**Parts of Speech Tagging using BERT**

**PROBLEM STAEMENT**

In this project we will be performing one of the most famous task in the field of nautal language processing i,e Parts of Speech Tagging using BERT.

**DESCRIPTION OVERVIEW**

Part-Of-Speech tagging (POS tagging) is also called grammatical tagging or word-category disambiguation. It is the corpus linguistics of corpus Text data processing techniques for marking meaning and context.

Part-of-speech tagging can be done manually or by a specific algorithm. Using machine learning methods to implement part-of-speech tagging is the research content of Natural Language Processing (NLP). Common part-of-speech tagging algorithms include Hidden Markov Model (HMM), Conditional Random Fields (CRFs), and so on.

Part-of-speech tagging is mainly used in the field of text mining and NLP. It is a preprocessing step for various types of text-based machine learning tasks, such as semantic analysis and coreference resolution.

1. CC Coordinating conjunction

2. CD Cardinal number

3. DT Determiner

4. EX Existential there

5. FW Foreign word

6. IN Preposition or subordinating conjunction

7. JJ Adjective

8. JJR Adjective, comparative

9. JJS Adjective, superlative

10. LS List item marker

11. MD Modal

12. NN Noun, singular or mass

13. NNS Noun, plural

14. NNP Proper noun, singular

15. NNPS Proper noun, plural

16. PDT Predeterminer

17. POS Possessive ending

18. PRP Personal pronoun

19. PRP$ Possessive pronoun

20. RB Adverb

21. RBR Adverb, comparative

22. RBS Adverb, superlative

23. RP Particle

24. SYM Symbol

25. TO to

26. UH Interjection

27. VB Verb, base form

28. VBD Verb, past tense

29. VBG Verb, gerund or present participle

30. VBN Verb, past participle

31. VBP Verb, non-3rd person singular present

32. VBZ Verb, 3rd person singular present

33. WDT Wh-determiner

34. WP Wh-pronoun

35. WP$ Possessive wh-pronoun

36. WRB Wh-adverb

**TECHNOLOGY USE**

Here we will be using **Anaconda Python 3.6 , Pytorch 1.4 with GPU support CUDA 10 with CuDNN 10.**

**INSTALLATION**

Installation of this project is pretty easy. Please do follow the following steps to create a virtual environment and then install the necessary packages in the following environment.

**In Pycharm it’s easy**

1. Create a new project.

2. Navigate to the directory of the project

3. Select the option to create a new new virtual environment using conda with python3.6

4. Finally create the project using used resources.

5. After the project has been created, install the necessary packages from requirements.txt file using the command pip install -r requirements.txt

**In Conda also it’s easy**

1. Create a new virtual environment using the command

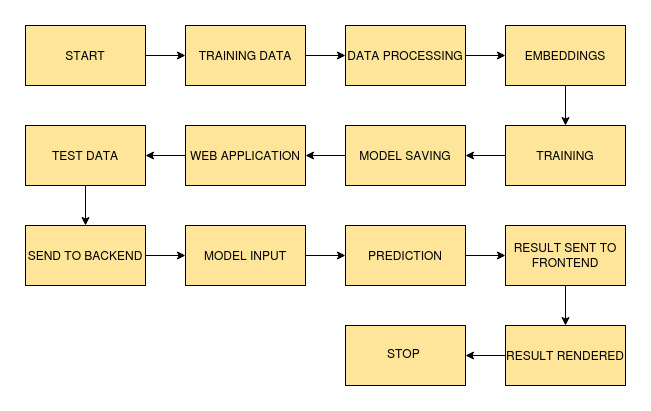
conda create -n your\_env\_name python=3.6

