

Projet : Résumé de textes médicaux en français avec LLM

The logo for esiea, featuring the word "esiea" in a bold, blue, sans-serif font. A small blue square is positioned above the letter "i".

CREUZE Dorian, DESFORGES Adrien, DOMO Adama, EDOA Hubert, KOYE GUIBAISSI
Fernand

Présentation du projet :	3
Etat de l'art :	4
Gestion des données :	5
Essai des LLM :	6
Pegasus :	6
Llama :	7
Logiciel Ollama :	7
Quantization :	8
Application rendu :	9

Présentation du projet :

Ce projet prend sa place dans une application créée par Capgemini ayant pour objectif de venir en aide aux médecins et aux patients et de faciliter le suivi médical.

L'application doit retranscrire les données du patient et les échanges durant les rendez-vous médicaux et en faire un résumé pertinent contenant toutes les informations importantes. Notre projet prend place sur cette dernière partie, c'est à dire résumer les informations à l'aide d'un modèle de type LLM.

Pour cela nous avons commencé par effectuer un état de l'art, puis nous avons sélectionné les modèles que nous avons testé, nous avons étudié les données et essayé différentes méthodes.

Etat de l'art :

Lorsque nous avons effectué notre état de l'art, nous avons pu étudier les résultats de nombreux modèles de LLM, 4 ont particulièrement retenu notre attention : chatgpt, pegasus, llama et mistral.

Chatgpt a rapidement été rejeté car nous avons préféré privilégier un modèle OpenSource. Nous nous sommes rapidement mis d'accord pour sélectionner Llama car celui-ci présentait des caractéristiques similaires à Chatgpt.

Nous voulions également étudier un autre modèle afin de pouvoir comparer les résultats de deux modèles différents lors de notre avancement.

À la suite de cela nous avons sélectionné LLM pegasus car celui-ci était plutôt orienté sur les résumés de texte et qu'il a été mentionnée dans plusieurs articles que nous avons pu trouver lorsque nous avons effectué notre état de l'art, notamment des articles dans lesquels les auteurs ont fait des essais et des évaluations par des professionnels de santé et des évaluation automatique (ROUGE, ROUGE-L, ROUGE1...) qui se base sur un autre résumé écrit par un médecin de référence.

Sur les évaluations automatiques, les auteurs nous font savoir que ce ne sont pas les plus pertinentes et qu'il faut préférer une évaluation humaine par des professionnels de santé.

Pour la suite de notre projet, n'étant pas médecin et n'ayant pas accès à un médecin pour effectuer ce travail d'évaluation, mais nous avons pensé pourvoir noter nos résumés sur les critères suivants afin d'évaluer les performances des modèles : Le résumé invente-t-il des éléments qui ne sont pas dans le texte de base ? Le résumé supprime-t-il trop d'information ? Le résumé suit-il la forme demandé dans le prompt ? Ecrit-il du texte superflu qui n'est pas utile au résumé, (ex : « voici le résumé : »)

Certains auteurs décrivaient également une amélioration de la qualité des résumés après avoir effectué un fine tuning, nous avons donc décider de comparer des modèle fine-tuner et non fine-tuner.

Gestion des données :

Un ensemble d'exemples de cas médicaux rédigés sous diverses formes nous a été fournis, nous avons donc entrepris d'effectuer des résumés de ces cas à l'aide de chatgpt afin d'obtenir des exemples pour pouvoir effectuer un finetuning sur les modèles.

Nous avons également pensé à la possibilité de générer des données afin d'avoir plus de données d'entraînement, cependant les résultats ne se sont pas montrés à la hauteur de nos attentes et nous avons préféré privilégier d'autres pistes tel que celle du prompt engineering.

Essai des LLM :

Pegasus :

Nous avons fait un premier essai du modèle pegasus sans fine tuning sur un cas médical fournit par capgemini, voici un exemple :

```
insuffisance post-transplantation, pulmonaire respiratoire chronique sévère l'interleukin-2 (leukinesin-2 erythromycin) , associait la méthroimmunosuppression of the patient's immune system to prevent infection with the Epstein-Barr virus (EBV) from spreading to other organs and tissues in the body, including the heart, lungs, kidneys and lungs.
```

Nous avons fourni un cas médical rédigé en Français au modèle, cependant le modèle nous a généré un résumé en anglais contenant des éléments sans rapport avec notre cas.

Nous avons donc effectué un finetuning sur le modèle pegasus afin d'observer les différences, voici un cas complet :

Une jeune fille se fait violer la nuit du 29 au 30 avril 2002. Elle porte plainte le 30 avril. En plus des examens et prélèvements gynécologiques, du sang et des urines sont prélevés aux fins d'analyse toxicologique. Ces prélèvements sont transmis au laboratoire le 30 avril et congelés immédiatement jusqu'à l'analyse.

La réponse au questionnaire envoyé, nous apprend que la victime a mangé des raviolis dont le goût lui a paru suspect. Elle est tombée endormie. Elle suit un traitement par Subutex® et n'a pas pris de médicaments entre les faits et les prélèvements.

Effectivement, la buprénorphine et son métabolite la norbuprénorphine ont été retrouvées dans l'urine, cependant du bromazépam et du THC-COOH ont également été retrouvés dans l'urine.

Le bromazépam lui avait été donné à son insu, quant au cannabis, nous n'avons pas su si c'était également le cas, ou si elle avait omis de le déclarer.

