TD IA

Ingé 2 – S8 – ENSEIRB-MATMECA Info

Dialogue avec les LLM

Attention : Dans ce TD-TP, vous allez utiliser des modèles LLM fine-tunés ou non, mis en production par l'école pour ce TD-TP. Même s'ils ont tous été entrainés sur des données ne contenant pas de tels propos. Les productions de texte peuvent donc être offensantes et nous vous demandons de ne pas chercher à produire ce genre de texte *(!! tous les logs des échanges que vous ferez en ligne sont sont conservés sur le serveur* !!).

Nous allons utiliser deux modèles `mistralai/Mistral-Small-3.1-24B-Base-2503` et `mistralai/Mistral-Small-3.1-24B-Instruct-2503`. Ce sont des modèles très gourmands en mémoire, donc il se peut que les délais soient un peu longs lorsque toute la classe les interroge en même temps. Les interrogations se feront directement en Python via une série de code pre-remplis. Les interactions avec les LLM se feront via un protocole proposé par Open AI. Prenez déjà le temps de regarder les classes dans `LLMQuery.py`.

Tout ce TD/TP peut également se faire en local (sur vos propres machines) en installant par exemple `LMStudio` ou `Ollama` sur votre machine avec un modèle plus petit (mais la qualité des réponses s'en sentira). Pour cela il suffit de changer l'URL d'interrogation dans le fichier donné.

Important : Pour vous connecter au LLM, vous devrez utiliser un « token » à écrire en dur dans le fichier principal. Ce code (le token d'authentification) vous sera donné par votre enseignant pour la séance.

Exercice 1 :

Dans cet exercice, vous allez vous familiariser avec un modèle de langage en version « brute » c'est-à-dire avant la phase « INSTRUCT » de fine tuning. Ce modèle a été entrainé avec des données filtrés (et non offosantes) mais n'a pas été « fine tuné » pour être mieux aligné avec les valeurs de l'école.

a) Vous devez utiliser le LLM pour répondre à la question : « **Pourquoi le ciel est-il bleu ?** ». La réponse du LLM devra être claire et concise. Vous devrez essayer différentes température (de 0 à 1) ainsi que differents max_token.
b) Généralisez votre méthode pour répondre à d'autres questions, en paramètre.

Exercice 2 :

Nous allons maintenant utiliser la version *fine-tuned* pour discuter (la version INSTRUCT). Cette version prend en entrée des échanges entre un utilisateur et son assistant. Cette version ne prend pas un simple prompt en entrée (sous forme de texte) mais un `json` structuré (voir le fichier concerné) qui représente un dialogue entre un assistant et un utilisateur (rôles « user » et « assistant »). Quand ces systèmes sont en production, le role « assistant » est caché à

l'utilisateur. C'est ainsi qu'un modèle peut prendre des instructions pour répondre à des questions d'utilisateur.

a) Comme vous pouvez le voir, l'assistant a un secret. Votre objectif est de trouver un prompt (en tant que « user ») tel que ce prompt fait dire au LLM son secret. Vous n'avez le droit qu'à une seule interrogation (comme indiqué dans le script).
b) Changer le prompt « assistant » pour renforcer la protection du secret.
c) Bouclez pour améliorer la solidité de la protection et l'efficacité de l'attaque.

Exercice 3

Il faut maintenant interroger le LLM pour avoir de bonnes réponses à des tests de différents niveaux, typiquement utilisés pour évaluer leurs performances.

a) Prendre le script `question3` et relever les questions pour lesquelles le LLM ne donne pas de bonnes réponses.
b) Proposez d'autres prompts que celui donné : celui-ci, ne laisse pas beaucoup de possibilités de réflexion au LLM. Il faut proposer un prompt système demandant au LLM de faire une analyse plus fine de la question pour augmenter ses performances. Vous devez par exemple lui demander de décomposer la question pour y répondre. Tester ce nouveau prompt **sur une seule question** pour ne pas surcharger le serveur.
c) Améliorez (encore) ce prompt en faisant du *few-shot* en lui montrant des exemples de réflexion à faire sur des questions et des réponses pour montrer la granularité de réflexion.
d) Comme la réflexion peut être un peu longue, récupérez tout le texte de réflexion et utilisez le pour demander au LLM d'en faire une synthèse courte et claire.

Exercice 4

Nous allons introduire le doute dans ce que dit le LLM. On va faire comme si c'est lui qui doute et se pose des questions sur ce qu'il vient de dire. C'est très simple. Nous allons poser des questions en anglais et, toutes les 2-3 phrases, ajouter un « wait, wait. » (comme si le LLM avait des doutes), puis interrompre le traitement recommencer en construisant un prompt adéquat (comme si le « wait, wait. » était dans sa sortie).

a) Lisez le code et testez le en augmentant ou diminuant les injections de doutes.
b) Ajoutez un mécanisme permettant de répondre à la question initiale en injectant des doutes (mais en limitant le nombre d'injections tout de même), puis en faisant une synthèse de toute la réflexion.

Exercice 5

Dans cet exercice, nous allons faire un petit RAG à la main (Retrieval Augmented Generation). Pour cela, on explique au LLM comment interroger wikipedia (regardez le prompt). Ensuite, on parse les sorties du LLM jusqu'à trouver le mot clés qui attend ensuite les observations. Essayez différentes questions et observez les réflexions internes du LLM.

a) Proposez d'autres prompts / interactions pour corriger les éventuelles défaillances ou boucles du système.