Chatbot de recommandation de films
Natural Language Processing

BARBOSA Mathias, LASGLEIZES David,

RIBAS Justine, THIRÉ Mael

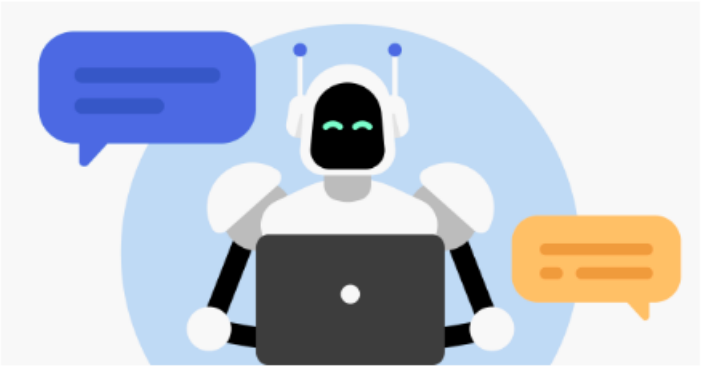
ING3 IA1

2023 / 2024

Table des matières

1 Introduction

Ce projet consiste à la conception d'un chatbot capable de recommander des films en se basant sur un jeu de données spécifique. Les interactions conversationnelles seront assurées par l'intégration d'un LLM. Ce rapport détaillera le processus de conception et d'implémentation du chatbot. Nous explorerons également les défis rencontrés lors du développement, les solutions adoptées, et les résultats obtenus.

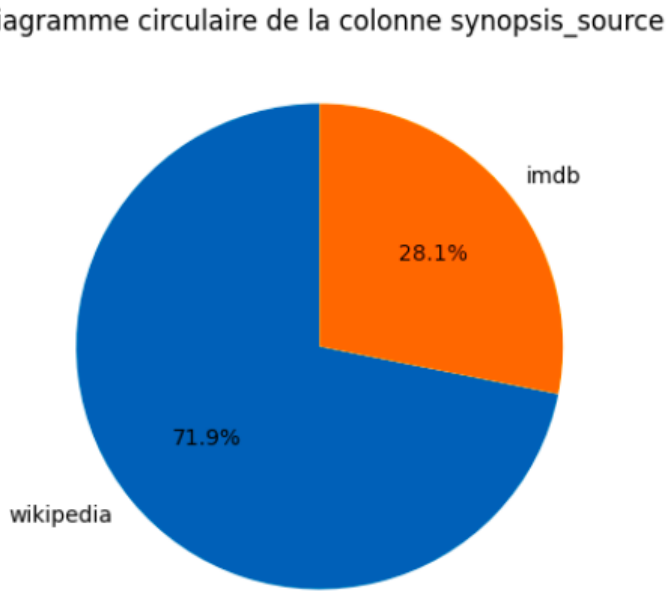


2 Base de données

La base de données que nous avons utilisée est “MPST : Movie Plot Synopsis with Tags” sur Kaggle : lien. Nous avons choisi cette dernière car chaque film a des tags ainsi que son synopsis et selon nous, ce sont des champs clé pour déterminer les similarités. Le jeu de données a été créé en 2018 et répertorie environ 14 000 films provenant de wikipedia et imdb. Le jeu de données contient les variables suivantes :

