Comp90049 Knowledge Technology

Project 2

Identifying Tweets with Adverse Drug Reactions Report

5/10/2017

Introduction

The report primarily illustrated the process of new attribute generator, how it cooperates with the data mining software Weka to promote the precision of determining ADR (Adverse Drug Reactions) tweets and the evaluation of the training output. First, the attribute generator program aimed to clean and evaluate the train and test tweet data sets which are provided by Abeed and Graciela (2015) through two algorithms. Then, storing the assessment result in several txt files as the values of the new attribute to create new arff files. Lastly, the Weka tested, evaluated and gained an output according to these arff files.

Brownlee (2017) stated that the appropriate feature design is able to improve the precision rate of the machine learning. Thus, although the Weka played the dominant role of training, testing and evaluation, the design of generating new attributes is actually the fundament to enhance the accuracy.

Description of the system's behaviour

The system is implemented in Java and contains one Main function with two algorithm functions, **New_attribute_length** and **New_attribute_nega**.

These two algorithm functions are similar in the aspects of files read-write since their goals are labelling 'Y' or 'N' for each tweet and consisting of a new attribute.

The difference between them is the method adopted. In the New_attribute_length function, based the length of words, the program judges whether a word satisfies the requirement that has more 11 letters since some words of drug names and symptoms are less than 11 letters. Hence, the first step is to clean the data as the punctuation attached in some words may influence the generated result. Next, it begins to search and contrast one by one to determine the label for each tweet. Finally, store all the labels in the file result_long_words.txt.

By contrast, the New_attribute_nega function determines whether there exist negative words (not, "'t", cannot) in each tweet. I added this attribute since I selected and browsed some tweets in the file and found some users may use negative words to express the emotions and adverse reactions. Secondly, it should be noted that there are many unrelated words may contain 'n', 'o', 't' letters as well, such as notice, note, another…etc. Hence, I utilised the method word[i].equals("not") rather than word[i].indexof("not"). Then, similarly, save the results of labels to the file result_nega_words.txt.

After adding these two columns to the arff files, we obtain these new train-dev files sets:

1. **dev_longwords.arff & train_longwords.arff**

   They are the original dev.arff and train.arff plus the length attribute respectively.

2. **dev_nega.arff & train_nega.arff**

   They are the original dev.arff and train.arff plus the nega attribute respectively.

3. **dev_combination.arff & train_combination.arff**

   They are the original dev.arff and train.arff plus the length and nega attributes respectively.

The last step is to analyse and evaluate these data. In the Weka, use the train.arff (train_longwords.arff…etc.) as the training model, the dev.arff (dev_longwords.arff…etc.) as the 'Supplied test set', select the classifier "Naïve Bayes", then the Weka will generate a new txt file to store the prediction result (Hall et al., 2009).

Evaluation

I used the Weka to output CSV files with the precision for each tweet and accuracy of different attribute groups.

```
=== Summary ===

Correctly Classified Instances         884               82.1561 %
Incorrectly Classified Instances       192               17.8439 %
Kappa statistic                          0.2676
Mean absolute error                      0.2067
Root mean squared error                  0.3873
Relative absolute error                103.8938 %
Root relative squared error            125.7529 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.862    0.518    0.934      0.862    0.896      0.279
                 0.482    0.138    0.293      0.482    0.364      0.279
Weighted Avg.    0.822    0.477    0.866      0.822    0.840      0.279

=== Confusion Matrix ===

   a   b   <-- classified as
 829 133 |   a = N
  59  55 |   b = Y
```

Figure 1. The result of accuracy of the original data set.

As the figure above, the original data set (train.arff & dev.arff) provided by Abeed and Graciela (2015) demonstrated an excellent accuracy, which reached 82.16%. Utilise it as the reference, the accuracy of new arff files with added attributes decreases slightly. To sum up, the selection and increase of attributes play the significant role in enhancing the efficiency of the data mining and the learning algorithm (Hall and Holmes, 2003).

1. The words length attribute

```
=== Summary ===

Correctly Classified Instances         883               82.0632 %
Incorrectly Classified Instances       193               17.9368 %
Kappa statistic                          0.266
Mean absolute error                      0.2069
Root mean squared error                  0.3875
Relative absolute error                103.9512 %
Root relative squared error            125.8002 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.861    0.518    0.933      0.861    0.896      0.278
                 0.482    0.139    0.291      0.482    0.363      0.278
Weighted Avg.    0.821    0.477    0.865      0.821    0.839      0.278

=== Confusion Matrix ===

   a   b   <-- classified as
 828 134 |   a = N
  59  55 |   b = Y
```

Figure 2. The result of accuracy of the long-words attribute data set.

After added the length attribute, the precision results showed an undulation, and the accuracy decreases to 82.06%. Find and select the tweet which become the incorrectly classified instance:

“Purata and seroquel yet im not asleep!!!!! Please dont take over my life again! #cry Work damnit!!!” (Abeed and Graciela, 2015)

Weka forecasted this tweet contains ADR and the ability of prediction of TP decreased. However, the prediction is incorrect. Some possible factors are influencing the predicted result. First, some long words with 11 or more letters, such as responsibility, are not related to the ADR and medicine which interfere the training and forecasting process of Weka.

The second reason probably is that the attribute I selected tended to predict the tweet does not contain the ADR, which led to the decrease of ability of TP prediction.

