COMP90049 Project 2 Report: Identifying Tweets with Adverse Drug Reactions

---- Yifan Wang 809706

1. Introduction

This project aims to assess on the performance of some supervised machine learning methods to determine whether a tweeter contains an ADR (Adverse Drug Reactions). By finding one or more non-word unigram features and generating the new train file and dev file, the quality of the newly found features will be judged using the weka machine learning tool.

2. Dataset

The train, develop and test data is from Twitter, and is an altered form of a dataset from the DIEGO Lab (Gonzalez, 2015).
In this project, the programs were implemented in Java and used some data from some websites, including Wikipedia and Medscape. The detailed usage wull be introduced in the following pages and has been cited in the readme file as well as the reference section.

3. Classifier Selection

The classifier used in this project is Naïve Bayes. Naive Bayes classifier is based on the Bayes' theorem with strong naive independence assumption between features, it is simple to build and fast to make decision, also is easy to scale to large data size(Verspoor, 2017). By using the Naïve Bayes classifier embedded in the Weka tool, the classiffication process is accomplished accordingly.

4. Methodology and evaluation explanation

4.1. Methodology

As indicated in the specification, the main task is to find one or more new attributes and evaluate the performance of that attribute for finding ADRs. The whole logic was accomplished in the following three steps.

a. Generate new features.

The new features need to be non-word unigram, because the best 92 tokens which were selected by their frequencies has been selected and listed in the ARFF files, therefore there is no need to select features using the token frequency again.

The aim of the feature is to distinguish sentences that contain ADR from those not. By viewing the given TXT files, there is a common regularity for those sentences that labelled with 'Yes', that is they mostly contain one specific body part, or contain a kind of bad symptom, or contain a name of a medicine, or contains more than one of these elements. Therefore in this project, the firstly selected features are:

- isBodyPart: Whether body part occurs in the tweet
- isDrugName: Whether drug name occurs in the tweet
- isSymptom: Whether one symptom occurs in the tweet.

After firstly extracting these three profound features, the next step is to evaluate them and combine them to form a new feature then evaluate it. The detailed process will be illustrated in the result section.

b. Update the train and dev files.

In this step, the new found features will be added into the train ARFF file and the dev ARFF file. Then the reduced VSM over the tweets need to be updated to adapt the new features.

This step is implemented in Java. The train ARFF file updating process is listed as the following steps and the dev ARFF file updating is in the same process.

i. Set three Arraylists (BodyParts, DrugName and Symptom) to store the word lists for judging whether one tweet contains this feature. The elements for body parts were extracted from http://www.enchantedlearning.com/wordlist/body.shtml, the drug names were found from http://www.medscape.com/viewarticle/825053 which indicates the top 100 saled drugs, the symptom word list is found in wekipedia.

ii. Read attributes from train.arff, write the header and attribute lines to the new ARFF file. This step appends new attributes behind the origin file.

iii. Read and update the VSM to the new ARFF file. Since the attributes represents whether the tweet contains the feature, if the tweet contains one element of the Arraylist then the value is 'Y', otherwise 'N'.

By completing the three steps could these three attributes updated to the new train ARFF file and the new dev ARFF file.

c. Evaluate using the Weka tool

Weka could help to train the model and analyse the performance. The evaluation result contains the value of the correctly classified instances and the incorrectly classified instances, and the detailed statistical data including the TP rate, FP rate, precision, recall, F-measure and so on for class N and class Y separately. The confusion matrix would also displayed to indicate the differences between the expectation and reality circumstance.

4.2. **Evaluation**

To assess whether the new features could improve the performance, one of the evaluation index is the correct rate. However in the given train.txt file, there are only 373 tweets that labelled with "Y" from 3166 tweets, the percentage of the class 'Y' and class 'N' are not balanced so the accuracy is highly affected by the performance of class 'N'.

Therefore the accuracy is insufficient to describe the performance, the precision for the two classes are important supplements. The TP rate is the true positive rate whose value is the same with the recall rate, it could judge the performance of a model hitting the correct class.

5. **Result**

By executing the Java programs and assessing the output ARFF files using Weka, the result for each features is list as the following captures.

1) Add no new feature, see figure 1.

```
=== 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      ROC Area  PRC Area  Class
                 0.862    0.518    0.934      0.862    0.896      0.279    0.758     0.961     N
                 0.482    0.138    0.293      0.482    0.364      0.279    0.757     0.228     Y
Weighted Avg.    0.822    0.477    0.866      0.822    0.840      0.279    0.758     0.884

=== Confusion Matrix ===

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

Figure 1. Add No new Feature

2)Only add the isBodyPart feature.