2. Navigate to the project directory.

3. Install the necessary packages from requirements.txt file using the command

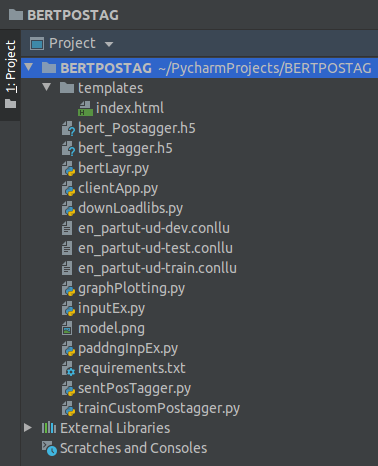
pip install -r requirements.txt

**WORKFLOW DIAGRAM**

****

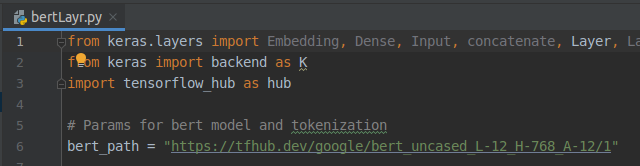
**IMPLEMENTATION**

**1. Project Directory**

****

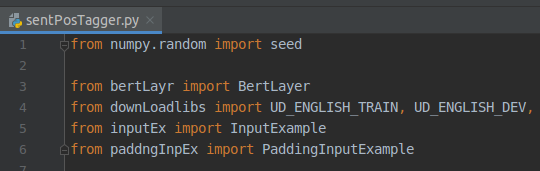
This above picture shows the folder structure of the project. Here project folder consists of data and BERT models.

**2. bertlayr.py**

****

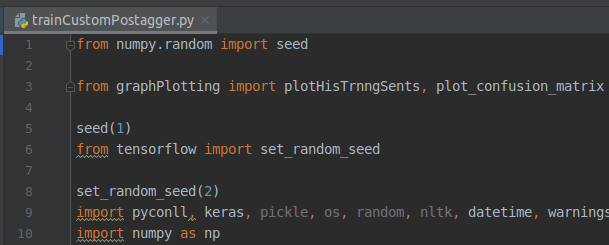
This file consists of the the bert model architecture which will be used to train the data.

**3. sentPosTagger.py**

****

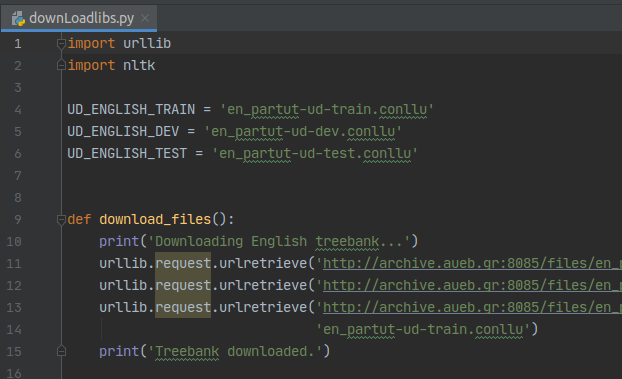
This file is used to train the model and to do the prediction.

**4. trainCustomPostagger.py**

****

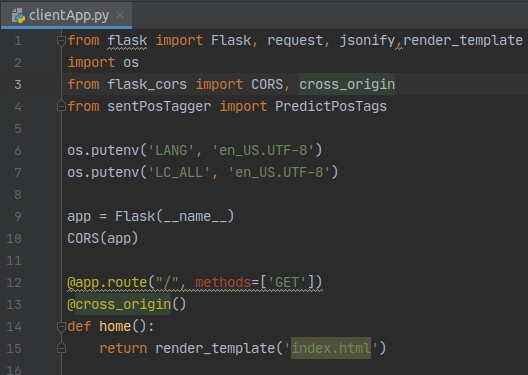
This file is used to train a custom pos tagging model if the user wants to train.

**5. downLoadlibs.py**



This file is used to download dataset.

