Applications of AI Assignment

INSTRUCTIONS:

1. This is a **group** assignment which consists of 3 or 4 members.
2. All the relevant assignment instructions are given below.
3. Do NOT submit **.zip** files. The assignment will not be evaluated otherwise.
4. Do NOT submit **.ipynb i.e., Jupyter NB** files.
5. Any late submission will attract a **penalty** as mentioned in the course outline.
6. The honor code for this Assignment is **0N.**

DELIVERABLES: There will be deliverables; **PLEASE READ THE LIST CAREFULLY**:

a. CSV File: Submit your knowledge base (a clean set of FAQ) as a database as a **CSV file**.
b. .py file: In case your work is **automated**, submit the codebase in **.py** files. In case, your work is **NOT automated**, please submit just the **CSV** file. (Submit: ***FAQ_codebase_n.py; n stands for file number***)

There are two chatbots to build:

While building **a simple similarity-based FAQ bot using python--**

c. .py file: Implement a simple command line interface to query the program and get a response. While evaluating, if asked a question, it should give a response. The evaluation will be based on the bot's response. (Submit: ***python_run.py***)

While building a **Dialogflow based FAQ bot--**

d. DialogFlow URL: Deploy and expose your bot as a **URL**. While evaluating, if asked a question, it should give a response. The evaluation will be based on the bot's response.

e. A report (.pdf format): **1.** Explain your data collection, building process and use **code screenshots and output screenshots (bot output & in case, automated knowledge base is built, it's output too).** **OUTPUT SCREENSHOTS ARE A MUST.**
2. Paste your Dialog Flow based FAQ bot URL in the starting of the report.
3. Draw and submit **an ideal conversational flowchart** for your bot (An example is given below)
4. Discuss **ways to improve your bot** (better data, algorithms, hybrid approaches, etc.) before you could host this for public use at production scale. (Submit: ***chatbot_report.pdf***)

QUESTION:

Task: Build a Corona Virus Information Bot

Description: Build a very comprehensive COVID-19 information chatbot that provides answers to questions around latest statistics, infected countries, medical information, prevention, symptoms etc.

Below are some sources of information: -

- https://www.cdc.gov/coronavirus/2019-ncov/faq.html

- https://www.who.int/news-room/q-a-detail/q-a-coronaviruses
- https://www.worldometers.info/coronavirus/
- https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset

For instance, a question such as "What is corona virus?" will return an answer from the FAQ corresponding to it. Similarly, a question "How many countries are affected?" will return a number as the answer, provided from a predefined dataset of questions and answers.

I. Data Collection: (30 points)

- As a first step you will need to create a clean set of FAQ for your chatbot by combining all the questions from all sources about Covid-19.
- Build an NLP pipeline to automate the question scraping, cleaning from the above websites and converting into question, answer pairs.
- In cases where it is difficult to automate the entire pipeline, resort to manual scraping and combining with NLP techniques (e.g. copy and paste into a CSV file and start the cleanup process)

Evaluation:

- Submit your knowledge base as a database as a **CSV** file.
- In case your work is **automated**, submit the codebase in **.py** files. In case, your work is **NOT automated**, please submit just the **CSV** file.
- Submit your codebase as .py file (***FAQ_codebase_n.py; n stands for file number)***.

II. Version 1: Build a simple similarity-based FAQ bot (30 points)

- Build a python program to take a question as input and return a <question, answer> pair as an output
- Use techniques like TF-IDF to convert documents to numbers and/or Word2Vec techniques (NLTK/ Spacy libraries work best)
- Implement a similarity-based approach (cosine similarity) to map between the question to the FAQ pair
- **Reference**: https://medium.com/analytics-vidhya/building-a-simple-chatbot-in-python-using-nltk-7c8c8215ac6e

Evaluation:

- Implement a simple command line interface to query the program and get a response.
- While evaluating, if asked a question, it should give a response. The evaluation will be based on the bot's response. (Submit: ***python_run.py***)

III. Version 2: Build a Dialogflow based FAQ bot (30 points)

- Use features of Dialogflow (refer to https://dialogflow.com/ and the tutorial in class)
- Add small talk capabilities (greetings, fallback, etc.)

Evaluation:

- Deploy and expose your bot as a URL.
- Paste your bot URL in the starting of the report.

IV. Discussion and Ways to Improve (10 points)

- Think of all the personas that may access this information bot (patients, users, kids, etc.) while building this flowchart and all the kinds of conversations they may want to have with your bot
- Draw and submit an ideal conversational flowchart for your bot. (Refer to this for an example of a conversation flowchart: https://static.packt-cdn.com/products/9781788993487/graphics/facfa90d-80e0-4da7-a2c0-2ca7b1120e31.png)
- Discuss ways to improve your bot (better data, algorithms, hybrid approaches, etc.) before you could host this for public use at production scale.

Evaluation:

- Explain your data collection, building process and use code screenshots and output screenshots (bot output & in case, automated knowledge base is built, it's output too). OUTPUT SCREENSHOTS ARE A MUST.
- Draw and submit an ideal conversational flowchart for your bot.
- Discuss ways to improve your bot (better data, algorithms, hybrid approaches, etc.) before you could host this for public use at production scale.
  (Submit: *chatbot_report.pdf*)