**Question Answering**

**PROBLEM STATEMENT**

The Question answering is an important NLP task and this one is a milestone for artificial intelligence systems. In this Project we will deal with deep learning + NLP approaches to build a Question answer.

**DESCRIPTION OVERVIEW**

Question Answering, It’s a computer science discipline which in the fields of information retrieval and natural language processing(NLP), which generally focuses on building that systems automatically answer questions just like a human in a natural language. Come on the topic of computer understanding of our natural language which consist of the capability of the program system to translate sentences into internal representation so that the system generates some valid answers to a question asked by the user. Valid answer in the sense answers relevant to the questions asked by the user.

**There are types of Question Answering**

Major are three modern paradigms of question answering**:**

**i). IR-based Factoid –** It’s goal is to answer a user’s question by just finding the short text segments on the Web or some other collection of documents. During question-processing phase there are number of pieces of information are extracted from the question. The answer type specifies some kind of entity the answer which consist of(location, person, time,etc.)

**ii). Knowledge-based** – The idea for answering a natural language question is by mapping it to the query over a structured database. Systems for mapping from the text string to any logical form are known as semantic parsers. Semantic parsers will do like map either to some version of base calculus or the query language like SPARSQL or SQL

**iii). From multiple information sources: IBM’s Watson [5,6]** – This system won the Jeopardy! Challenge in 2011 which set an example of a system that relies on the wide variety of resourses to answer questions.

**TECHNOLOGY USE**

Here we will be using Anaconda Python 3.6 , Keras 2.3.1 using TensorFlow 1.14.0 backend.

**INSTALLATION**

Installation of this project is quite simple. 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

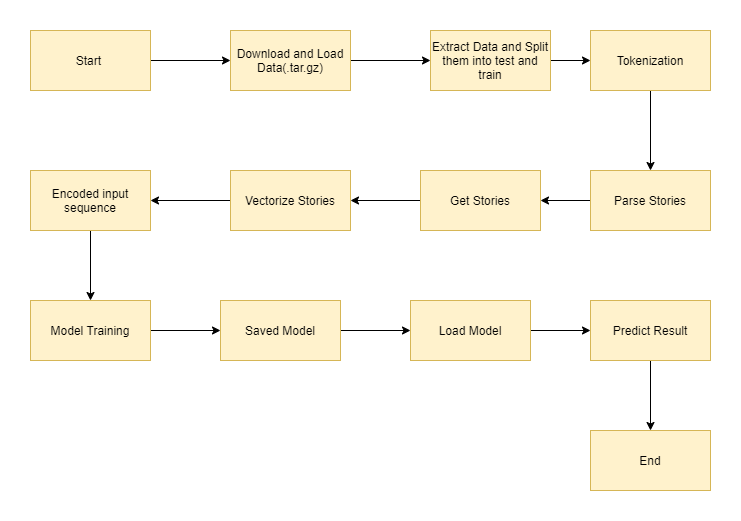
conda create -n your\_env\_name python=3.6

2. Navigate to the project directory.

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

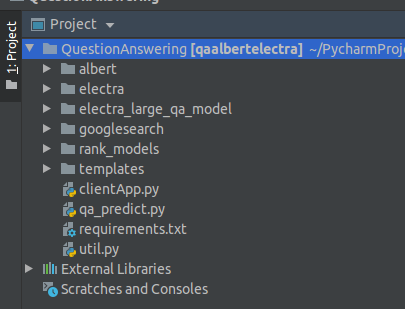
pip install -r requirements.txt

**WORKFLOW DIAGRAM**



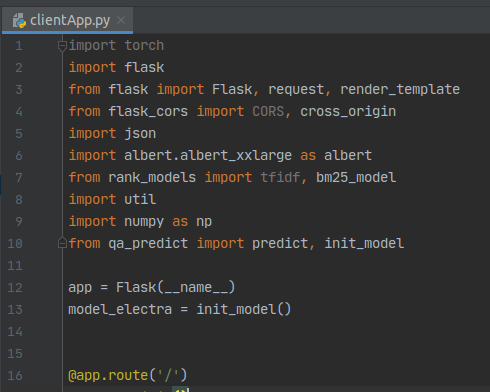
**IMPLEMENTATION**

1. **Project directory**

****

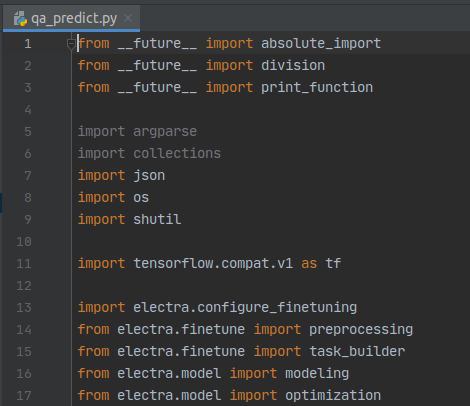
In the above picture represents the project directory if we open the project folder using PyCharm. In this directory different types of files are there like for clientApp, utils, qa\_predict etc. Few folder also there for storing data into this folder.

1. **clientApp.py**

****

In this file we can check our results by passing our strings and get results as an output.

