**Semantic Text Similarity**

**PROBLEM STAEMENT**

In this project we need to fing the similarity for given pair of sentences and calculate the distance between them.

**DESCRIPTION OVERVIEW**

Semantic textual similarity deals with determining how similar two pieces of texts are. This can take the form of assigning a score from 1 to 5. Related tasks are paraphrase or duplicate identification.

Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical representation (e.g. their string format). These are mathematical tools used to estimate the strength of the semantic relationship between units of language, concepts or instances, through a numerical description obtained according to the comparison of information supporting their meaning or describing their nature. The term semantic similarity is often confused with semantic relatedness. Semantic relatedness includes any relation between two terms, while semantic similarity only includes "is a" relations.

For example, "car" is similar to "bus", but is also related to "road" and "driving".

**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

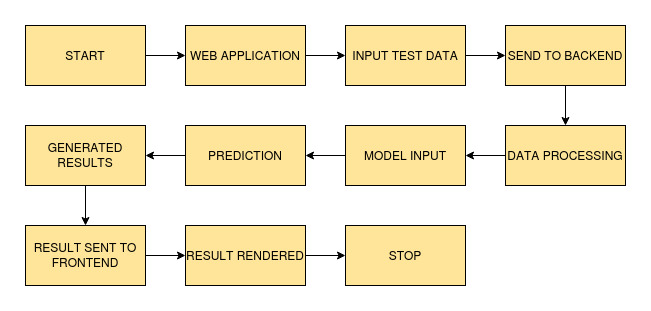
conda create -n your\_env\_name python=3.6

2. Navigate to the project directory.

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

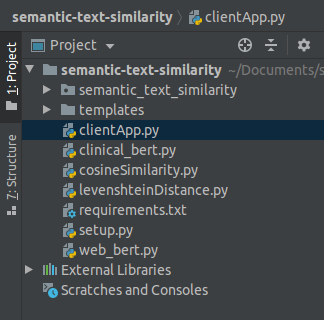
pip install -r requirements.txt

**WORKFLOW DIAGRAM**



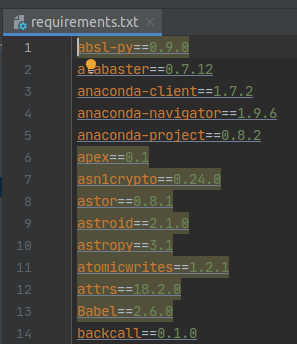
**IMPLEMENTATION**

**1. Project Directory**



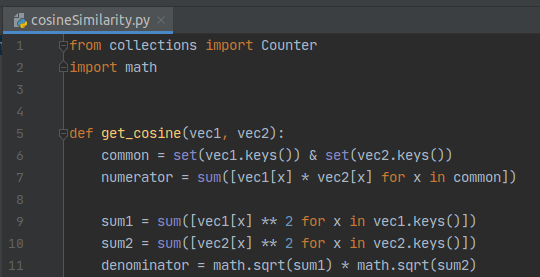
This above picture shows the folder structure of the project.

**2. requirements.txt**



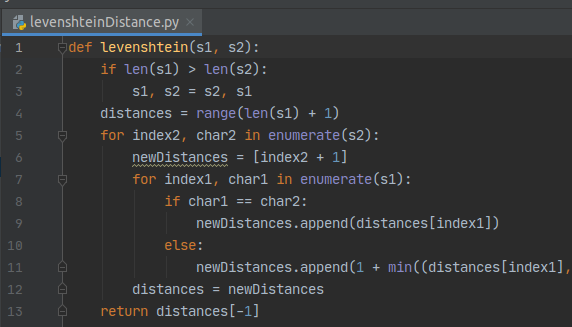
This file contains all the necessary packages to be installed for this project.

**3. cosineSimilarity.py**



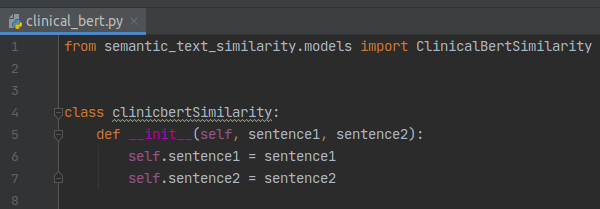
This file calculates the the cosine similarity distance between the sentence pairs.

**4. levenshteinDistance.py**



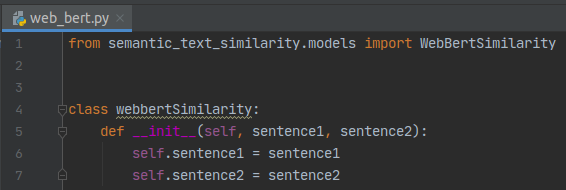
This file calculates the the levenshtein distance between the sentence pairs.

**5. clinical\_bert.py**



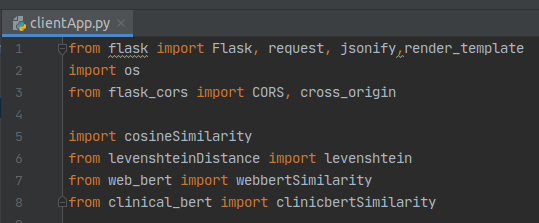
This file calculates the the semantic text similarity between the sentence pairs using the clinic bert model.