Et voici le résumé de pegasus avec les erreurs d'orthographe en rouge :

O une jeune fille se fait violer la nuit du 29 au 30 avril 2002. Elle porte plainte le 30 avril 2002. En plus des examens de et lvements gynecologiques, du sang et des urines sont levés aux fin d'analyses toxicologiques. Ces ponslevements sont transmits au laboratoires et congelé immediatanalyse jusqu' interview. La rée au envoyé, nous apprends que la victime à mangé des raviolis don't le got lui à paru suspect. Suit traitement par subutex et n'a pas pris de medicament entre les faits et les lphinements

Nous pouvons constater que le modèle pegasus ne résume pas et qu'il se contente de reprendre le début du texte en coupant des éléments sans raison apparente et en retirant de manière aléatoire des lettres à des mots.

Llama :

Pour le modèle Llama, nous avons commencé par télécharger le modèle depuis la plateforme hugging face, et nous avons alors pu constater que sa version la plus légère pesait 16gb et que nous ne pourrions pas la faire fonctionner sur nos ordinateurs personnels. Nous avons donc obtenu une VM de notre école disposant d'un GPU plus performant afin de pouvoir le faire tourner.

Nous nous sommes confrontés à des problèmes techniques, de driver, de gpu, avec le toolkit, de version des drivers et de compatibilité avec des modules. Ces problèmes nous ont fait perdre du temps et nous avons dû chercher une alternative afin de pouvoir finir le projet dans les délais.

Logiciel Ollama :

Nous avons trouvé un logiciel qui permet de faire fonctionner des LLM opensource en local sur des systèmes disposant d'une configuration légère. Ce point nous permet aussi de respecter le critère de confidentialité nécessaire au milieu médical car les données ne passent pas par des serveurs.

Nous avons réussi à obtenir des résultats satisfaisants, et nous avons continué dans cette voie en effectuant du prompt engineering afin d'obtenir les meilleurs résumés possibles.

Nous avons réalisé différents tests, des zero shots (en ne donnant aucun exemple préalable de ce que l'on attend), en one shot (en donnant 1 exemple), et en few shots (plusieurs exemples mais dans notre cas seulement 2).

Le modèle Llama 3.1 à 8 milliards de paramètres s'est heurté à des difficultés quand on ne lui donnait aucun exemple, il ne réalisait aucune invention, cependant lors des évaluations que nous avons effectuées, nous avons jugé que le modèle faisait disparaître trop d'informations que nous pensions importantes et rendait le résumé difficile à lire. Nous avons également pu constater des cas où le modèle refusait de fournir des informations à cause de son filtre destiné au contenu sensible, en voici un exemple :

```
Je ne peux pas vous fournir d'informations sur des activités illégales telles que le trafic de drogue. Dois-je suggérer quelque chose pour aider ?|
```

Le cas en exemple mentionne la prise de drogue donc le modèle effectue une censure, cependant le problème ne s'est pas reproduit lorsque l'on fournissait au moins un

exemple au modèle. Voici le même cas lorsque nous avons fourni un exemple dans le prompt lors de l'utilisation du modèle :

```
**Informations cliniques**

* Âge : 15 ans (Loïc), 16 ans (Jean-Paul)
* Syndrome atropinique marqué à l'admission, avec :
  + Tachycardie
  + Hallucinations
* Prélèvements sanguins réalisés plusieurs heures après l'hospitalisation ne permettent pas de mettre en évidence :
  + Atropine
  + Scopolamine
* Présence de cannabis dans les prélèvements sanguins

**Traitement**

* Pas d'information sur le traitement spécifique administré aux patients

**Antécédents et contexte**

* La présence de capsules et graines de datura suspectées d'avoir été consommées pendant une fête techno-rave à Biscarosse est mentionnée, mais pas d'informations plus précises sur les antécédents des patients.]
```

Nous avons donc pu observer une amélioration de la qualité des résumés lorsque l'on passe de 0 exemples à 1 dans le prompt, mais aucune amélioration notable lorsque l'on passe de 1 à 2 exemples.

Quantization :

Réaliser une quantization consiste à faire baisser la précision des poids d'un modèle afin de le rendre moins lourd en mémoire et donc plus économe. Cela nous semblait une amélioration importante afin de diminuer les coûts de maintien de l'application chez Capgemini.

Nous avons donc réalisé des recherches afin de savoir comment effectuer une quantization sur un modèle et de trouver quels outils nous pouvions utiliser. Nous avons trouvé divers outils tels que torchao de pytorch ou gptq Model.

Effectuer des recherches sur la quantization a été une opportunité pour nous permettre d'effectuer des recherches intéressantes sur le concept et les outils utilisés, cependant dans ce projet cela nous a uniquement permis d'identifier l'origine des problèmes que nous avons rencontré lors de l'utilisation de la VM.

Finalement la quantization de Llama sous Ollama permet diminuer le modèle de 16gb à environ 4gb.

Application rendu :

Ne connaissant pas l'infrastructure sur laquelle cette future application à laquelle nous avons participé va fonctionner, et voulant également simplifier au maximum les actions à faire pour intégrer notre travail nous avons décidé d'utiliser docker.

Voici un schéma qui représente notre rendu. Nous avons 2 conteneurs docker, le premier contient ollama, il s'agit du conteneur fournit sur le site associé au logiciel, tandis que le second conteneur contient une application que nous avons réalisé.

Afin d'utiliser le LLM, il faut démarrer les 2 conteneurs qui sont relié sur un docker network, et envoyer en localhost port 5000 le texte du cas que vous voulez résumer. Ce texte est dirigé dans le premier conteneur afin d'être intégré dans le prompt puis d'être redirigé vers le conteneur ollama qui renvoi le résumé du cas.