```
=== Summary ===

Correctly Classified Instances         885               82.2491 %
Incorrectly Classified Instances       191               17.7509 %
Kappa statistic                          0.2736
Mean absolute error                      0.2055
Root mean squared error                  0.3862
Relative absolute error                103.2497 %
Root relative squared error            125.3937 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.862    0.509    0.935      0.862    0.897      0.285    0.758     0.961     N
                 0.491    0.138    0.296      0.491    0.370      0.285    0.758     0.230     Y
Weighted Avg.    0.822    0.470    0.867      0.822    0.841      0.285    0.758     0.884

=== Confusion Matrix ===

   a   b   <-- classified as
 829 133 |   a = N
  58  56 |   b = Y
```

Figure 2. Only add the isBodyPart feature

3) Only add the isDrugName feature, see figure 3.

```
=== Summary ===

Correctly Classified Instances         884               82.1561 %
Incorrectly Classified Instances       192               17.8439 %
Kappa statistic                          0.2676
Mean absolute error                      0.2068
Root mean squared error                  0.3875
Relative absolute error                103.9258 %
Root relative squared error            125.8222 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.862    0.518    0.934      0.862    0.896      0.279    0.758     0.962     N
                 0.482    0.138    0.293      0.482    0.364      0.279    0.758     0.228     Y
Weighted Avg.    0.822    0.477    0.866      0.822    0.840      0.279    0.758     0.884

=== Confusion Matrix ===

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

Figure 3. Only add isDrugName feature

4) Only add the isSymptom feature.

```
=== Summary ===

Correctly Classified Instances         884               82.1561 %
Incorrectly Classified Instances       192               17.8439 %
Kappa statistic                          0.272
Mean absolute error                      0.2043
Root mean squared error                  0.3852
Relative absolute error                102.6574 %
Root relative squared error            125.0574 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.861    0.509    0.935      0.861    0.896      0.284    0.764     0.963     N
                 0.491    0.139    0.295      0.491    0.368      0.284    0.764     0.235     Y
Weighted Avg.    0.822    0.470    0.867      0.822    0.840      0.284    0.764     0.886

=== Confusion Matrix ===

   a   b   <-- classified as
 828 134 |   a = N
  58  56 |   b = Y
```

Figure 4. Only add isSymtom feature.

5) Combine the isBody and isDrugName feature to the new isBody_Drug feature and only add this feature, see figure 5.

```
=== Summary ===

Correctly Classified Instances         884               82.1561 %
Incorrectly Classified Instances       192               17.8439 %
Kappa statistic                          0.2676
Mean absolute error                      0.2068
Root mean squared error                  0.3874
Relative absolute error                103.9214 %
Root relative squared error            125.7652 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.862    0.518    0.934      0.862    0.896      0.279    0.758     0.962     N
                 0.482    0.138    0.293      0.482    0.364      0.279    0.758     0.228     Y
Weighted Avg.    0.822    0.477    0.866      0.822    0.840      0.279    0.758     0.884

=== Confusion Matrix ===

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

Figure 5. Add the isBody_Drug feature

6) Combine the isBody and isSymptom feature to the new isBody_Symp feature and only add this feature, see figure 6.

```
=== Summary ===

Correctly Classified Instances         886               82.342  %
Incorrectly Classified Instances       190               17.658  %
Kappa statistic                          0.2796
Mean absolute error                      0.2039
Root mean squared error                  0.3851
Relative absolute error                102.4641 %
Root relative squared error            125.0283 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.862    0.500    0.936      0.862    0.897      0.292    0.763     0.963     N
                 0.500    0.138    0.300      0.500    0.375      0.292    0.763     0.234     Y
Weighted Avg.    0.823    0.462    0.868      0.823    0.842      0.292    0.763     0.886

=== Confusion Matrix ===

   a   b   <-- classified as
 829 133 |   a = N
  57  57 |   b = Y
```

Figure 6. Add the isBody_Symp feature

7) Combine isDrugName and isSymptom feature to the isDrug_Symp feature and only add this feature, see figure 7.

