**QuestBot:**

**AI-Powered Question Answering System**

QuestBot is an AI-powered question-answering system that utilizes the power of the "deepset/roberta-base-squad2" pre-trained model fine-tuned on the SubjQA dataset. It can answer questions based on reviews from various domains, such as books, movies, grocery, electronics, hotels, and restaurants.

**Project Overview**

This project involves three main parts:

1. **Model Selection and Preprocessing**
   * Selected base model: deepset/roberta-base-squad2
   * Tokenizer: AutoTokenizer from the Hugging Face Transformers library
   * Maximum context length: 384 tokens
   * Stride: 128 tokens
2. **Model Fine-Tuning**
   * Utilized the SubjQA dataset for fine-tuning
   * Training script: QuestBot\_Implementation.ipynb
   * Fine-tuning parameters:
     + output\_dir: "roberta-finetuned-subjqa-movies\_2"
     + evaluation\_strategy: "epoch"
     + logging\_strategy: "epoch"
     + save\_strategy: "epoch"
     + learning\_rate: 2e-5
     + num\_train\_epochs: 5
     + weight\_decay: 0.01
     + fp16: True (mixed-precision training)
3. **Demo Application using Gradio**
   * Created an interactive demo application using Gradio
   * Users can input a question and context to receive insightful answers from the QuestBot model

## **Features**

* Seamlessly fine-tuned model for question-answering task on the SubjQA dataset.
* Gradio-powered user interface for interactive question-answering.
* Quick and relevant responses to user queries based on domain-specific reviews.
* Easy deployment and usage of the QuestBot application.

## **Prerequisites**

To run the QuestBot application locally, you need to have Python installed. You can install the required Python packages by running:

pip install transformers gradio

## **Usage**

1. Clone this repository to your local machine.
2. Install the required packages as mentioned in the Prerequisites section.
3. Run the Gradio-powered QuestBot application:

python questbot\_app.py

1. Access the QuestBot interface by opening a web browser and navigating to the provided local URL.

**Plan of Attack:**

1. ***Data Preprocessing***: Clean and preprocess the dataset to remove any unwanted information and tokenize the text for input to the model.
2. ***Model Selection***: Choose a pre-trained roberta model from the Hugging Face Transformers library as the base model for fine-tuning.
3. ***Fine-Tuning***: Fine-tune the roberta model on the custom dataset using the question-answering objective. Train the model to predict the answer given a question as input.
4. ***Save the Fine-Tuned Model***: Save the fine-tuned roberta model after the training process so you can use it later for inference.
5. ***Deployment***: Deploy the fine-tuned roberta model as a QnAbot using a user-friendly interface (e.g., with Gradio) to allow users to input questions and receive relevant advice or information as responses.

**Dataset Info:**

**SubjQA**

SubjQA is a question answering dataset that focuses on subjective (as opposed to factual) questions and answers. The dataset consists of roughly 10,000 questions over reviews from 6 different domains: books, movies, grocery, electronics, TripAdvisor (i.e. hotels), and restaurants. Each question is paired with a review and a span is highlighted as the answer to the question (with some questions having no answer).

**Citation:**

@inproceedings{bjerva20subjqa, title = "SubjQA: A Dataset for Subjectivity and Review Comprehension", author = "Bjerva, Johannes and Bhutani, Nikita and Golahn, Behzad and Tan, Wang-Chiew and Augenstein, Isabelle", booktitle = "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing", month = November, year = "2020", publisher = "Association for Computational Linguistics", }

link: <https://github.com/megagonlabs/SubjQA.git>

**Model Training**

To fine-tune the "deepset/roberta-base-squad2" model on the SubjQA dataset, follow the steps outlined in the "Model Implementation” file named as “QuestBot\_Implementation.ipynb”.

**Challenges Faced**

* Handling overlapping answers within the context during fine-tuning.
* Tuning hyperparameters to balance training speed and model performance.

**Acknowledgments**

This project was completed as part of Course: Neural Network and Large Language Models from Learners Space, IITBombay.

Special thanks to the all course instructors for providing constant support.