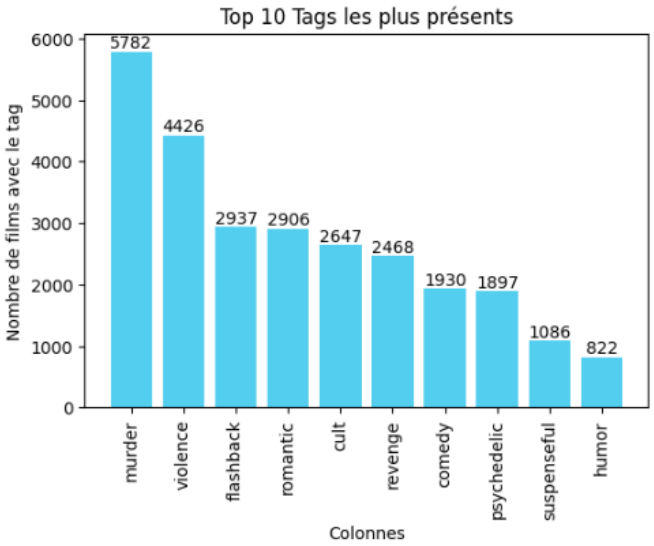
- l’identifiant du film
- le titre du film
- le synopsis du film
- une liste de tags
- un split de train, test, validation
- la source de donnée du synopsis du film

Nous avons effectué une rapide exploration de ce jeu de données. Premièrement, voici la répartition des sources de données pour le synopsis des films :



En analysant le jeu de données, nous avons remarqué qu’il n’y avait pas de données Null. Cependant nous avons constaté qu’il y avait des doublons de titres de films, nous en avons dénombré 1071. Ces doublons peuvent impacter notre recommandation de film, en effet, si nous cherchons les films similaires à Titanic et que notre solution renvoie tous les films qui ont le titre Titanic sans information complémentaire cela a peu d’intérêt pour l’utilisateur, il manquerait par exemple la date de réalisation. Nous allons donc enrichir notre jeu de données pour pallier ce problème.

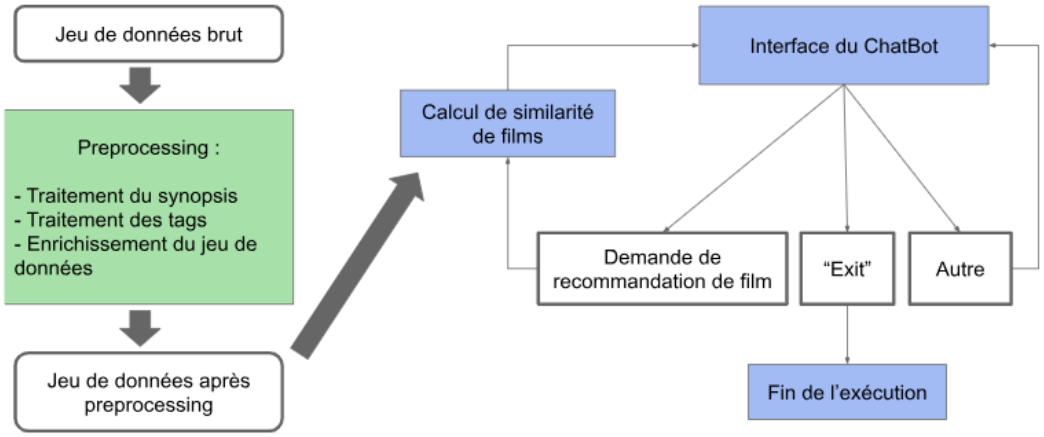
Nous avons identifié 71 tags différents dans l'ensemble du jeu de données, voici les tags les plus récurrents :



3 Notre solution

3.1 La description de notre solution

Dans notre solution, nous pouvons voir deux parties principales : le preprocessing du jeu de données et le chatbot. La partie preprocessing sera effectuée en amont et donnera un jeu de données nettoyé et exploitable. Ce sera dans la partie chatbot que nous déterminerons les recommandations de films, en utilisant le jeu de données avec preprocessing, lorsque ceci sera demandé par l'utilisateur. Voici un schéma récapitulatif :



3.2 Les étapes de traitement de la solution

Le pre processing des données

Lors du preprocessing de notre jeu de données, nous avons tout d'abord traité le champ synopsis. Le synopsis est un long texte, donc pour le rendre exploitable nous l'avons tokenisé puis avons effectué divers traitements. Nous avons décidé d'utiliser deux méthodes de tokenisation : treebank et spacy. Ensuite nous avons supprimé les stopwords, mis le texte en minuscule, gardé les tokens de plus de 2 caractères seulement et gardé les tokens alphabétiques seulement.

