Hackathon Airbus

Génération de résumé de paragraphe jurique d'Airbus Hélicopter

Eliott Crancée, Cléa Han, Yanis Labeyrie, Léa Triquet, Adrien Zabban

Centrale Méditerranée, 13013 Marseille, France

1 Introduction

Notre objectif est de réaliser des résumés de textes issus de contrats légaux. Cela est traduit mathématiquement par la maximisation de la métrique que nous avons appelée *god metric* qui correspond à la moyenne du f1 score de rouge entre le résumé généré et le résumé de référence et leur similarité des embeddings.

2 Traitement des données

Nous avons décidé de ne pas utiliser les données open sources, car elles n'étaient pas adaptées à notre tâche. On a découpé les données en une base d'entraînement et une base de test. On les a converties au format csv qui nous a été utile les cas de finetuning. La base d'entraînement nous a permis de finetune le transformer et de nourrir le RAG. La base de test nous a permis d'évaluer les résultats de nos différentes options.

3 Architecture du modèle

On peut voir que notre modèle comporte deux branches: le transformer et le LLM. Il faut noter que dans deux tiers des cas, le résumé sélectionné sera celui du transformer au détriment de celui du LLM. D'ailleurs, on fait en sorte de ne pas effectuer l'inférence par le LLM lorsque l'on est sûr de son inutilité. Ainsi, si vous souhaitez avoir le meilleur résumé dans tous les cas, il est conseillé d'utiliser tout le modèle, mais si vous souhaitez une architecture plus légère, le transformer seul donne déjà de bons résultats.

3.1 Branche Transformers

Flan T5 En premier lieu, nous faisons passer le texte original dans le transformer Flan T5 que nous avons fine-tune sur 10 epochs. Par ailleurs, avant d'arriver sur Flan T5, nous avons testé plusieurs modèles différents (Bart et T5) avant de choisir Flan T5 pour ses performances. Nous générons 10 résumés différents avec des variations aléatoires d'hyperparamètres comme le *num_beams*, *length_penalty* (en plus de la température de Flan T5).

Xgboost Nous avons entraîné un xgboost pour prédire la *god_metric* sachant que les f1 score des différentes métriques rouges et la similarité cosinus entre le texte d'origine et le résumé généré. Nous avons pris la décision de ne pas lui donner en entrée le rapport entre des longueurs du résumé et du texte d'origine pour ne pas favoriser les longs résumés.

Find Best Find best est un module qui prend le texte d'origine et une liste de résumés générés de ce texte et qui estime le meilleur résumé. Pour cela, il moyenne les métriques rouges et similarité cosinus avec la prédiction du xgboost.

3.2 Branche LLM

On a séparé les textes en cinq catégories de taille: très court, court, standard, long, très long. Les textes très courts ne passent pas par le LLM.

RAG Grâce au RAG, on sélectionne les 3 embeddings les plus proches de notre texte original. Ensuite, on sélectionne celui dans la longueur de texte est la plus proche, c'est celui-ci qui servira d'exemple au LLM. On le lui fournira avec son résumé.

LLM: gemma 7b Après quelques tests de différents LLM, comme Mixtral, Mistral, Zephyr, T5, nous avons choisi d'utiliser Gemma 7b it (instruction tuned) car il donne de meilleurs résultats. Nous avons fait du prompt engineering pour tirer parti du meilleur de Gemma. Voici le prompt que nous avons utilisé :

On commence par la phrase d'introduction: "As a legal expert of Airbus Helicopter's terms and conditions, you are skilled in generating summaries of text."

On fournit ensuite l'exemple trouvé par le RAG: "Here is a well-crafted example that follows the rules: Example text : [texte d'exemple] Example

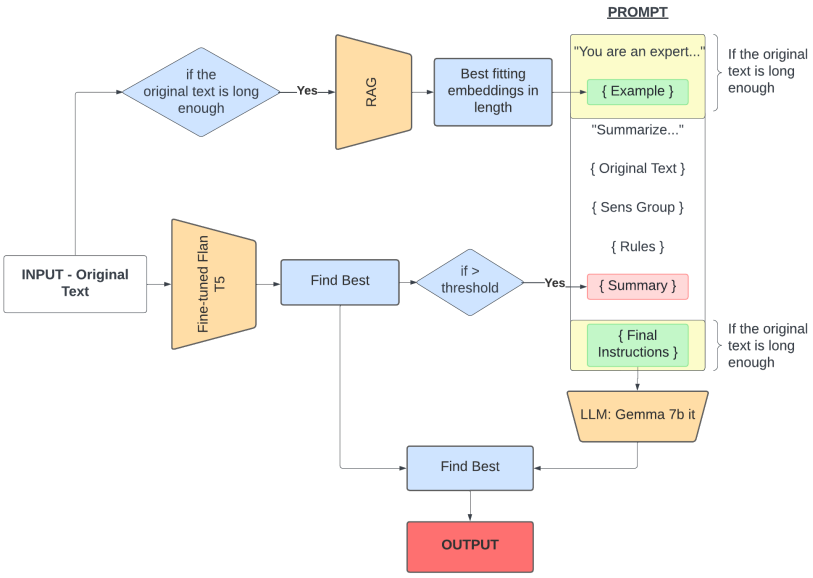

Fig. 1. Architecture de notre modèle

summary : [résumé d'exemple] (Sauf pour la catégorie de texte courts).

Puis, on met la consigne principale: "Summarize the following text accurately by keeping the semantics and sens group." (A noter que pour les textes courts, on mets plutôt: "Summarize the following text accurately")

Ensuite, on fournit le texte à résumer.

Puis viennent les groupes de sens: grâce à la bibliothèque *spaCy*, on a découpé les textes en:

- Un verbe racine autour duquel le texte s'articule
- Tous les groupements nominaux rattachés à ce verbe. C'est ce qu'on appelle 'groupe de sens'.
- La formulation du prompt est la suivante : "Here are the sens group of the text: The head and most important sens group is: [verbe racine] [énumération des autres groupes de sens] "

On ajoute les règles de résumé que l'on a repérées en analysant la base de données (passage à la voie active, formulations positives, suppression des références, . . .)

On fournit également le résumé produit par la branche Transformer. Pour les textes courts et standards, on utilise le prompt: "Here's a very good summary generated by a language transformers: [résumé T5]", et pour les textes plus longs, on utilise plutôt: "Here's a good but too long summary generated by a language transformers: [résumé T5] "

Enfin, on ajoute la dernière consigne: "Copy the transformers summary, propose only very small improvements if necessary. Return only the string text content without commenting or formatting the new summary:", et pour les textes très longs: "Improve the transformers summary, propose only very small improvements if necessary, shorten it. [...] "

Find Best On renvoie le meilleur résumé entre celui généré par le LLM et le Transformers Flan T5.

4 Ressources

Le temps d'inférence dépend proportionnellement des ressources informatiques utilisées.

Nom	time (s)	GPU	vRAM
t5	442	RTX 4050	6 Go
t5	393	RTX 3050	4 Go
t5	340	A100	40 Go
t5 et gemma	356	A100	40 Go

Tableau des temps d'inférence en fonction des ressources informatiques. Les deux premières inférences ont été faites en local sur nos ordinateurs portables, et les deux dernières ont été exécutés sur un serveur.

5 Conclusion

Notre modèle donne de bonnes performances selon les métriques d'évaluation, mais pourrait être encore amélioré en changeant ces dernières. En effet, elles pénalisent les textes courts, ainsi les résumés générés sont parfois encore un peu longs. En ajoutant une pénalité sur la longueur du résumé, on pourrait donc obtenir de meilleures performances.