Documentation

Ruturaj Mohanty

July 25 2020

1 Preliminaries

Please ensure that you have an updated UBUNTU version 18+ installed as your operating system with python3 installed. For training and running the classifier models gpu is preferred but it can also be run on cpu with some limitation on the execution speed. For training the language model you would need at least 1 v100 gpu with minimum memory capacity of 32GB.

2 Overview

The software has 5 sections :-

- Collected Raw News Articles
- Final Project Software
- Supervised Classifier Model Training
- Unsupervised Model Training
- Documentation
- brat-v1.3CrunchyFrog

Next,we would be going each of these sections one after the other.

3 Collected Raw News Articles

Here you can find all the collected raw news articles which consists of around 0.5 million Indian media articles and 33,775 Kashmir specific articles.

```
Operating Guidelines:-

- Make sure you have python3 and pip3 installed installed.
- Install ijson using the command :- pip3 install ijson
```

```
-------------------------------------------------------------------------------------

- Commands to extract the text files:-

python3 pythonFileName fileToParse outputFolderName numberOfArticlesToGenerate

- maximum number of articles in news_kashmir_dump.json = 33775

- maximum number of articles in news_India_dump.json = 0.5million.

- So your numberOfArticlesToGenerate range should be within the range.

--------------------------------------------------------------------------------------

- Example :-
- Kashmir Articles:-
python3 parsing_json.py news_kashmir_dump.json Kashmir 100

- Indian Articles:-
python3 parsing_json.py news_India_dump.json India 10

---------------------------------------------------------------------------------------

- To generate all the files use the following commands:-
- Kashmir Articles:-
python3 parsing_json.py news_kashmir_dump.json Kashmir -1


- Indian Articles:-
python3 parsing_json.py news_India_dump.json India -1
```

Note :- Don't try to open the folder by double clicking the icon after extracting 0.5 million articles inside that folder. It will crash your system.

4 Final Project Software

Here you can launch the software into the web. Chrome browser is preferable and also was tested on the same. I have provided 3 different models. The default one is the best of all these models. You can run any one of them by using the commands in the terminal:-

```
- if you wish to run the software using NBSVM model:-
bash run.sh 1

- for BERT:-
```

```
bash run.sh 2

- for MLSTM:-
bash run.sh 3
```

Incase you face issues running bash script you can also run them manually using the commands:-

```
source env/bin/activate
pip3 install -r requirements.txt
python3 -m spacy download en_core_web_sm
python3 app.py 1 (for NBSVM)
python3 app.py 2 (for BERT)
python3 app.py 3 (for MLSTM)
python3 app.py 4 (for Default Original)
```

Once you successfully run the commands, a link will pop up in the screen, something like this:- http://127.0.0.1:5000/. Open this link on your chrome browser to view the software.

5 Supervised Classifier Model Training

Here you can train your classifier on top of the extracted features from language model. I have provided 3 language models namely, NBSVM, MLSTM and BERT. You can train the softmax classifer on top of it using the following commands:-

```
- To train using NBSVM:-

bash run.sh 1

- To train using BERT:-

bash run.sh 2

- To train using MLSTM:-

bash run.sh 3


---------------------------------------------------------

- You can also train manually using the below commands:-

- To train the classifier use the commands:-
```

```
python3 runDistilBert.py

or

python3 runMLSTM.py

or

python3 runNBSVM.py
```

Once the classifier model is trained it is stored inside 'Prediction Classifiers'. Transfer this model from there to 'Software/FinalProjectSoftware/models' to be able to successfully test it live on our web browser. If you wish to add more training data then you can append more labelled chunks to the file **Software/SupervisedClassifierModelTraining/LabelledChunks/trainV2.csv** Also you can provide any other csv file, just make sure that the name of the csv file is **trainV2.csv**

6 Unsupervised Model Training

Here you can train our default mLSTM language model using more news articles. Make sure to train it on at least 1 V100 gpu having memory capacity of 32GB. It was trained on IITD HPC. The 'extractFromZip.py' file opens the zip folder consisting of all the raw news articles that we extracted in the section 'Collected Raw News Articles' above. You can skip it if you haven't zipped the folder. Next 'generateTrainingData.py' combines all the news articles by appending one after the other and builds a single file. This file will be huge in size. **Don't try to open it by double clicking else you system might crash**. You can divide this huge file into smaller files which will help you easily fit in memory using the command `split -b 13389585 train.txt`. Here 13389585 is the size of the smaller size i want. You can change it as per your gpu memory constraints. Now you can check the file 'pbsbatch.sh' to see all the commands required for training. You can check this link https://github.com/jonny-d/openai_reproduction to find more guidelines on the language model training. Additionally i have also provided the 'condaRequirement.txt' file using which you can directly setup the conda environment that was used to train the model.

7 Documentation

Here you can find all the required documentation files. Aditionally you can open the readme files inside each folder or look at the comments done inside code for better understanding.

8 brat-v1.3CrunchyFrog

Here you can find all the files required for labelling the chunks. Although i would ask to download the latest version from brat website and do the setup. For more detailed instructions on the same please go through the 'InstallationGuideToLabellingTask.pdf' documentation file.