* Description
* This report is written for the second project of the subject knowledge technology.
* The purpose of this project is to analysis the sentiment of the tweets using different data mining approaches and acquire knowledge from the results. After the feature engineering, four machine learning approaches are applied in this project including Naïve Bayes, Decision Tree, Random Forest, and KNN. Then, a comparison of the efficiency for each method will be present with the limitation of the project. Finally, the acquired knowledge from analysis will be shown.
* Background
  + Data set
    - The tweet collections used in this project is a sub-sample gathered from Twitter and own by a research group (Rosenthal, 2017). There are around 23000 data points in the train.txt which is the training data for the models, and around 5000 data points in the dev.txt which are used to evaluate the models.
  + Weka
    - Weka is used in the project, which is a commonly used tool providing a number of machine learning algorithms.
  + Classifiers
    - In this project, four classifiers are chosen to compare with each other. The four classifiers are Naïve Bayes, Decision Tree, Random Forest, and KNN. The classifiers are chosen from the ones introduced in the subject due to the interpretability.
* Feature engineering
  + Twitter features
  + To construct more effective features for the tweets, it is critical to notice that a single tweet is not just a plain text message, but also contains features other than plain words which may contribute to its sentiment.
  + Negation words
    - The present of the negation word will change the opinion of the tweet entirely. For example, not good means bad. But the negation words could various representations, such as not, didn’t, doesn’t, which means that for each of the negation words, the frequency could be relatively low, and to capture all of them may increase the dimension of the data significantly.
    - Another property of the negations words is that the existence of double negations. For example, “I couldn’t not help her” actually means “I felt I should help her”.
    - To solve the problems mentioned above, a feature named “NEGATIONWORD” is constructed. First, the system was given a collection of negation words. Then, instead of using the number of appearance of the negation words in the tweet, the modulo operation was applied to handle the double negation. The attribute NEGATIONWORD is calculated out of the following formula.
    - NEGATIONWORD(t) = c mode 2
    - c indicating the number of negative words in the tweet.
    - Emoticons
    - Emoji and other symbol expressions such as “:)”, “:(”, are widely used now to represent emotions. A good news for sentiment analysis is that some of tend to indicate obvious positive or negative emotions and the negative words will not be applied on the emoji and facial expressions, which means that these emoticons usually express strong and clear opinions than normal words.
    - In this project, two features are constructed to have a better use of the emoticons. They are called “POSIEMOJI” and “NEGEMOJI” and contains the most commonly used positive emoticons and negative emoticons respectively. And the value of each attribute is the occurrence of the corresponding emoticons in the tweet.
  + Preprocessing

The tweets used in the project has been preprocessed to remove less informative contents such as author, time stamp, etc. The preprocessing is still needed for the project.

In the preprocessing stage, the URLs, tags, and mentions is the tweets will be removed. And most of the stop words such as ‘the’, ‘a’ will also be removed due to limited information they provide. Another critical step for preprocessing is stemming. Stemming is an approach to assemble the strongly related tokens to the same type of token. For example, the present tense and the past tense of the same word usually represents the similar opinion.

* Feature selection

To select features that are more related to the sentiment of the tweets, a score imitating the degree of purity of the classification have been given to each of the features. The purity score of feature f towards class c can be calculated using the following formula.

Score(f,c) = p(c|t)

The system ranks all the features based on this score for three classes. And select the top 200 features from the result of class positive and class negative.

A comparison between the provided attribute set and the selected attribute set on the performance of the decision tree model is shown in the figure #. It is easy to see that both the accuracy and the F-Measurement increased hugely in the selected attribute set. One of the reason is that in the provided attribute set, there are features represents the words like “at”, “are”, which may have high occurrence but barely contribute to the sentiment of the tweet. Another possible reason is that, the score function used in this project is basically assessing the purity of the classes, which would be beneficial for the decision tree to have a better performance.

Figure #: the comparison on

* Results and effectiveness analysis
  + The results

As mentioned above, the project chose four machine learning algorithms, including Naïve Bayes, Decision Tree, Random Forest, and KNN. All of them using the same dataset training.txt to build and train the model, then they evaluate the performances on the same dev.txt data. Specifically, for the KNN, the project use three nearest neighbors.

The main statistic results of the data are shown as table #.

