Practical Session Neural Machine Translation

The goal of this practical work is to compare the results of different models for Machine Translation in terms of the evaluation metrics as discussed in the lecture. Following are the steps in a nutshell:

1. Complete the exercises in Notebook `RNN_Machine_Translation.ipynb`.
2. Download the benchmark datasets 1.
3. Get the results for the advanced models (transformers and prompt-based model).
4. Evaluate and Compare the performance of different models.
5. Perform error analysis.
6. Finally Upload two Files: 1) RNN_Machine_Translation notebook. 2) Write down your analysis and the comparison of different models in a separate file and upload it along with your code. (You can just compare the transformers and prompt-based models.)

Note: Points 3, 4, and 5 are optional for the Recurrent Neural Network (RNNs).

1 Benchmark Datasets for Machine Translation

During this lab, we will use one of the datasets from the shared task on machine translation and specifically work on English-French translation. We will use the parallel corpus of Europarl. The Europarl parallel corpus is extracted from the proceedings of the European Parliament. It includes versions in 21 European languages: Romanic (French, Italian, Spanish, Portuguese, Romanian), Germanic (English, Dutch, German, Danish, Swedish), Slavik (Bulgarian, Czech, Polish, Slovak, Slovene), Finni-Ugric (Finnish, Hungarian, Estonian), Baltic (Latvian, Lithuanian), and Greek.

The dataset can directly be downloaded from this link: `https://www.statmt.org/europarl/v7/fr-en.tgz`. It contains two files, **(i)** europarl-v7.fr-en.en and **(ii)** europarl-v7.fr-en.fr consisting of the English as well as the French versions respectively. The corpus contains 2007723 sentences which can lead to a lag in the computational time given the large number of sentences. Consequently, for this practical session, you can extract the 15000 sentences.

2 Models for Machine Translation

During this practical work, we are going to use the following three models.

- Recurrent Neural Network based Model
- Transformers-based Model
- Prompting Large Language Model (Llama3)

Recurrent Neural Networks based Models. The first model uses a Recurrent Neural Network for Neural Machine Translation. More details are given in the dedicated notebook: `https://colab.research.google.com/drive/1aQmoh4Hl18Ll1pS9AN9jEFtuhD6RXRIV#scrollTo=ta5yGpZ23amq`.

Transformers based Models. This part uses performs transformer based machine translation. The Transformer, introduced in the paper "Attention Is All You Need", is a powerful sequence-to-sequence modeling architecture capable of producing state-of-the-art neural machine translation (NMT) systems. You can use the following notebook to play with Transformer-based models: `https://colab.research.google.com/drive/13y0PLrr14sXnDB-cK1WvkeEAA1ffzKgC?usp=sharing`.

Prompting Large Language Models. In this part of the exercise, we will use the open-source Large Language Models and prompt them to perform translations. In particular, we will use Ollama specifically to run LLMs locally.
Installation.

```
$ curl https://ollama.com/install.sh | sh
$ ollama pull llama3:8b
$ ollama run llama3:8b
```

You can interact via the command line. For running Ollama with Langchain, we will follow this documentation: `https://tinyurl.com/4f6v8st9`
Now it is time to start translating!