COMP90049 Knowledge Technologies

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

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]) 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.

2. FEATURE ANALYSIS

2.1.ORIGINAL FEATURES ANALYSIS

The original data set has 46 features, which are generated by applying the methods of mutual information[2] and Pearson's X-Squared test[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.

Table.1 Estimate Naïve Bayes and 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

From *Table1*, some useful information can be concluded:

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.

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.

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 likely 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 does

[1] Weka: http://www.cs.waikato.ac.nz/ml/weka/

[2] http://en.wikipedia.org/wiki/Mutual_information

[3] http://en.wikipedia.org/wiki/Person_chi-sequared_test

not need a lot of data to perform well. Further, it can be assumed Naïve Bayes is less likely to overfit the training data with a smaller sample size. It just needs abundant data to understand the probabilistic correlation of each attribute in isolation with the output variable. On the contrary, J48 will work better with more data compared to Naive Bayes.

2.2. FEATURE ENGINEERING

2.2.1. GROOM FEATURES

As mentioned in 2.1, many features may not have a strong correlation with sentiment classes. In order to mitigate this situation, manually select some possible unrelated features, delete them and observe accuracy change to find if they have an impact on the two classifiers. The statistic result as shown in *Table.2*, all of them just delete one feature from original 46 features.

Table.2 The accuracy after deleting possible unrelated features for two types of classifier (%)

	Origin	id	a	antman	are
NB	54.4052	54.0601	54.4255	54.3849	54.4458
J48	55.5826	57.5849	55.8059	55.6435	55.6638
	at	cream	day	gucci	i
NB	54.4052	54.4255	54.2631	54.4255	54.7909
J48	55.7653	55.5826	55.2984	55.6029	56.5773
	ice	im	is	leftists	liberals
NB	54.6285	54.6082	54.5067	54.4052	54.3849
J48	55.5217	56.0089	56.2525	55.5014	55.9277
	my	national	nazi	night	obama
NB	53.9383	54.4661	54.3849	54.7300	54.3646
J48	55.9277	55.8668	55.6638	55.9683	55.7247
	people	racist	see	so	th
NB	54.4255	54.4052	54.4255	54.4052	54.4052
J48	55.5014	55.6332	55.7653	55.2781	55.5826
	they	trump	tomorrow	supremacists	
NB	54.7300	54.3443	54.5473	54.4052	
J48	56.2322	55.5826	56.4149	55.7044	

Note: Green values represent reference values. Red values mean related features to that classifier. Black values mean unrelated features to that classifier.

Therefore, original 46 features can be respectively deleted to fit Naïve Bayes and J48 classifier. Accuracy after deleting is as *Table.3*.

Table.3 The accuracy after deleting all unrelated features for two classifiers

	Origin	All Unrelated	Total Feature
NB	54.4052%	55.5623%	29
J48	55.5826%	56.4149%	25

Compare to original 46 features, 29 features for Naïve Bayes and 25 features for J48 can gain higher accuracy.

However, simply delete all unrelated attributes should not be a good choice, it's better to explore some combination of the probabilities of different attributes together and evaluate their performance at predicting the output variable. But, this report will just use the results shown in Table.3 in the following section.

2.2.2. PRUNED DECISION TREE

The statistic for J48 shown before is calculated from an unpruned tree. But pruning decision trees is a basic step to optimize the computational efficiency and classification accuracy for a model, because pruning is a technique in machine learning can reduce the size of the tree by removing sections of the tree that provide little power to classify in instance. Also, it can reduce the complexity of classifying, hence improves predictive accuracy by the reduction of overfitting.

Table.4 Compare the accuracy of pruned with unpruned J48 (minnumObj is omitted as minObj)

Unpruned	Accuracy	Pruned	Accuracy
minObj=2	56.4149%	minObj=2	56.6179%
minObj=5	56.4352%	minObj=5	56.6179%
minObj=10	56.6179%	minObj=10	56.6179%
minObj=15	56.6179%	minObj=15	56.6179%
minObj=20	56.5773%	minObj=20	56.5773%

Note: 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.

Table.5 Control "confidenceFactor" and "minnumObj" parameters to a pruned tree
M --- minnumObj   C---confidenceFactor

C \ M	2	5	10
0.1	56.5164%	56.4961%	56.5367%
0.25	56.6179%	56.6179%	56.6179%
0.3	56.6179%	56.6179%	56.6179%
0.4	56.6179%	56.5367%	56.6179%
0.5	56.4352%	56.5367%	56.6179%

Note: Both miniObj and confidenceFactor in weka are used for pruning a tree

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

J48 decision tree after pruning can always gain higher or equal accuracy.

Either unpruned or pruned J48, with the value of minnumObj increasing, the accuracy is firstly ascending then descending. Ascending implies the reduction of overfitting, since 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 possibly leads to an underfitting problem.

