DATAROBOT

Report on Use of Google Prediction API to make Predictions,

Submitted by,

Sourabh Chougale.

Email: sourabhchougale.99@gmail.com

Illinois Institute of Technology

Chicago, IL, USA

The project is about making use of Google Prediction API to build a prediction model and make prediction on our own data. We can do many things using google prediction API. I worked on sentiments predictions using twitter data. I collected twitter data using R related to android Lollipop (Latest android version having mixed reviews). Reason for selecting this data is we can have both positive and negative reviews of it so that training dataset we form will train model very efficiently. I trained model by providing labelled tweets from CSV file. The built model has accuracy of 74% approximately. The accuracy can be increased by training the model with larger size of training dataset.

Following is the procedure followed to build model and train it:

1. Created project as TweeterSentiments on google cloud platform.
2. Then in this project built one model as tweetentmodel which is using google prediction API.
3. Data collection and preprocessing:
   - The data is collected from twitter using twittR package in R. It will take tag about which we want to retrieve the data. Provided androidlollipop as a tag to collect the tweets. **100** tweets collected and removing duplicates I finalized training dataset with **78** tweets.
   - In preprocessing I used R with command gsub which removes the unnecessary part from your data as you provide it. It takes parameter from user and removed that part from dataset. Usually majority twitter tweets contains url which is of no use for analysis. So such things can be removed in using gsub command. (All commands are mentioned in appendix below).
   - After preprocessing I manually labelled the data putting label in the first column of CSV file and tweet in second column. Next thing is to upload the data to GCS.
4. Then created one bucket as twitstore in which stored one CSV file containing training data collected and preprocessed earlier.
5. Training the model:
   - For training *prediction.trainedmodels.insert()* method is used. In this method we pass the training data file stored on Google Storage.
   - The same thing can also be done using R. In R we need to call the method and install package googlepredictionapi. The model will get trained after successful implementation. (code is provided in appendix)
6. In next step we can pass the tweets to find its sentiments. The trained model will take the tweet and predict its sentiment value out of 1 as positive and negative. And classify it into category which has higher value.

7. We can also pass the tweet from R and find its sentiments. Also when we need to update the training set then we can upload the updated training set and train the model again using that training data.

Before implementing googleapi functions in R we have to install gsutil in our system.

Comments about the predictions:

- The prediction made by the model is good since it calculates polarity of sentence on both categories and select the better one.
- Also accuracy is better as it is trained over significant training dataset.
- It can be increased if we increase the size of training dataset.
- If any tweet has both the meanings then we need to put it in training set twice and labelled with both categories so classifier can see it with both sides equally and classify as neutral one.

Appendix:

1. R script to retrieve and preprocess twitter tweets:

install.packages(twitteR)

install.packages(wordcloud)

install.packages(RColorBrewer)

install.packages(plyr)

install.packages(ggplot2)

install.packages(sentiment)

library(twitteR)

library(wordcloud)

library(RColorBrewer)

library(plyr)

library(ggplot2)

library(sentiment)

install.packages(httr)

library(httr)

oauth_endpoints("twitter")

created twitter application using twitter account

recorded api key, secret, access token and secret on authenticating application and used them below

api_key <- "API key recorded from twitter"

api_secret <- "Secret key recorded from twitter"

access_token <- "Access Token recorded from twitter"

access_token_secret <- "Access Token Secret key recorded from twitter"

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)

traindata = searchTwitter("androidlollipop", n=100, lang="en")

##retrieved 100 tweets from twitter repated to android lollipop

training_data = sapply(traindata, function(x) x$getText())

training_data = gsub("(RT|via)((?:\\b\\W@\\w+)+)", "", training_data)*

##all retweeted entities are removed

training_data = gsub("@\\w+", "", training_data)

all people names removed

training_data = gsub("http\\w+", "", training_data) ## all links removed

2. R script to train the model and predict the sentiments by passing tweets:

install.packages("RCurl")

install.packages("rjson")

install.packages("googlepredictionapi_0.1.tar.gz", type="source")

library(RCurl)

library(rjson)

library(googlepredictionapi)

##all required packages are installed.

Now we have to train the model build on Google cloud platform.

tweetsentmodel <- PredictionApiTrain(data="gs://twitstore/training_data.csv")

once model is trained we need to make predictions. By passing tweets.

To do so we can use method predict and pass model name and tweet to be checked

predict(tweetsentmodel, "new android lollipop is having best features among all versions.")

##this will pass the tweet to model and get its sentiments.

Github Link: ***https://github.com/sourabh99/Google-Prediction-Api***