Concernant les tags, nous avons également décidé de les traiter pour les rendre plus exploitables. En effet, dans les données brutes les tags étaient stockés sous forme d'une liste. Nous avons décidé de créer une colonne par tag et de remplir le champ par 0 ou 1 selon la présence ou non du tag dans la liste, un peu comme du One-Hot Encoding. Cette façon de représenter les données est plus compréhensible pour les modèles et facilitera nos calculs.

Nous avons décidé d'enrichir notre base de données lorsque l'on a fait face à un problème de répétition de titres. Notre jeu de données contient des films différents ayant le même nom. Cependant nous n'avions pas de moyen pour les différencier et avons donc décidé de faire du web scraping pour étoffer nos informations. Ainsi nous récoltons les années de sortie des films que nous avons en doublons. De cette façon, lors de la proposition de film en corrélation, il est possible de différencier les films homographes.

Le calcul de similarité

Le calcul de la distance à l'aide de la similarité du cosinus se révèle particulièrement adapté dans le contexte d'une base de données constituée de synopsis de films, en prenant en compte la concaténation des synopsis avec les tags associés aux films. Chaque synopsis, enrichi des tags attribués, est représenté comme un vecteur où chaque composante correspond à un terme spécifique présent dans les résumés ou les tags. Les valeurs de ce vecteur reflètent les pondérations attribuées à ces termes dans chaque synopsis. L'utilisation de la similarité du cosinus permet ensuite de mesurer l'angle entre ces vecteurs, offrant ainsi une indication précise de la similarité sémantique entre les films. Cette approche intégrant les tags et les résumés facilite des comparaisons de titres de manière efficace, particulièrement utile pour des applications telles que la recommandation de films ou la recherche de titres similaires au sein d'une base de données cinématographique.

Le chatbot intégré avec un LLM

Pour assurer la partie conversationnelle du chatbot, nous avons utilisé un LLM. Premièrement nous avions voulu utiliser le modèle GPT-2 puis notre notre choix s'est finalement porté sur le modèle Flan-T5 développé par Google Research. Lorsque le chatbot est lancé, on analyse si la question de l'utilisateur concerne la recommandation de film ou non. Le cas échéant nous identifions le nom du film et appelons notre méthode de distance pour recommander les 10 films les plus proches. Si la question de l'utilisateur ne concerne pas ce sujet, nous afficherons la réponse du modèle. Enfin nous avons défini une condition d'arrêt : “exit” ou "bye".

3.3 Les résultats

Démonstration

Pour demander une recommandation de film, le prompt doit être une phrase contenant les mots clé “recommend” ou “give” et les mots clé “film” ou “movie” avec le nom du film entre guillemets. Voici quelques cas de test de notre solution pour chacun des modèles.

Cas de demande de recommandation de film :

Entrez votre message : Can you recommend me a film similar to 'The Shop Around the Corner' ?
Bot : Here is a list of movies you could like, based on your preferences : Alfred the Great, Salmon Fishing in the Yemen, Helium, Dance of the Vampires, The Burning, Legends of the Fall, Die Blechtrommel, The Batman vs. Dracula, The Pillars of the Earth, Careful
Entrez votre message : exit
Bot : Bye.

Cas de recommandation d'un film similaire à un film qui a des doublons de titre dans la base :

Entrez votre message : Can you give me a movie similar to 'Titanic' ?
Bot : Here is a list of movies you could like, based on your preferences : Titanic, Titanic (2012), Titanic (1953), Titanic (1943), State of Play, Crazy, Stupid, Love., East of Eden, The Ballad of Jack and Rose, Shank, Shank (2010)
Entrez votre message : exit
Bot : Bye.

