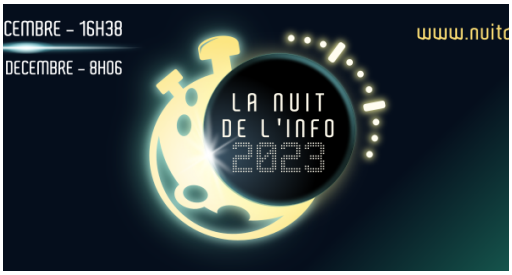
Reporting: Mettez de l'intelligence dans votre moteur



Intégration de l'IA dans l'Application

L'intégration de l'IA s'est manifestée principalement à travers le chatbot, capable d'interagir efficacement avec les utilisateurs en répondant à leurs questions sur le climat. Parallèlement, l'analyse des données climatiques via l'IA a permis de transformer des ensembles de données complexes en visualisations compréhensibles. Imaginairement, un outil d'IA prédisant les impacts à long terme des actions climatiques aurait été un ajout idéal.

Use-Case Illustratif

Imaginez un utilisateur demandant au chatbot l'efficacité des énergies renouvelables. L'IA analyse la requête, recherche dans sa base de données enrichie et fournit une réponse éducative basée sur des données fiables, démontrant ainsi la capacité de l'IA à faciliter l'éducation sur le climat.

Annexe Technique

Pour mettre en place ce use-case, nous nous sommes dans un premier lieu penché sur gpt4all, un outils permettant de faire tourner un model LLM en local sur nos machines (ne voulant pas utiliser ChatGPT qui est payant), cette solution était bonne mais nous avons rencontré des problèmes lorsque la carte graphique était disponible. Nous nous sommes finalement rabattus sur un modèle utilisant llama. Llamacpp offre un serveur python avec la même spécification technique qu'OpenAI que nous avons pluggé sur notre api au travers du package openai.

Spécification technique de la machine :

- 24 coeurs
- A100
- 500GB de disque

Spécification du modèle d'IA :

- TheBloke/Llama-2-70B-Chat-GGUF
- llama-2-70b-chat.Q5_K_M.gguf

Nous n'avons malheureusement pas eu le temps ni les ressources afin de fine-tuner un modèle avec des données précises vis à vis du sujet de la nuit, alors nous avons décidé de prendre un large modèle avec un connaissance large sur le sujet et d'injecter des prompts avant ceux de l'utilisateur pour guider l'IA sur ces réponses.

Problématiques IA Rencontrées

L'installation du serveur nous a causé plusieurs problèmes, la machine étant provisionnée sur un OpenStack, la choix du version de OS et du kernel était imposé (Fedora 39) et la version de

gcc nécessaire pour llamacpp était trop récente, il a donc fallu compiler gcc et plusieurs autres librairies. Cuda-toolkit a besoin d'une version gcc <= 12, tandis que gcc de fedora 39 est gcc 13. Cuda nous a également joué des tours avec des mauvaises versions. Mais nous avons pu faire face à tous ces problématiques pour enfin pu charger notre modèle en vram sur le gpu :

```
[fedora@nuitdelinfo-simon ~]$ nvidia-smi
Fri Dec  8 01:45:11 2023
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 545.29.06              Driver Version: 545.29.06    CUDA Version: 12.3     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA A100 80GB PCIe          Off | 00000000:00:05.0 Off |                    0 |
| N/A   33C    P0              62W / 300W |  48351MiB / 81920MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A    335504      C   ./server                                  48338MiB |
+---------------------------------------------------------------------------------------+
```

Nous donnons ces performances :

```
print_timings: prompt eval time =     131.18 ms /     0 tokens (     inf ms per token,     0.00 tokens per second)
print_timings:        eval time =   16791.57 ms /   260 runs   (   64.58 ms per token,    15.48 tokens per second)
print_timings:       total time =   16922.76 ms
```

Ce modèle est utilisé pour deux fonctionnalités:

- Répondre aux questions de l'utilisateur.
- Proposer à l'utilisateur une auto-complétion des questions selon les premiers mots de sa requête.

Puisque le modèle n'est pas entraîné sur des données en particulier, il est nécessaire d'injecter des prompts avant les requêtes de l'utilisateur, afin de guider l'IA sur sa génération. Les prompts indiquent à l'IA que son but est de générer/répondre à des questions liées à la transition écologique, le développement durable, etc..