COMP90049 Project2 Report

1. Introduction

Sentiment analysis in text becoming increasingly important for political and social sciences with social media platform such as Twitter (Silva et al. 2014), which is a process to determine the writer's attitude (positive, negative or neural) towards a topic or a product by using machine learning methods.

The purpose of this report is to apply four different supervised machine learning methods to classify, evaluate the performance and make predictions for given datasets.

Supervised machine learning methods in this report include Naive Bayes (NB), Trees classifier (decision tree (J48) and Random Forest (RF)) and Support Vector Machine (SVM); and the datasets is a sub-sample of actual data posted to Twitter from the 11th International Workshop on Semantic Evaluation (Rosenthal et al. 2017).

2. Related Work

Several studies for tweet sentiment analysis have been presented over two decades. Jonathon Read used emoticons as lexicon and Naïve Bayes as classifier to reduce dependency for sentiment classification (Read, 2005); Davidov uses twitter hashtags and smileys as sentiment labels and K-Nearest Neighbors (KNN) as classifier for sentiment learning (Davidov et al. 2010); Agarwal uses three types of models (unigram model, a feature based model and a tree kernel based model) and Support Vector Machine as classifier for sentiment analysis of twitter data (Agarwal et al. 2011); also, the combination of multiple classifiers for sentiment analysis is an active study recently. Kuncheva introduced basic knowledge of this topic of combining pattern classifiers (Kuncheva, 2004).

3. Methods

The basic idea of this experiment has 2 steps:

(1) Preprocess the given datasets including deleting attributes and adding new attributes.
(2) Use Weka to train and evaluate different modules and predict the sentiment of given dataset.

In the next section, we will briefly introduce the algorithms and evaluation metrics we will use in this paper.

3.1 Naïve Bayes (NB)

Naïve Bayes algorithm is based on Bayes' Theorem with an assumption of independence among features, which is easy to build and commonly used machine learning classifier. We use default value of NB in Weka.

3.2 Decision Tree (J48)

Decision tree algorithm is a supervised learning model that uses tree-like graph to predict the result by learning decision rules from features. In this paper, we used C4.5 algorithm, which is used to generate a decision tree developed by Ross Quinlan (Quinlan, 1993); and J48 is an open source Java implementation of the C4.5 algorithm in Weka.

3.3 Random Forest (RF)

Random Forest algorithm is a supervised learning model, which operates by building multiple decision trees to get a more accurate and stable prediction. We use default value of RF in Weka.

3.4 Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning model, which separates hyperplane. In other words, it builds a model that assigns new examples to an optimal hyperplane. We use default value of SVM in Weka.

3.5 Evaluation Metrics

We will use four evaluation metrics, which are accuracy, precision, recall and F-measure.

(1) Accuracy:
Accuracy will be used to calculate the proportion of correctly classified instances.

(2) Precision:
Precision will be used to calculate the proportion of correct predictions among that certain class.

(3) Recall:
Recall will be used to calculate the ratio of correct predictions to the all predictions in actual class.

(4) F-Measure:
F-Measure is to balance the precision and recall, which is a harmonic mean of precision and recall.

4 Results and Discussion

In this section, we will show the results of evaluation and analysis.

4.1 Original Feature Analysis

The original dataset after deleting “id” attribute has 45 features. The evaluation of four algorithms is shown as follows. Table 1 is the accuracy and running time to build model of four algorithms; Table 2 shows the result of precision, recall and F-Measure; and Table 3 shows the result of accuracy for training and evaluating, which implies the phenomenon of overfitting.

Algorithms	Accuracy	Runtime
NB_45	53.45%	0.06s
J48_45	54.84%	5.04s
RF_45	53.27%	25.34s
SVM_45	54.69%	37.52s

Table 1: Accuracy and runtime of Algorithms for 45 Features

NB_45	**Precision**	**Recall**	**F-Measure**
negative	43.30%	25.30%	31.90%
positive	52.00%	38.80%	44.40%
neutral	55.90%	74.20%	63.80%
Avg	52.00%	53.40%	51.30%
J48_45	**Precision**	**Recall**	**F-Measure**
negative	50.80%	18.20%	26.80%
positive	57.60%	29.90%	39.30%
neutral	54.70%	85.10%	66.60%
Avg	54.70%	54.80%	50.20%
RamFor_45	**Precision**	**Recall**	**F-Measure**
negative	43.10%	18.90%	26.30%
positive	53.40%	30.30%	38.70%
neutral	54.60%	81.40%	65.30%
Avg	51.70%	53.30%	49.20%
SVM_45	**Precision**	**Recall**	**F-Measure**
negative	58.90%	11.30%	18.90%
positive	68.10%	19.70%	30.60%
neutral	53.20%	93.60%	67.90%
Avg	58.70%	54.70%	46.60%