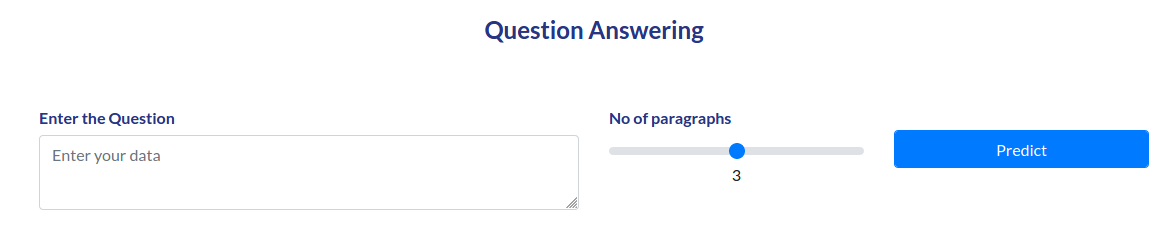
**4.qa\_predict.py**



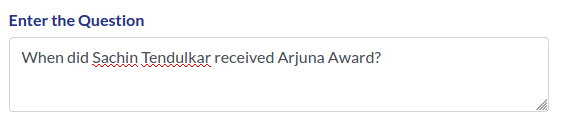
In this file we would arrange functions to get the proper result by applying clientApp.py

**TESTING IN LOCAL/API**

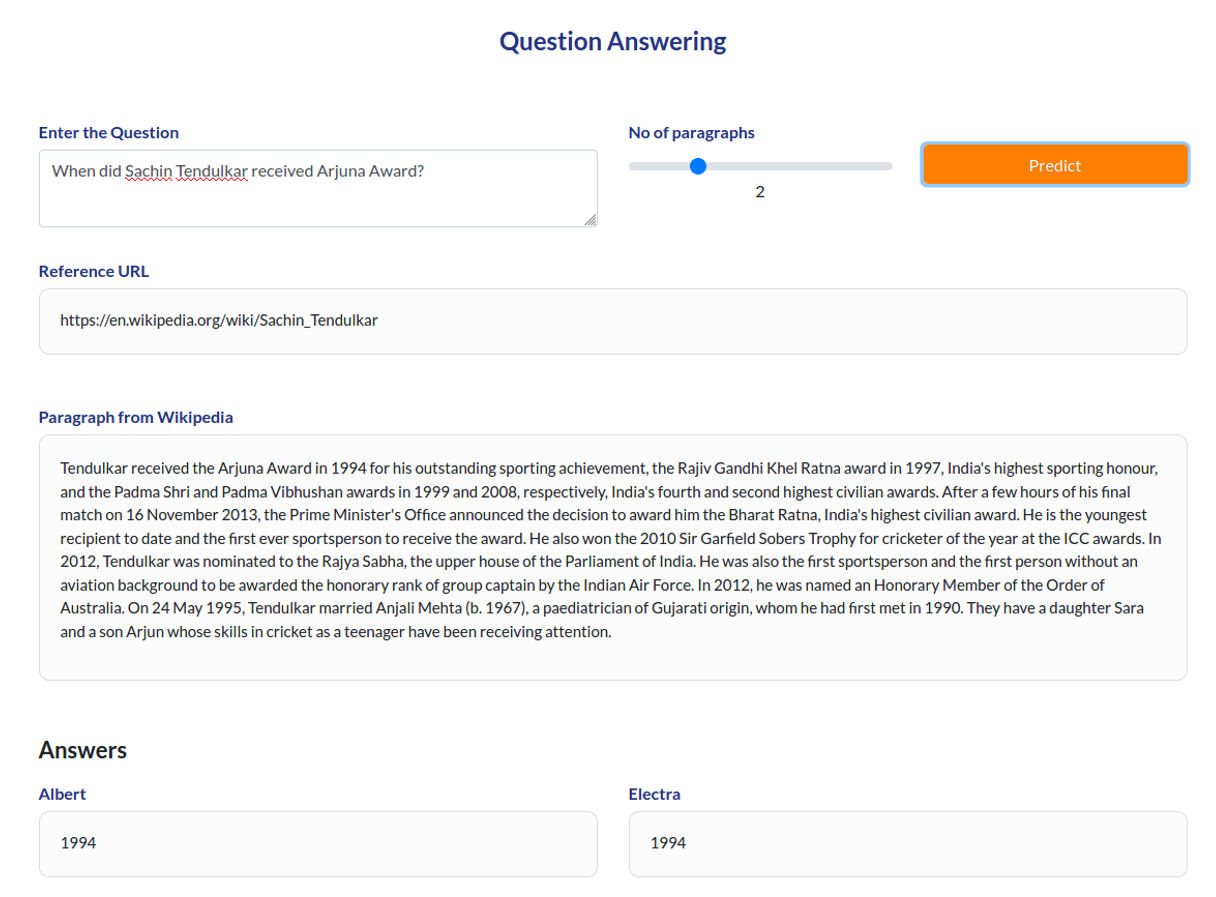
To do the test testing we need to run the clientApp.pyand after that web server will start at <http://0.0.0.0:5000/>

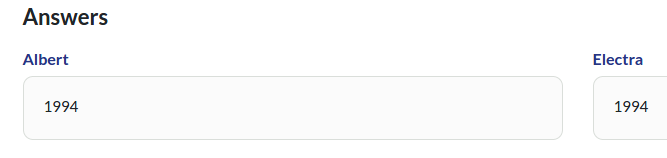


Enter the question and keep no of paragraphs = 2 for now, Click on Predict button.



After clicking Predict





Finally results are shown for the both of the models corresponding to their boxes.

**CONCLUSION**

Here we successfully built a Question answering model which we can use it to answer based on our data available.

**COMPARISION**

We can create the same Question Answer model with various other pre trained models available from Hugging Face transformers.