**6. web\_bert.py**



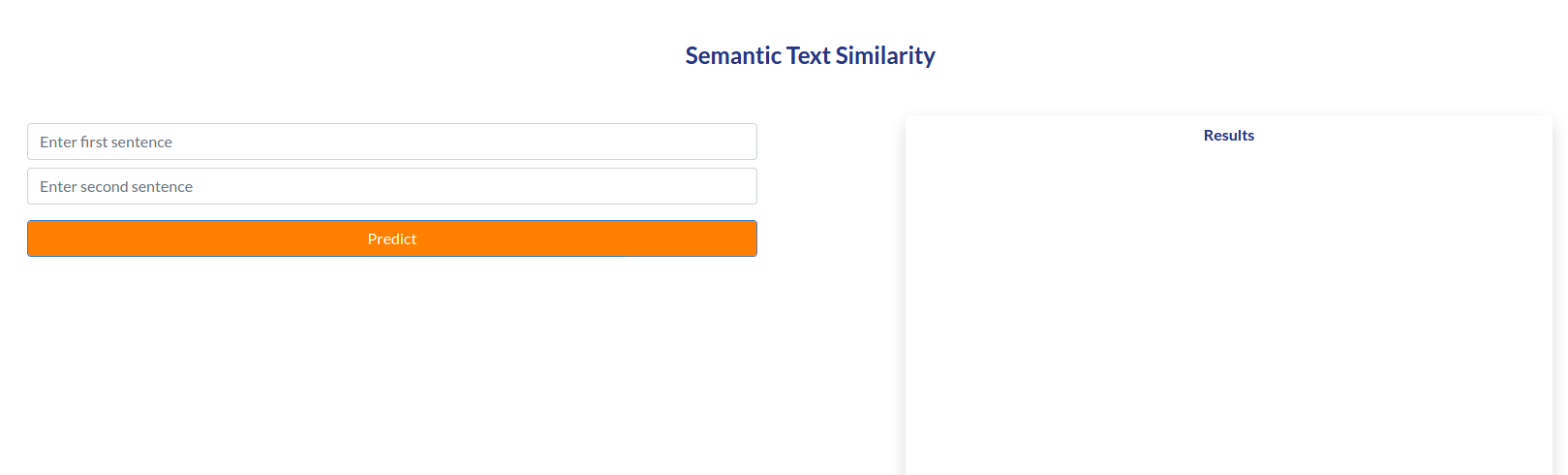
This file calculates the the semantic text similarity between the sentence pairs using the web bert model.

**7. clientApp.py**

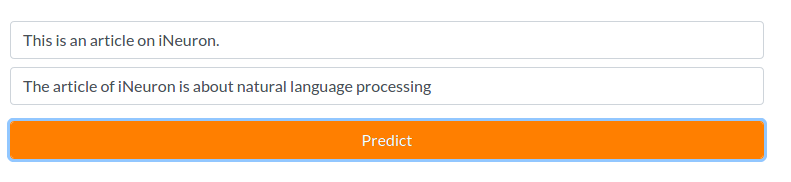


This is the flask server file and entry point of the application.

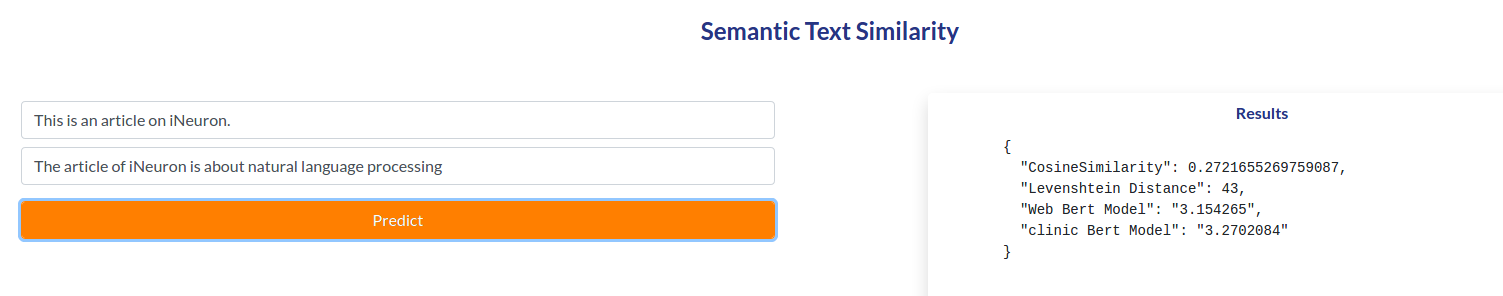
**TESTING IN LOCAL/API**



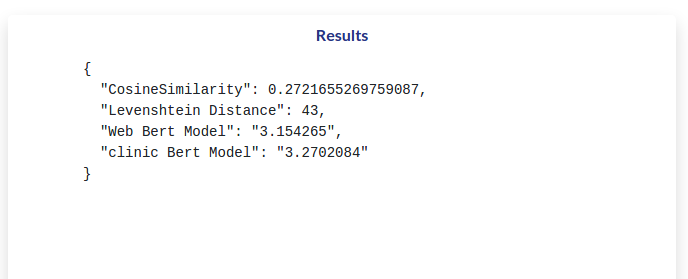
Enter the sentences and click on Predict



After giving input & clicking on Predict.



Results are shown.



Results are shown for diffrent metrics and STS tasks.

**CONCLUSION**

Here we have successfully completed the task of creating of calculating semantic similarity between the sentences.

**COMPARISION**

Here we can go for better pre trained or we can increase dataset size.