Table 2: Evaluation of Algorithms for 45 Features

Algorithm	Accuracy for training	Accuracy for Evaluation	Difference
NB_45	54.26%	53.45%	-0.81%
J48_45	58.47%	54.84%	**-3.63%**
RF_45	65.55%	53.27%	**-12.28%**
SVM_45	54.76%	54.69%	-0.07%

Table 3: Accuracy for Training and Evaluation for 45 Features

From Table 1 and Table 2, we can get some useful information.

Comparing Tree methods, we can easily find that J48 has a better performance than Random Forest. The reason may be too many useless feature, which influence the performance of generating decision trees in Random Forest algorithm， because Random Forest algorithm randomly selects observations and features to build several decision trees and then averages the results. Therefore, we made an assumption that feature selection will greatly affect the performance of decision tree and Random Forest.

The performance of Naïve Bayes algorithm is not very good comparing with other algorithms, and the reason may be because that the assumption for Naïve Bayes is that the features are conditional independent, but it is impossible for the real world, which may lead to some bias.

SVM works well comparing to other algorithms, which is because that SVM can efficiently perform a non-linear classification by finding the hyperplane that differentiate the classes in n-dimensional space, which is suitable for this task. However, the learning is very expensive, because it takes 37.52 seconds to build the model.

An interesting phenomenon can be found, which is that the recall for neutral class is much higher than negative and positive class for all algorithms. In other words, all algorithms can’t classify positive and negative class very well. The reason is that the original 45 features are not a good feature collection for sentiment task, because some features like “and”, “big”, “is” and “to” don’t have strong connection for sentiment analysis, which will cause some bias.

From table 3, we can get the information that overfitting is common in supervised learning algorithms, especially for tree methods. Surprisingly, Random Forest has a big difference between accuracy for training and evaluation, which is against the common sense of Random Forest can avoid overfitting. The reason may be because that the bad feature selection greatly influences the performance of Random Forest.

We made some assumptions based on the reason of bad feature collection; therefore, we will discuss the performance after choosing a better feature collection.

4.2 New Feature Analysis

We used AttributeSelection filter in Weka to select features. Comparing 45 features as original, 17 features are left. Table 4 is the accuracy, running time and the difference between the accuracy of 45 features and 17 features; Table 5 shows the result of precision, recall and F-Measure for new features; and Table 6 shows the result of accuracy for training and evaluating for new features.

Algorithms	Accuracy	Difference	Runtime
NB_17	54.05%	+0.60%	0.02s
J48_17	54.99%	+0.16%	0.59s
RF_17	54.93%	+1.66%	7.40s
SVM_17	54.60%	-0.09%	28.11s

Table 4: Accuracy for 17 Features, Differences and Runtime

NB_17	**Precision**	**Recall**	**F-Measure**
negative	59.50%	11.90%	19.80%
positive	57.70%	31.90%	41.10%
neutral	54.30%	**87.20%**	66.90%
Avg	56.40%	55.10%	49.30%
difference	**+4.40%**	**+1.70%**	**-2.00%**
J48_17	**Precision**	**Recall**	**F-Measure**
negative	59.30%	10.50%	17.80%
positive	59.30%	28.90%	38.90%
neutral	54.10%	**89.40%**	67.40%
Avg	56.70%	55.00%	48.40%
difference	**+2.00%**	**+0.20%**	**-1.80%**
RanFor_17	**Precision**	**Recall**	**F-Measure**
negative	56.90%	11.80%	19.50%
positive	60.10%	27.80%	38.00%
neutral	54.00%	**89.30%**	67.30%
Avg	56.40%	54.90%	48.50%
difference	**+4.70%**	**+1.60%**	**-0.70%**
SVM_17	**Precision**	**Recall**	**F-Measure**
negative	57.50%	9.90%	16.90%
positive	64.00%	23.10%	34.00%
neutral	53.40%	**92.20%**	67.60%
Avg	57.30%	54.60%	47.00%
difference	**-1.40%**	**-0.10%**	**+0.40%**

Table 5: Evaluation of Algorithms for 17 Features

Algorithm	Accuracy for training	Accuracy for Evaluation	Difference
NB_17	54.06%	54.05%	-0.01%
J48_17	55.30%	54.99%	**-0.31%**
RF_17	56.03%	54.93%	**-1.10%**
SVM_17	54.65%	54.60%	-0.05%

Table 6: Accuracy for Training and Evaluation for 17 Features

From Table 4 and Table 5, some useful information can be concluded.