With the same minnumObj parameter, accuracy will firstly grow then decline when decreasing the confidence factor. This is reasonable because the confidence is used to calculate a pessimistic upper boundary on the error rate to a node. Smaller confidence means more pessimistic to the estimated error and will generally lead more pruning, which will result in over-training and let accuracy down.

2.3. NEW FEATURE ANALYSIS

2.3.1 EMOTICONS FEATURE

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]. Some tweets just use emoticons to express themselves instead of a real-word. Examples are shown in *Table.6*.

Table.6 Example of Emoticons in tweets

@Mirna_elhelbawi I'll do the same in november :) budapest &amp; milan Enjoy a lot
My copy of Iron Maiden's new album will be with me tomorrow :)
@MikeHudema Yep true, that's why they partied hard ! :) https://t.co/V7DIUn1VPZ
Absolutely mortified and gutted that Sam Smith may be doing the new Bond song :(
Forever jealous of those who are going to the Sam Smith concert tomorrow :(

The negative and positive emoticons set used in the report are as below *Table.7.*

Table.7 Example of Sentiment Emoticons

<table>
<tr><td>Negative_Emoticons</td><td>“:-(“, “:(”, “:-|”,”;-(”, ”;-<”, ”|-{”</td></tr>
<tr><td>Positive_Emoticons</td><td>“:-)”, “:)”,”:o)”,”:-}”, “:->”, “;-)”</td></tr>
</table>

After adding emoticon feature, the accuracy is described in *Table.8*.

Table.8 The accuracy after adding Emoticon feature

	Delete Unrelated	Add Emoticon	Add_Emo (Balance)
NB	55.5623%	55.5826%	56.7600%
J48(unpruned)	56.4149%	56.7803%	54.9376%
J48(pruned)	56.6179%	56.9428%	55.0142%

Note: “Balance” in Table.8 means using SMOTE Filter in weka for oversampling the instance so as to balance the instance in the majorclass and minorclass.

From *Table.8,* some information can be concluded:

After adding “Emoticon” feature, the accuracy has increased on either Naïve Bayes or J48. But only to Naïve Bayes, the accuracy increased after balancing. The reason for accuracy increasing in Naïve Bayes is possibly because the training data for emoticon feature are unbalance and not enough (Table.9). Also, just 0.8% of the testing data contains emoticons, so it is reasonable that there is not a big increase.

Table.9 Statistic for Emoticons on given dataset

	Pos	Neg	Total_Emo	Total_Tweets
train-tweets	136	41	177	22988
dev-tweets	31	9	40	4927
test-tweets	35	7	42	4927

The reason why J48 does not perform well after balancing is possibly corresponding to the same assumption as depicted in 2.1, which is that J48 would perform worse with low amounts of data compared to Naïve Bayes.

2.3.2 BIGRAM FEATURE

Bigram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllables, or words[5].

Firstly uses training data to generate 100 highest frequency bigrams and record each bigram’s frequency for each classes of sentiment. Then use these bigram sets to label each tweet:

i. Let each tweet generate its own bigrams

ii. Compare its own bigrams with training bigram sets.

iii. Label this tweet with one sentiment class. (Choose higher frequency bigram if make a tie)

The result is shown in *Table.10*.

Table.10 The accuracy after adding Bigram feature

	Delete Unrelated	Add Bigram	Add_Bigram & Emoticon
NB	55.5623%	56.5367%	56.5570%
J48(unpruned)	56.4149%	57.1458%	57.3082%
J48(pruned)	56.6179%	57.4097%	57.6127%

From *Table.10*, some information can be concluded:

After adding “Bigram” feature, the accuracy increases slightly on both Naïve Bayes and J48. Also, it can obtain higher accuracy after adding both “Emoticon” and “Bigram” features.

[4] Emoticon:http://en.wikipedia.org/wiki/Emoticon

[5] Bigram: https://en.wikipedia.org/wiki/Bigram

Since Naïve Bayes is more sensitive to unbalance data comparing to J48 from the conclusion on 2.3.1, so use SMOTE filter to balance data again and obtain 56.9022% accuracy for Naïve Bayes, which confirms the assumption again.

After combining "Emoticon" and "Bigram" features, the accuracy of Naïve Bayes decreases. It possibly means those two features has correlations, since the performance of Naïve Bayes can degrade if the data contains related features.

3. CONCLUSION

This report uses Naïve Bayes and J48 classifiers' accuracy to analyze the given dataset on some original given features and new features. After improving the performance, the accuracy of Naïve Bayes and J48 is, 56.9022% and 57.6127% respectively. There are many reasons for that accuracy is still not high enough, one is that not many tweets contain emoticons and just choose 100 highest frequency bigrams for each types of sentiment will occur bias. Besides, testing the data set with Vader sentiment analysis library[6], which specially aims to analyze sentiment for tweets, the given accuracy is about 80%, which is not high enough as well. But Vader considers many other features that the report does not implement so that makes a different accuracy between the report's results and Vader's.

4. 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)

[6] vader sentiment analysis: https://github.com/cjhutto/vaderSentiment