OBJECTIVE: CLASSIFICATION OF PATENT ON THE BASIS OF TOPIC OF PATENT

Problem Statement:

I was given a dataset of around 49345 bank reviews and had to classify them in three categories 'Credit Card', 'Insurance' and 'Loan'.

1>Given a set of bank reviews filed on different topics, train a classifier or use a tool which when provided with an user input of a review can provide a judgement as to what topic the review is on as well as the probabilities that the review is talking on a set of topics using rasa nlu.

2>Do the named entity recognition (NER) using spacy on the reviews to extract information like NAME, BANK ACC NUMBER, CREDIT CARD NUMBER, DATE and PLACE from the reviews.

3>Do the sentiment analysis of the review with 1 as positive and 0 as negative, considering any review which has even a slight negative comment as negative using random forest classifier.

Link to github repo: https://github.com/shaliniiit/SkybitsBankReview

Requisites:

1.Rasa_nlu

2.spacy

3. spacy download en

4.Jupyter Notebook

5.Excel

6.Sklearn

7.NLTK for stopwords

Steps:

0.Open cmd.

1.Mkdir Desktop\BankBot

2.cd BankBot

3.Run npm i -g rasa-nlu-trainer in a second cmd window which has been opened in Administrator mode.

4.Run python -m spacy download en in a second cmd window which has been opened in Administrator mode.

5.mkdir data in first cmd window

6.cd data

7.echo ‘’->data.json and the open data.json file under data.Copy paste the contents of the data file in git repository.(Cloning the git repository can save the time of copying all files actually).

8.rasa-nlu-trainer in first cmd window.

9.Enter the training data in the server window which pops up by clicking on Add Example at the top and then click on Save.

10.Open another third cmd window while the server runs on this window.

11.Create configure_spacy file and copy paste code from Git Repository.

12.Copy paste requirements.txt from Git Repository.

13.Copy paste nlu_model.py file from Git Repository.

14.Run pip install -r requirements.txt in same cmd window.

15.Run nlu_model.py to create a folder called models in BankBot which will contain all pipeline related files.

16.Enter input in nlu_model.py file in line 14 Interpreter.parse field.

17.Run nlu_model.py again to see the results.

18.Enter every new test case in the server which is running.

19.Run train_nlu as well as run_nlu every time so that the model gets trained with every new test case.

20.Copy results to another file if needed.

21.Enter review in line number 5 in doc=nlp(“”) field of entity.py file after copy pasting entity.py file from the Github repo.

22.Run the entity.py file in cmd to get the list of entities, nouns and verbs.

23.Get the f.csv file from the git repository and store it in Downloads section of my computer.

24.Change all file paths in sentiment.py code to respective username on user’s computer.(line 2, 35 and 59 from C:\Users\Shalini\Downloads\<filename> to C:\Users\<user computer name> \Downloads\<filename> for Windows and likewise for Ubuntu.

25.Copy paste the sentiment.py file from the git repository to a Jupyter Notebook. Enter all test cases into an excel file in a column with column header ‘review’ and run sentiment.py. 0 means negative and 1 means positive.Any review with even a single negative comment has been tagged as negative. It would be a good idea to enter new test cases in the f.csv training

data file as the classifier has been trained on a limited list of sentiment based reviews, most Indian bank reviews are mixed and it is always a good idea to train more data.

26.The results of sentiment analysis can be seen by opening the lll1.csv file which gets created in Downloads section every time sentiment.py is run on Jupyter Notebook.Do not forget to delete the file from Computer everytime after checking results as otherwise it will provide permission rights problem.

27.The entities can also be visualised by copying the result from to in cmd to an html file and running it on localhost by double clicking.

Suggestions:

1.Increasing the number of training examples. By adding examples to f.csv and rasa-nlu-trainer server.

2.Running both train_nlu and run_nlu each time. It takes more time but trains the model every time.

3.Adding every test case encountered to the training data in the server.

4.Add every sentiment test case to f.csv file.

Useful Links:

1. https://rasa.com/docs/rasa/
2. https://forum.rasa.com/?_ga=2.219041177.1002252592.1562763324-389945078.1558386011
3. https://github.com/shaliniiit/SkybitsBankReview