Natural Language Processing - ECS763U/ECS763P
Assingment 1 Report

Fake News Detection

Gitesh Deshmukh(220900436)
School of Electronic Engineering and Computer Science
Queen Mary University of London
Mile End Road, London E1 4NS, UK

Abstract

As mentioned Fake news is a sound problem in modern online life and having methods to automatically detect is becoming increasingly important. In this coursework, we have built and evaluated a classifier which detects fake news. Step by step we will dive deeper into each step and try to comprehend it.

Simple Data Input and Pre-Processing

The Parse Data Line is used to convert the data format. It returns label and text where a label 'REAL' will represent real news and the label 'FAKE' as fake news. For this convert_label function is implemented to segregate the two variant labels. The convert_label function returns the label_value which consists of the parameters. Further I have used "re" function for the string and have defined the pre-process function in order to perform processing on the desired data. The lower keyword converts all the uppercase text into the lower ones to make it easier to work on data. Using the Natural Language toolkit i.e. NLTK, the tokenising for the text is done.

Simple Feature Extraction

In the question 2, a dictionary named global_feature_dictionary is created to keep the track of all the tokens and the name occurring in the complete dataset. The to_feature_vector is predefined in the dataset where a feature_dict is created to apply if-else conditions on the tokens dataset. The two dictionaries stores the subsequent token values based on the if else loop applied to it separately. The Support Vector Classifier (SVC) is been used to classify the text i.e. the tokens. The train classifier fucntion ensures the validation of the whole classifier method.

Cross-validation on Training Data

Using cross validation method, we aid our machine to generate the potential to predict new data. Each element of the list or the dictionary stores the subsequent count of that those elements as the values.
The k-folds simply divides the data into smaller samples. The cross validate function passes two parameters in dataset and folds in the function. The test_token and train_token variables are used to sample the data in distinct variables. The train1 takes the i:i+ fold_size data whereas test1 takes the total of :i and i+ fol_size data.

The mypredict variable stores the values of labels after it takes two arguments in the function. In the last section, various variables are defined where result is stored in the myresults and called as we return the average values using mean function.
In the main cell, using data_file_path as a

reference the tsv file named fake news is recalled and load data function pre loads the data in the desirable format. The parsed dataset is initiated using the load_data function.

The split_and_preprocess_data function helps to split the dataset into the training data set and the testing data in 80% and 20% respectively after passing the value. We then print the training samples and the features of the train and raw data using the print function.

Error Analysis

The already defined confusion matrix heatmap is implemented to plot a graph showing the ‘REAL’ and ‘FAKE’ count on the graph. The variable ‘g’ stores the desired train_data and is been called in new variables in mytest and mytrain. Using the “for loop”, the testing data is appended and the testing prediction too. The Data is recalled using another store variable in myprediction where predict_labels shows the data. Finally using the confusion_matrix_heatmap method, we could see the graph observations for the certain values. The training classifier prints the values as Precision: 0.564841, Recall: 0.565642, F Score:0.565188