2. The negative words attribute

```
=== Summary ===

Correctly Classified Instances         880               81.7844 %
Incorrectly Classified Instances       196               18.2156 %
Kappa statistic                          0.2613
Mean absolute error                      0.2076
Root mean squared error                  0.3887
Relative absolute error                104.306  %
Root relative squared error            126.1922 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.858    0.518    0.933      0.858    0.894      0.273
                 0.482    0.142    0.286      0.482    0.359      0.273
Weighted Avg.    0.818    0.478    0.865      0.818    0.837      0.273

=== Confusion Matrix ===

   a   b   <-- classified as
 825 137 |   a = N
  59  55 |   b = Y
```

Figure 3. The result of accuracy of the negative-words attribute data set.

Similarly, the accuracy decreases to 81.78%, and select tweets which become the incorrectly and correctly classified instances:

“@synapsid oh God yeah effexor is a med where skipping a single dose wrecks you because it doesn't stay in your system” (Abeed and Graciela, 2015)

“India is a beautiful, amazing, exhilarating country! Our indulgences have caught up with us, but thank goodness for Cipro!” (Abeed and Graciela, 2015)

Actually, both of them do not include ADR, just mentioned the name of drugs. This time only the Weka’s prediction for the second tweet is correct. However, compared to the original result, Weka predicted the first tweet includes ADR. The reason may be that the negative word ‘not’ and “ 't ” are not related to ADR or a symptom, which affect the judgement of the Weka. In addition, the prediction of TP decline continually as the negative-word attribute also aims to predict a tweet does not include ADR.

3. The combination attribute

```
=== Summary ===

Correctly Classified Instances         879               81.6914 %
Incorrectly Classified Instances       197               18.3086 %
Kappa statistic                          0.2553
Mean absolute error                      0.2077
Root mean squared error                  0.3888
Relative absolute error                104.3646 %
Root relative squared error            126.2367 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.858    0.526    0.932      0.858    0.893      0.267
                 0.474    0.142    0.283      0.474    0.354      0.267
Weighted Avg.    0.817    0.486    0.863      0.817    0.836      0.267

=== Confusion Matrix ===

   a   b   <-- classified as
 825 137 |   a = N
  60  54 |   b = Y
```

Figure 4. The result of accuracy of the combination-attribute data set.

Unfortunately, the accuracy decreases to 81.69% and one more tweet become incorrectly classified instance. The reason may be that there are various attributes which cause the overfitting problem. Overfitting refers to the over-training, which means that the hypothesis over approach training data and lead to the increase of error (CHEN, CHNG and ALKADHIMI, 1996). Meanwhile, the combination of these two attributes illustrated low performance which tried to predict a tweet contains the ADR.

Overall, the Naïve Bayes has plenty of useful features and indicates satisfactory performance during the experiment. The Naïve Bayes algorithm is simple, legible and interpretable, and demonstrate high performance in the medicine area (Mayo and Frank, 2017). However, the accuracies still decrease in these three circumstances. Fortunately, from the output and observation, some precisions are stable and maintain in 1, and large amounts of instances illustrate correct prediction. Apart from the irrelevant attributes influence the Weka, the features of the classifier also lead to the decrease. The attributes in the Naïve Bayes classifier are set to be independent (Zaidi et al., 2013). Thus, the interrelations of these attributes are not solid, which results in the decline.

Conclusion

The selection and implement of attributes are able to affect the predicted result, and efficient selection will increase the precision of machine learning algorithm.

The Naïve Bayes classifier provides the high predicted precision. However, in the actual sample test, the practical results sometimes are opposite to the prediction. There exist limitations in this experiment that building attributes and testing accuracy. To improve the precision and effectiveness, it requires better training model (artificial dataset for training Naïve Bayes model) (Mayo and Frank, 2017). On the one hand, it should increase the capacity of the instances, and delete and remain more accurate attributes. On the other hand, applied regularisation method, which refers to explore a balance between making hypothesis better fit to the training data and maintaining the minimum of the attributes (Cogswell et al., 2015). Moreover, the tweet dataset should be vocabulary correction before utilised to test.

Lastly, the Naïve Bayes maybe not the most appropriate classifier for the circumstance. Adopt decision tree (J48) instead should obtain higher performance (Patil and Sherekar, 2013).

Reference

1. Abeed Sarker and Graciela Gonzalez. (2015) *Portable automatic text classification for adverse drug reaction detection via multi-corpus training.* Journal of Biomedical Informatics, 53: 196-207.
2. Brownlee, J. (2017). *Discover Feature Engineering, How to Engineer Features and How to Get Good at It - Machine Learning Mastery*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/ [Accessed 5 Oct. 2017].
3. CHEN, S., CHNG, E. and ALKADHIMI, K. (1996). Regularized orthogonal least squares algorithm for constructing radial basis function networks. *International Journal of Control*, 64(5), pp.829-837.
4. Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., and Batra, D. (2015). Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*.
5. Hall, M. and Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6), pp.1437-1447.
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and

Witten, I. (2009). The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, 11(1), p.10.

7. Mayo, M. and Frank, E. (2017). Improving Naive Bayes for Regression with Optimised Artificial Surrogate Data. *arXiv preprint arXiv:1707.04943*.
8. Patil, T. R., and Sherekar, S. S. (2013). Performance analysis of Naive Bayes and J48 classification algorithm for data classification. *International Journal of Computer Science and Applications*, *6*(2), pp.256-261.
9. Zaidi, N. A., Cerquides, J., Carman, M. J., and Webb, G. I. (2013). Alleviating naive Bayes attribute independence assumption by attribute weighting. *The Journal of Machine Learning Research*, *14*(1), pp.1947-1988.