Table #: the results of the classifiers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Naïve Bayes | Decision Tree | Random Forest | KNN |
| Testing time(s) | 2.02 | 0.24 | 3.15 | 217.72 |
| Kappa | 0.209 | 0.336 | 0.336 | 0.336 |
| Avg. Precision | 0.545 | 0.647 | 0.651 | 0.639 |
| Avg. Recall | 0.553 | 0.621 | 0.626 | 0.625 |
| Avg. Accuracy | 0.5534 | 0.6208 | 0.6265 | 0.6251 |
| Avg. F-Measure | 0.504 | 0.593 | 0.604 | 0.608 |
| Avg. ROC area | 0.679 | 0.679 | 0.73 | 0.73 |

The effectiveness measurements that will be used to evaluate the classifiers in the paper are Avg. ROC Area and Avg. F-Measure. The main reason is the imbalance of the data set. Accuracy could be impacted significantly by the class distribution. For example, a classifier who make wrong decisions to all instances of class A, and if class A is in the minority of the test set, the classifier could still get a high accuracy. On the contrary, the Avg. F-Measure and Avg. ROC Area can handle the unbalanced classes much better. (Ref)

A comparison on these measurements along with Accuracy between the classes are show as Figure #.

Figure #

As can be seen from the Figure #, Naïve Bayes is the lowest in all the three measurements. Random Forest perform slightly better in F-Measure and Accuracy and acquired notably higher ROC area compare to Decision Tree. Random Forest and KNN behave similarly with hardly noticeable differences with KNN.

Figure #:

Comparison

Naïve Bayes is based on the assumption that the attributes are independent from each other, which is not true in this case. Another serious problem raised from this assumption is the zero-frequency problem. If a certain feature value has not occurred within a certain classification, which is quite common in sentiment analysis, the related conditional probability would be zero which will influence the performance of the classifier hugely.

Decision tree performs better than Naïve Bayes mainly because it is able to easily handle the relationships between the features. And also redundant feature will not affect the accuracy of the classifier because the best one will be chosen to make decisions. The main problems behind the Decision tree is over-fitting.

Due to the minor difference between the Decision Tree, Random tree and KNN, a detailed comparison between those classifiers have been show in the Figure #. It compares on not only the Avg. F-Measure, but also on the F-Measure of each classes in the dataset.

Random Forest select the subsets of the features and the instances using bagging to form multiple decision trees to overcome the over-fitting issue. It is obviously to see that in the Figure #, the Negative F-Measure increased although it is not shown in the Avg. F-Measure. Furthermore, it is critical to notice that the Positive F-Measure of Random Forest decrease notably compare to the Decision Tree. Because the stability of the decision tree decides the performance of the Random Forest (), it is fair to say that decision tree is stable when identifying instances in class Positive. That is the reason why the performance of classifying class Positive dropped after bagging.

KNN perform slightly better than Random Forest on most of the measurements. As an instance-based classifier, KNN is able to generate more complicated decision boundaries than the rule-based classifiers. () As a lazy leaner, KNN will build the model only when the classification is required, which is not satisfied for the real time applications. In contrast, eager learners react rapidly once the training process is done.

* Limitation
* Conclusion

To sum up, Naïve Bayes were found to behave worse than others after the comparison on their performances. Random Forest and KNN are generally more effective than Decision Tree and Naïve Bayes, while the difference is that Random Forest could be used in real time application due to rapid classification after training. Also, a feature selection process based on the purity of classification has been presented in the paper to derive a more sensible attribute set from the data.

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

Facing Imbalanced Data Recommendations for the Use of Performance Metrics L´aszl ´o A. Jeni1 , Jeffrey F. Cohn1, 2, and Fernando De La Torre1 1Carnegie Mellon University, Pittsburgh, PA, laszlo.jeni@ieee.org,ftorre@cs.cmu.edu 2University of Pittsburgh, Pittsburgh, PA, jeffcohn@cs.cmu.edu

<http://ca.cs.cmu.edu/sites/default/files/4Jeni_Metrics.pdf>

ROC Graphs: Notes and Practical Considerations for Data Mining Researchers Tom Fawcett Intelligent Enterprise Technologies Laboratory HP Laboratories Palo Alto HPL-2003-4 January 7th , 2003\*

<http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf>

An introduction to ROC analysis Tom Fawcett Institute for the Study of Learning and Expertise, 2164 Staunton Court, Palo Alto, CA 94306, USA Available online 19 December 2005

<https://ccrma.stanford.edu/workshops/mir2009/references/ROCintro.pdf>