TD no5	**Utilisons un LLM**

Objectif de la séance :
Comprendre le fonctionnement d'un LLM. Trouver le bon prompt.

1 Prochain token

Les modèles de langage (LLM - Large Language Models) sont des intelligences artificielles entraînées à prédire des séquences de mots à partir d'un texte donné, appelé *prompt*. Ils fonctionnent en générant des *tokens*, qui sont des unités de texte pouvant être des mots, des parties de mots ou des caractères individuels.

L'objectif de cet exercice est de mieux comprendre le fonctionnement d'un LLM.

Connectez-vous sur la page Web `https://iti-llm.insa-rouen.fr/prochain_token` et testez les deux LLM avec les prompts suivants :

— Le petit chat boit
— La choucroute est plus grande
— Qui est le président de la république française ?
— Qui est le premier ministre de la république française ?

2 Un peu de *prompt engineering*

L'objectif de cet exercice est de mieux comprendre l'impact du prompt lors de l'utilisation d'un chatbot. Pour cela nous allons utiliser le mini chatbot téléchargeable depuis moodle. Ce script prend en paramètre :

— l'URL du LLM (`https://iti-llm.insa-rouen.fr`)
— le nom du modèle (utilisez Codestral)
— la clé API

En plus ce script accepte les paramètres optionnels suivants :

— `--temperature` qui permet de fixer la température du LLM (par défaut 0, donc déterministe) ;
— `--preprompt` qui permet de fixer un préprompt (par défaut une chaîne vide) ;
— `--avec_memoire` qui permet d'indiquer que le chatbot doit conserver en mémoire les échanges précédents (par défaut `False`).

2.1 Contextualisation des questions

Lancez le chatbot sans utiliser les paramètres optionnels. Posez les questions suivantes :

— Explique moi ce qu'est un LLM
— En informatique, explique moi ce qu'est un LLM
— En intelligence artificielle, explique moi ce qu'est un LLM

2.2 Influence de la température

Lancez le chatbot en fixant la température à 0.5. Posez plusieurs fois la même question.

2.3 zero et one shot prompting

L'objectif de cet exercice est de transformer une chaine de caractères ayant une certaine structure en une donnée python structurée. Par exemple transformer :

```
"Alice, 30 ans, Paris; Bob, 27 ans, Lyon; Clara, 35 ans, Marseille"
```

en :

```
[{'nom': 'Alice', 'age': 30, 'ville': 'Paris'},
 {'nom': 'Bob', 'age': 27, 'ville': 'Lyon'},
 {'nom': 'Clara', 'age': 35, 'ville': 'Marseille'}]
```

L'objectif est d'obtenir uniquement cette liste, rien d'autres.

- Essayez de trouver un prompt qui réalise cette action sans donner d'exemple (zero-shot-prompting).
- Donnez dans le prompt la transformation de `"Jean, 45 ans, Lyon; Marie, 32 ans, Nantes"` et demandez le même résultat pour `"Alice, 30 ans, Paris; Bob, 27 ans, Lyon; Clara, 35 ans, Marseille"`.

3 Un LLM pour MNIST

Nous allons voir si un LLM peut remplacer un algorithme d'apprentissage supervisé. Nous allons utiliser pour cela le jeu de données MNIST.

Téléchargez et décompressez l'archive `mnist.zip` disponible sur Moodle. Dans le répertoire `mnist` vous trouverez :

- `main.py`, le programme principal ;
- `mnist.py`, le module proposant une fonction (`reconnaitre_chiffre`) pour reconnaitre le chiffre d'une image en interrogeant un LLM avec un prompt donné et une fonction pour évaluer le taux de bonne reconnaissance d'un LLM sur une banque d'images (un répertoire par chiffre)
- `llm_image_et_texte_vers_texte.py` le module qui interroge le LLM avec une image et un texte à l'aide de la fonction `interroger_llm`.

1. Lancez le programme `main.py` avec l'option `-h` pour étudier les options possibles. Vous utiliserez le modèle Qwen2.
2. Trouvez le bon prompt pour que le LLM produise uniquement le chiffre reconnu.
3. Donnez le taux de bonne reconnaissance sur l'extrait du jeu de données MNIST pour votre prompt.