Synthèse et conclusions

Jeu de données

Durant ce test, nous avons exploré un jeu de données composés de contenus de brevet et de leurs résumés. Le jeu de données est construit sous forme de triplet (query, positive_patent, negative_patent) où le positive_patent est sensé être le brevet le plus proche a retrouver a partir de la query, et le negative_patent un brevet proche en terme de contenu, mais qui ne correspond pas a l'objet de la query.

Problème à résoudre

Le problème à résoudre est de trouver une manière de réaliser les embeddings des brevets de telle sorte que les exemples positifs soit plus similaires à la query que les exemples négatifs. De plus, dans une tache de retrieval, pour la query donnee, il faut que l'exemple positif arrive dans le top K, contrairement a l'exemple négatif.

1- Exploration des données

L'exploration des données nous a permis de faire notamment les observations suivantes:

- le jeu de données est composé de contenus de brevet et d'abstracts de brevets
- la longueur des documents est importante, il faut donc réaliser du chunk avant d'ingérer les données dans un LLM

2- Méthodes testées

Nous avons testé trois méthodes pour réaliser l'embedding des brevets, toutes utilisant le modèle intfloat/e5-small-v2 :

- méthode 1: on fait l'embedding directement sur tout le contenu du document (méthode zero-shot). On teste aussi la variante en finetunant le modèle avec la contrastive loss.
- méthode 2 : on fait l'embedding des brevets a partir des embeddings de sections en realisant un max pooling. Chaque embedding de section etant lui-même obtenu par max pooling des embeddings de chunks composant la section
- méthode 3 : on fait l'embedding des brevets a partir des embeddings de résumés de sections en réalisant un max pooling. Les résumés de section étant constitué des 5 phrases les plus importantes constituant la section

3- résultats obtenus

- méthode 1 Zero-Shot:
  - Accuracy : 74.15 %
  - top_5_accuracy positive : 93.4 %

    - top_5_accuracy negative : 71.3 %
- méthode 1 Finetuning (probleme d'overfitting) :
    - Accuracy : 100 %
    - top_5_accuracy positive : 19.4 %
    - top_5_accuracy negative : 0 %
- méthode 2:
    - Accuracy : 90.18 %
    - top_5_accuracy positive : 93.4 %
    - top_5_accuracy negative : 41.7 %
- méthode 3:
    - Accuracy : 81.76 %
    - top_5_accuracy positive : 88.37 %
    - top_5_accuracy negative : 58.1 %

4- Interprétation des résultats

La méthode 2 donne les meilleures résultats. Elle a nécessité un travail plus approfondi pour comprendre les brevets en exploitant les sections.

Les résultats de la méthode 3 ne sont pas si satisfaisants car la manière de générer les résumés n'est efficace. Il faudrait plutôt générer des vrais résumés à partir de LLM entrainés sur une tâche de summarization, plutôt que de prendre directement les phrases du document.

5- Idées a explorer pour la suite

Les idées qui viennent tout de suite pour une future exploration sont les suivantes :

- Tester des LLM plus performants
- Résumer chaque section directement par un LLM
- Faire le finetuning des modèles en exploitant la contrastive loss sur les chunks construits avec la méthode 2 plutôt qu'en brute force directe sur tout le document
- Avoir un plus gros dataset, avec plus d'exemples négatifs pour chaque query

In [ ]: