COMP90049 Project 2 Report

Identifying Tweets with Adverse Drug Reactions

1. Introduction

The aim of the project is to identify tweets which show signs of adverse drug reactions (ADR) [2] using machine learning algorithms. A tweet which shows the sign of ADR is classified as Y (Yes), otherwise N (No). This analysis can help in observing any injury occurred by taking a medication or by organizations to study the review of their products. Organizations can then, give service to users who experienced such problems expressed via social media.

The dataset [1] used in this project as training set is the collection of 3166 tweets which is used to train the machine learning method, development set consist of 1076 tweets used to evaluate machine learning method and test set consists of 1087 tweets for which ADR predictions to be made.

2. Literature Review

Drug, a chemical substance used to cure a disease. Everybody need and use drugs every now and then. But they can be beneficial or have unwanted side-effects, called adverse drug Reactions (ADR). Although organizations run expensive clinical trials to identify these ADRs, still serious ADRs exist. Identifying ADR is very crucial for government agencies, drug manufacturers, and public health. One way is through clinical trials, while on the other hand social media like, twitter. Millions of tweets discuss ADR on a day-to-day basis. These tweets contain discussion about side effects, advantages of a drug. Twitter contains a large set of unlabelled training data. For e.g.

Ofloxacin is great for infections. (No ADR)

Went 2 doc, ofloxacin caused stomach pain. #no use (ADR)

There is an immense research in this area, due to growing use of internet for expressing user experiences about ADR. From a learning perspective, it is important to explore some supervised machine learning methods like, Naive Bayes, Support Vector Machines, etc. to address the problem of ADR in tweets.

3. Analysis

The dataset have six files, train.txt, train.arff, dev.txt, dev.arff, test.txt, and test.arff.

Using the train.txt file, I was able to find attributes (out of existing 93 including id) that can help in comparing different supervised machine learning methods. For e.g. a bigram 'side effects' occurred in 64 tweets, out of which only 9 were classified 'Y'. Out of those 64 probability of occurring a 'N' prediction for an ADR is high. Attributes having similar occurrences of 'Y' and 'N' can be ignored.

Frequencies of those attributes occurring in tweets contained in train.txt, dev.txt, and test.txt files are collected and added in respective .arff and used to evaluate supervised machine learning methods using Weka.

3.1 Naïve Bayes Classifier

Naïve Bayes [3] is a learning and classification method based on probability theory. It is based on Bayes theorem with naïve independence assumptions between the features.

Naïve Bayes Classifier is applied in the bayes folder of the classifiers in Weka. By applying this classifier on the development set it gives the accuracy of 82.1561%, i.e. it has correctly classified 82.1561% of tweets with respective classes. Naïve Bayes learns from the frequencies of the attributes in training set and calculates the frequency of the attributes in the tweets and based on these frequencies predicts the ADR and non-ADR tweets.

From the confusion matrix and true positives (TP) rate, TP rate of class 'N' is high, i.e. more tweets with class 'N' is rightly classified as 'N' because Naïve Bayes calculates the prior probability of the class given a description of an instance. Number of tweets belonging to class 'N' is more as compared to class 'Y', resulting in more prior probability and chances of tweets getting classified to 'N' increases.

One problem with Naïve Bayes is having no occurrences of a certain attribute and class label value together. For e.g. in case of tweet, let say, class = 'N', and label = 'feeling sleepy' never occurs together, then the frequency-based probability calculation will be zero. This in turn effect the posterior probability estimate because of conditional independence assumption of Naïve Bayes, after multiplying all probabilities will get result zero. This problem also effects with accuracy of Naïve Bayes as compared to other classifier models. This problem can be solve by

Lagrange correction. Recall and precision were 0.822 and 0.866 respectively.

3.2 **Decision Tree (J48) Classifier**

Decision Tree [5] is a chart-like structure where, internal node denotes test on attribute, branch gives result of test, and leaf nodes represent class labels.

J48 (Decision tree) Classifier in trees folder of Weka is based on C4.5 algorithm [6]. It uses Information Gain as a criteria for splitting decision trees. It gives the accuracy of 89.0335%.

Time taken to build the model is added with decision tree as compared to Naïve Bayes because it has extra overhead of building different decision trees, then selecting the optimal one to classify data. But, once the model is made, time to test the model is very less. It constructs the optimal decision tree using training set and classify development set tweets from frequency of features in tweets.

Decision tree has significant problems of overfitting and underfitting. For e.g. classifier becomes prone to noise or unwanted data after learning the training data too well leads to overfitting. Let say, we take attributes like, 'is', 'if', 'the' etc. all these are unwanted attributes giving wrong result. This will reduce accuracy of decision tree. Solution will be to trim the unwanted branches of tree. Confusion matrix shows the classifier correctly predicts most of class 'N' tweets. Recall and precision are 0.890 and 0.851 respectively.

3.3 **Support Vector Machines**

Support Vector Machines (SVM) [4] is a classifier that takes a labelled training data and gives an optimal hyperplane that categorizes new features.

This project consists of multi-class, i.e., Y, and N. We need few SVMs for classification: Negative vs positive (SVM 1), and positive vs negative (SVM 2).

SMO is selected by clicking on functions folder of classifier in Weka. Learning through SVM is expensive and it adds computation overhead. It gets slow for test data which is not good a performance factor. It takes 1.4 seconds to build model. Although it gives the accuracy of 89.4052% which is better than other two algorithms. But, on the other hand suffers from overfitting, i.e. adding unwanted data to the result set.

4. **Conclusions**

Machine learning algorithms provide significant accuracy in identifying tweets with adverse drug reactions. Some machine learning algorithms add computation like, SVM and J48 taking time for classification and optimization. But, once the model is build, testing happens quickly. In case of Naïve Bayes, it is the first choice in terms of classifiers because of the simple approach, but it has his own disadvantages due to conditional independence assumption.

However, in order to provide a well-defined solution to the problem, some research is needed. For e.g. use of sarcastic tweets, tweets using images, videos in terms of ADR. Machine learning methods should improve their performance with respect to such tweets. The accurate prediction of ADR can improve public health as well as help organizations to monitor their products.

References

[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] Adverse Drug Reaction. https://en.wikipedia.org/wiki/Adverse_drug_reaction.

[3] Naïve Bayes. https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[4] Support Vector Machines. https://en.wikipedia.org/wiki/Support_vector_machine.

[5] Decision Tree. https://en.wikipedia.org/wiki/Decision_tree.

[6] C4.5 algorithm. https://en.wikipedia.org/wiki/C4.5_algorithm.