Cas de recommandation d'un film similaire à un film qui n'existe pas dans la base :

Entrez votre message : Can you recommend me a movie similar to 'Imaginary Film' ?
Bot : Sorry we don't have Imaginary Film in our database.
Entrez votre message : exit
Bot : Bye.

Cas de question qui utilise la partie conversationnelle du LLM Flan-T5 :

Entrez votre message : Do you like movies ?
Bot : yes, i like movies.
Entrez votre message : exit
Bot : Bye.

Cas de question qui utilise la partie conversationnelle du LLM GPT-2 :

Entrez votre message : Do you like movies ?

Bot : I like to watch movies. I'm not a big movie guy, so I don't really know what I want to do with my life. But I do have a lot of movies that I like, and I
Entrez votre message : exit
Bot : Bye.

Vous trouverez un exemple de test dans une vidéo de démonstration présente dans le rendu final.

Les améliorations possibles

La première amélioration qui nous vient à l'esprit, concerne le modèle LLM conversationnel. En effet, nous avions dû le brider en terme de puissance car notre puissance de calcul ne permet pas de faire tourner un modèle un peu plus performant. C'est pourquoi, le chatbot a parfois des réponses qui ne sont pas toujours cohérentes.

Une autre amélioration pourrait être de conserver un historique de discussion. L'intérêt est de pouvoir rebondir sur ce qui a été recommandé : par exemple, l'utilisateur pourrait demander d'afficher le synopsis d'un des films recommandés, car ce film en particulier l'intéresse. Ceci demanderait également une plus grande puissance de calcul pour conserver un tel historique. Il est cependant possible de créer une fonction qui affiche le synopsis d'un film précis et qui ne nécessiterait pas de conserver l'historique, mais dans ce cas, l'utilisateur ne pourrait pas se contenter d'une demande comme : "Quel est le synopsis du troisième film recommandé ?", il devrait préciser le titre dans sa demande.

Une dernière amélioration possible, serait de continuer le scrapping des données, pour cette fois-ci, obtenir les avis des personnes ayant vu le film. Cela permettrait d'affiner la recommandation de films, en mettant en avant les films les plus appréciés.

4 Notre méthodologie de travail

4.1 Les outils utilisés

Pour réaliser ce travail, nous avions à notre disposition une base de données détaillée précédemment, un script sur google colab pour travailler en groupe et un dossier drive pour pouvoir utiliser la base de données et conserver des modifications de chaque membre du groupe.

4.2 La répartition du travail

Pour ce qui est de la répartition du travail, nous avons procédé comme-ci :

Mathias s'est occupé de tout ce qui est interaction avec les deux chatbots LLM, ainsi que l'importation des données. Il a également participé à la synchronisation des différents modules réalisés par chacun, pour que le tout puisse fonctionner à la fin. Enfin, il a rédigé la fin du rapport : à partir de la partie concernant les amélioration et a mis en forme le rapport sur LaTeX.

David, quant à lui, s'est occupé du scrapping des données pour obtenir les années de réalisation, il a également créé la fonction NLP de calcul de distance pour ensuite recommander une liste de films. Il a rédigé les parties calcul de similarité et enrichissement des données.

Justine a réalisé toute la partie exploration de données : séparation des tags, one-hot encoding des tags, répartition des sources et des tags de la base de données. Elle a rédigé tout le reste du rapport et a joué le rôle de cheffe de projet, en supervisant régulièrement notre travail.

Enfin, Mael a réalisé le pre-processing des synopsis : tokenisation et suppression des stopwords.

5 Conclusion

Ainsi, ce projet consistait à réaliser un chatbot de recommandation de films, les résultats obtenus sont globalement satisfaisants mais laissent une perspective d'amélioration. Un tel chatbot en pratique, est très utile, à condition d'avoir la puissance de calcul nécessaire pour lui permettre d'être encore plus efficace.