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

Problem Statement:

Given a set of patents filed on different topics train a classifier or use a tool which when provided with an user input of a patent can provide a judgement as to what topic the patent is on as well as the probabilities that the patent is talking on a set of topics.

I collected four sets of patent titles 'Cybersecurity', 'Data Mining', 'Solar Energy' and 'Graphene' from Google Patents. I then used an open source tool called RasaNLU to classify the patents based on the patent title.

Link to github repo: https://github.com/shaliniiit/Kreatrade-NLP-Patent-Corpus

Steps:

- 0.Open cmd.
- 1.Mkdir Desktop\PatentBot
- 2.cd PatentBot
- 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 config 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 PatentBot 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.

Accuracy:

92.8%

18 errors in 250 test cases with 123 training examples. Optimum would be around 450-500 training data cases. It gives almost no error.

Suggestions:

- 1.Increasing the number of training examples.
- 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.

Useful Links:

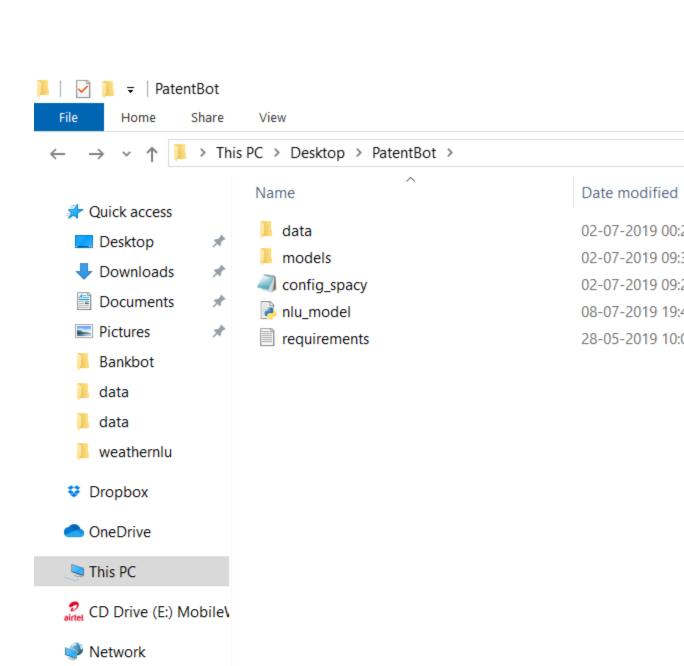
- 1. https://rasa.com/docs/rasa/
- 2. https://forum.rasa.com/?ga=2.219041177.1002252592.1562763324-389945078.1558386011
- 3. https://github.com/shaliniiit/Kreatrade-NLP-Patent-Corpus

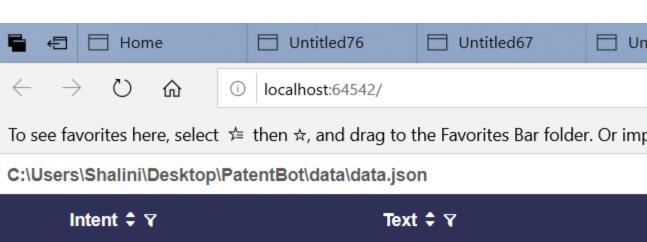
Images:

C:\Users\Shalini\Desktop\PatentBot\data\data.json (react) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

```
data.json --- PatentBot\data
FOLDERS
▼ i react
                            "rasa nlu data": {
 todoapp
                              "regex_features": [
                                  "name": "Solar Energy",
                                  "pattern": "solar.*"
                                },
                                {
                                  "name": "Cybersecurity",
                                  "pattern": "cybersecurity.*"
                    10
                                },
                    11
                                {
                    12
                                  "name": "Graphene",
                    13
                                  "pattern": "graphene.*"
                    14
                                },
                    15
                                {
                    16
                                  "name": "Data Mining",
                    17
                                  "pattern": "data mining.*"
                    18
                                },
                    19
                                {
                    20
                                  "name": "Solar Energy",
                    21
                                  "pattern": "Solar.*"
                    22
                                },
                    23
                                {
                    24
                                  "name": "Cybersecurity",
                    25
                                  "pattern": "Cybersecurity.*"
                    26
                                },
                    27
                                {
                    28
                                  "name": "Graphene",
                    29
                                  "pattern": "Graphene.*"
                    30
                                },
                    31
                    32
                                  "name": "Data Mining",
                    33
                                  "pattern": "Data mining.*"
                    34
                    35
                    36
                              ],
"common_examples": [
                    37
                    38
                    39
                                   "text": "Booster unit, power o
```





C:\Users\Shalini\Desktop\PatentBot\data\data.json						
	Intent ≎ ♥	Text ‡ ▼				
+	Solar Energy	Booster unit, power conditioners, and				
+	Solar Energy	Direct beam solar lighting system				
+	Solar Energy	Demounting of inverted metamorphic r				
+	Solar Energy	Thin-film solar reflectors deployable from				
+	Solar Energy	Solar panel				
+	Solar Energy	Reactive power compensation in solar				
+	Graphene	3-dimensional graphene structure and				
+	Graphene	Resin plating method using graphene				
+	Graphene	Preparation method for three-dimension				
+	Graphene	Doped graphene electronic material				
+	Graphene	A method graphene reinforced aluminu				
+	Graphene	Method for manufacturing graphene-co				

```
r = requests.get(url)
 File "C:\Users\Shalini\AppData\Local\Programs\Python\F
   return request('get', url, params=params, **kwargs)
 File "C:\Users\Shalini\AppData\Local\Programs\Python\PC:\Users\Shalini
   return session.request(method=method, url=url, **kwa
 File "C:\Users\Shalini\AppData\Local\Programs\Python\PC:\Users\Shalini
   resp = self.send(prep, **send_kwargs)
                                                      searching for the
 File "C:\Users\Shalini\AppData\Local\Programs\Python\Pfound C:\Users\SI
   r = adapter.send(request, **kwargs)
                                                      server listening
 File "C:\Users\Shalini\AppData\Local\Programs\Python\P^CTerminate batcl
   raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPSConnectionPoolc:\Users\Shalini
('<urllib3.connection.VerifiedHTTPSConnection object at
                                                      C:\Users\Shalini
C:\WINDOWS\system32>python -m spacy download en
Requirement already satisfied: en_core_web_sm==2.1.0 frcc:\Users\Shalini
i\appdata\local\programs\python\python37\lib\site-packagC:\Users\Shalini
Download and installation successful
                                                      ner\server.js
You can now load the model via spacy.load('en_core_web_snpm WARN slick-ca
symbolic link created for C:\Users\Shalini\AppData\Locals yourself.
Linking successful
                                                      npm WARN optional
-->
                                                      "} (current: {"o:
C:\Users\Shalini\AppData\Local\Programs\Python\Python37\
You can now load the model via spacy.load('en')
                                                      + rasa-nlu-train
                                                      updated 2 package
C:\WINDOWS\system32>npm i -g rasa-nlu-trainer
npm WARN registry Using stale data from https://registryC:\Users\Shalini
during revalidation.
                                                      searching for the
C:\Users\Shalini\AppData\Roaming\npm\rasa-nlu-trainer ->found C:\Users\SI
npm WARN slick-carousel@1.8.1 requires a peer of jquery@server listening
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for f
+ rasa-nlu-trainer@0.2.7
updated 2 packages in 156.159s
```

C:\WINDOWS\system32>python -m spacy download en