```
=== Summary ===

Correctly Classified Instances         902               83.829  %
Incorrectly Classified Instances       174               16.171  %
Kappa statistic                          0.3363
Mean absolute error                      0.1888
Root mean squared error                  0.3657
Relative absolute error                 99.3628 %
Root relative squared error            118.8132 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.871     0.439     0.944      0.871     0.906      0.351     0.784     0.966     N
               0.561     0.129     0.340      0.561     0.424      0.351     0.784     0.280     Y
Weighted Avg.  0.838     0.406     0.880      0.838     0.855      0.351     0.784     0.893

=== Confusion Matrix ===

   a   b   <-- classified as
 838 124 |   a = N
  50  64 |   b = Y
```

Figure 7. Add the isDrug_Symp feature.

8) Combine all these three features to the is_All feature and only add this feature, see figure 8.

```
=== Summary ===

Correctly Classified Instances         886               82.342  %
Incorrectly Classified Instances       190               17.658  %
Kappa statistic                          0.2796
Mean absolute error                      0.2039
Root mean squared error                  0.3851
Relative absolute error                102.4641 %
Root relative squared error            125.0283 %
Total Number of Instances             1076

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.862     0.500     0.936      0.862     0.897      0.292     0.763     0.963     N
               0.500     0.138     0.300      0.500     0.375      0.292     0.763     0.234     Y
Weighted Avg.  0.823     0.462     0.868      0.823     0.842      0.292     0.763     0.886

=== Confusion Matrix ===

   a   b   <-- classified as
 829 133 |   a = N
  57  57 |   b = Y
```

Figure 8. Add all features together.

Compared with the original dataset with the given 94 features, the performance did not improve prominently when single feature (isBodyPart, isDrugName, isSymptom) was added. The isBodyPart feature only added one correctly classified instance and improved the TP Rate, precision and recall rate to a minor degree. The isDrugName feature has no influence at all. The isSymptom feature only improved the TP Rate, precision and recall rate for class Y to a small extend.

When single features were combined as a new one and added to the origin features, the performance is overall improved except the combination of isBodyName and isDrugName feature. The best performance feature is the isDrug_Symp feature, which added 16 correctly classified instances and improved the TP Rate, precision and recall rate for both Class N and class Y in a larger extend.

6. Evaluation

6.1. Result Evaluation

From the summarization above, we could find the following three characteristics:

- Single attributes did not improve the performance well.
- Combining two single features to one new feature improved the performance, especially the combination of the isDrugName and isSymptom feature.
- Combining three attributes to one improved the performance while not as good as the combination of isDrugName and isSymptom.

The reasons that might lead to this result is listed as followed.

- Judging whether the tweet has the single feature is inaccurate because the lists for attribute judging is not fully scaled. For instance, the drug name list only contains the top 100 sold medicines which is insufficient to cover all drugs. So the judge itself could be incorrect.
- The vector space contains about nighty five columns for the attributes, the value of one dimension is not able to impact on the whole result if its classification performance is not good enough.
- The combination of isDrugName and is isSymptom could distinguish the classification better than other combination, because this relation is more close to the ADR. If a tweet contains at least one drug name and symptom, it is more likely to contain ADR.
- The reason why combining all three attributes did not perform so well might due to the occurrence of the body part, which means the body part feature is less related to the classification as the drug name collated with the symptom name.

6.2. Critical analysis

The tweets which contain ADR are more likely to express the bad emotion, which could be used as another feature. The sentiment analysis could be considered to be used in this case. During the implementation, the textblob lib in Python was used however the polarity for most tweets were 0 since the model has not been trained. Thus the method was not used here.

7. Conclusions

This report introduced some new features and their combinations for the Weka machine learning tool to classify whether a tweet contains ADR, and researched on their influence on the system's performance with the help of Weka. The feature that whether the drug name and symptom occurred

together in the tweet has the best performance, which could be the result of the close relationship between the feature and the ADR.

References

Gonzalez, G. and Sarker, A. (2015). Portable automatic text classification for adverse drug reaction detection via multi-corpus training. Journal of Biomedical Informatics, 53: 196-207.

Verspoor, S. E. (2017). Retrieved from https://app.lms.unimelb.edu.au: https://app.lms.unimelb.edu.au/bbcswebdav/pid-5863190-dt-content-rid-26337415_2/courses/COMP90049_2017_SM2/lectures/14-classification.pdf

Wekipedia. (2017, October 7). *Random forest*. Retrieved from https://en.wikipedia.org: https://en.wikipedia.org/wiki/Random_forest

Brooks, M. (2014,May 13). Top 100 Most Prescribed, Top Selling Drugs. Retrieved from http://www.medscape.com/: http://www.medscape.com/viewarticle/825053

Wikipedia. (2017,August 21). List of medical symptoms. Retrive from https://en.wikipedia.org/wiki/List_of_medical_symptoms