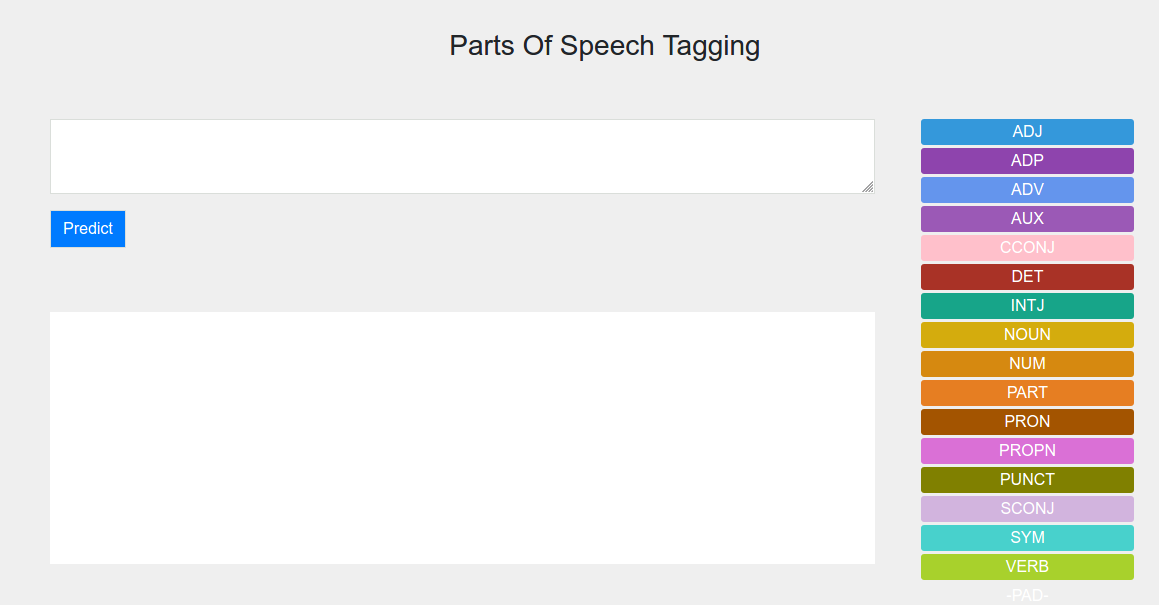
**6. ClientApp.py**



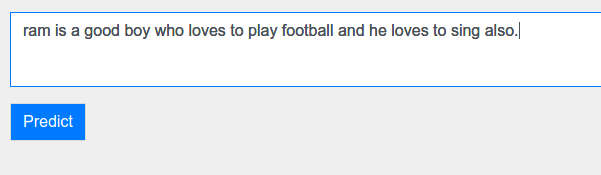
This is tha flask server file.

**TESTING IN LOCAL/API**

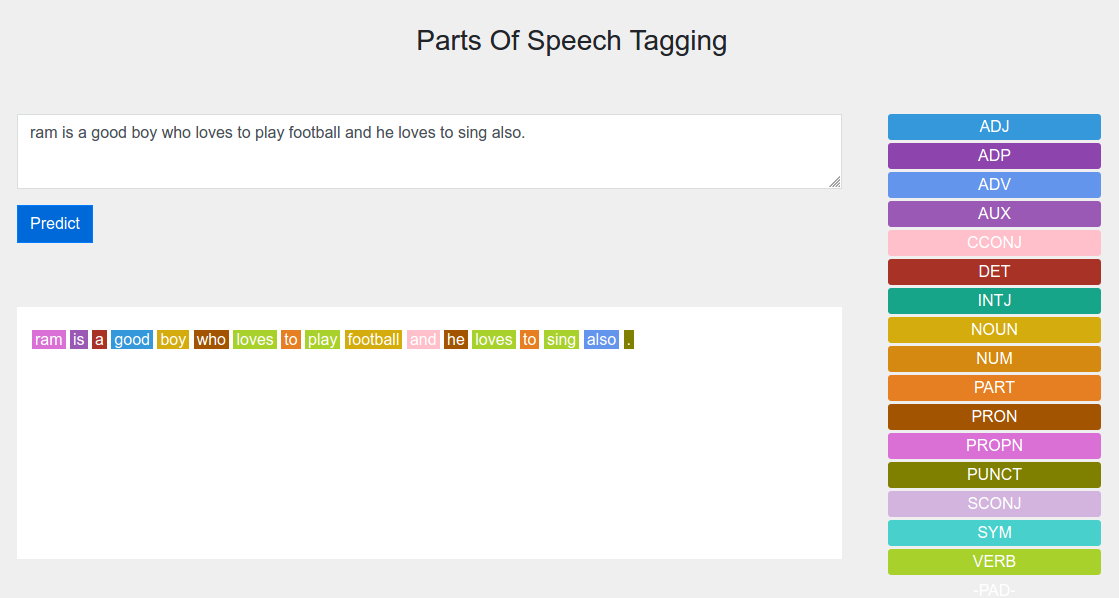
To do the test testing we need to run the clientApp.pyand after that web server will start at <http://0.0.0.0:5000/>



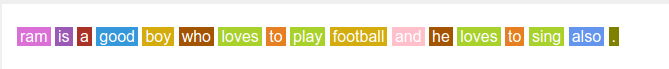
Enter the sentence and click on predict button.



After clicking predict



Results are shown below.



Do the matching of the words with corresponding colours.

**CONCLUSION**

Here we successfully performed Parts of Speech tagging on the given dataset.

**COMPARISION**

More data or better larger dataset can be used to build a better model. We can also try out better pre trained model with fine tuning to increase the performance.