The performance of Naïve Bayes improved especially for precision. The reason is that Naïve Bayes classifier is based on probability, which will perform bad when some features don't have strong relation with sentiment. Therefore, it works well after selecting better feature collection.

The performance of decision tree and Random Forest also improved because of the better feature selection, especially for Random Forest, because deleting some unrelated features will decrease a number of unrelated trees, which will lead to a better performance.

The performance of SVM slightly decreased because of the smaller number of features. It is effective in high dimensional spaces, therefore, it performed better in the original feature.

From Table 6, the difference between accuracy of training and evaluation decreased, which means that it relieved the phenomenon of overfitting. The problem of overfitting in Random Forest seems more serious, the reason is because there are not enough trees in the forest, which means that it should add more related features. Therefore, if we add more features in the model, Random Forest will work better.

4.3 Discussion

From the experiment, we compare four algorithms to study the problem of sentiment analysis in Twitter. Some information is concluded.

We find that Naïve Bayes algorithm is fast to train and predict and could get a competitive result for large number of data, which can make it as a baseline for classification task.

As for tree methods, decision tree is a bit slower than Naïve Bayes to train and predict, but also have a great performance. The limitation of decision tree is the feature selection, which could greatly affect the performance; also, it is easy to cause the phenomenon of overfitting. As for Random Forest, there is a limitation, which is that a large number of trees can make the algorithm slow for predictions. In other words, it is not slow to train, but time-consumed to create predictions. A more accurate prediction requires more trees, which will cause a slower model.

Support Vector Machine requires more time to train and predict, but it performs better than any other algorithms in this experiment; and we find that it is effective in high dimensional spaces.

Besides, we find some big challenges in tweet sentiment analysis. Firstly, we find that neutral tweets are more common than negative and positive. Therefore, it is hard to classify positive and negative class, which cause the low fraction of recall. Secondly, the noises are very common in tweets, so it is hard to get a great feature selection.

Based on papers, smiley could be a major factor for feature selection, which means a presentation of a facial expression using punctuation like ":)" and ":(". Some examples using smiley in tweets are as follows (see Table 7).

id	content
629849094777745408	Boruto: Naruto the movie is out in Japan and I have to wait till October to watch it):
637495730798628864	i'm still sad that shawn is coming tomorrow and im not going):
639189394910416896	And Shawn just said we can go to the commons on Friday so now I'm happy (:
639279231554334720	Going to see Ed Sheeran tomorrow with @abbyxrene and I'm super excited (:

Table 7: Some tweets using smiley

However, we find that there are only about 150 tweets containing smileys over 22,987 tweets, which is not a good idea to use smiley in training, especially for tree methods. Therefore, we didn't add smileys as features.

In conclusion, we find that it is possible to use tweet text to identify people sentiment on Twitter when we get a better feature selection.

5 Conclusion

In summary, this report applies 4 supervised learning methods to train, evaluate and predict tweet sentiment, which are Naïve Bayes, Decision Tree, Random Forest and Support Vector Machine.

We compared the evaluation result of four algorithms and observed that feature selection is very important for all algorithms, especially for Decision Tree and Random Forest; and we find that it is possible to use tweet text to identify people sentiment on Twitter by fixing the problem of feature selection, which will greatly increase the accuracy of algorithms.

For future work, the optimization of feature selection will be included.

6. References

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

Da Silva, N. F., Hruschka, E. R., & Hruschka Jr, E. R. (2014). Tweet sentiment analysis with classifier ensembles. Decision Support Systems, 66, 170-179.

Read, J. (2005, June). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of the ACL student research workshop (pp. 43-48). Association for Computational Linguistics.

Davidov, D., Tsur, O., & Rappoport, A. (2010, August). Enhanced sentiment learning using twitter hashtags and smileys. In Proceedings of the 23rd international conference on computational linguistics: posters (pp. 241-249). Association for Computational Linguistics.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In Proceedings of the Workshop on Language in Social Media (LSM 2011) (pp. 30-38).

Benediktsson, J. A., Chanussot, J., & Fauvel, M. (2007, May). Multiple classifier systems in remote sensing: from basics to recent developments. In International Workshop on Multiple Classifier Systems (pp. 501-512). Springer, Berlin, Heidelberg.

Kuncheva, L. I. (2004). Combining pattern classifiers: methods and algorithms. John Wiley & Sons.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.