bigram: https://en.wikipedia.org/wiki/Bigram [↑](#footnote-ref-5)
6. vader sentiment analysis: https://github.com/cjhutto/vaderSentiment [↑](#footnote-ref-6)