**NLP TRAINING LAB**

**QUESTION 1**

Build a model for doing the Generative Question Answering task. Specifically, given a question and support documents related to the given question, the model will output the answer.

* **Framework**: Pytorch, Tensorflow, JAX/FLAX, …
* **Backbone**: BART, T5, GPT family, … Choose 1 backbone but recommend experimenting with multiple backbones to compare their performance (LoRA)
* **Keywords**: Generative Question Answering, Abstractive Question Answering, Long-form Question Answering, Text Generation, Decoding Text algorithms, …
* **Target**:
  + Metric: rougeL score (f1) >= 18% on validation dataset
  + Which algorithms do you use to generate text? Analyze it and others if possible.
  + Report: submit a report on the research process to build the model, the model building process and the results achieved (metric and loss). Extract 10 worst results and 10 best results for analysis

**QUESTION 2**

Unlike the Generative Question Answering task, the dataset will often provide more context (paragraphs containing knowledge used to answer the question, not the answer). In case the dataset does not provide context, I have to find a way to provide context to train. During the inference process, random questions should also be provided with support documents.

Build a Question Answering system that answers open-domain topic questions. The user enters a question from the keyboard, the system receives the question, queries support documents for that question in the database and generates the answer.

* **Framework**: Pytorch, Tensorflow, JAX/FLAX
* **Generative Question Answering backbone**: from **QUESTION 1**
* **Keyword**: Information Retrieval, Dense Passage Retrieval, Sparse Passage Retrieval, Long-form Question Answering, …
* **Target**:
  + Successfully built a database (PostgreSQL, SQL, ...) with knowledge taken from Wikipedia Snippets (optionally choose 17 million or 33 million data versions). There must be at least 500,000 data in the database, it would be nice if the database has enough data of Wikipedia Snippets
  + Interactive feature: Allows to enter questions from the keyboard and return answers, support documents are queried from the database (if possible)
  + Report: submit a report on the process of building the Question Answering system

**INSTRUCTION**

* Please submit your code via GitHub and construct your repository on [this structure](https://drive.google.com/file/d/1L9pK1LeCfykEC_mb2xPMQo4T09dRP77c/view?usp=sharing).
* All training and validation history (losses, scores, etc.) must be plotted using tensorboard, mlflow, wandb, ... and plots must be in the report.
* All test results must be visualized.
* Three questions must be finished within 3 weeks. Please schedule your time carefully in order to complete this training lab in time*.*
* Dataset:
  + Generative Question Answering dataset (**QUESTION 1** and **QUESTION 3**): [here](https://drive.google.com/drive/folders/18WxyhC7abCsuWYwjwHdbl6Z84MXuCQEw)
  + Wikipedia Snippets (**QUESTION 2**): [here](https://huggingface.co/datasets/wiki_snippets)
* TRLX github: <https://github.com/CarperAI/trlx>
* [wikipedia · Datasets at Hugging Face](https://huggingface.co/datasets/wikipedia/viewer/20220301.de/train?row=0) dataset update for question 2

Happy coding !