3ème année du cycle ingénieur CYTech, option Intelligence Artificielle

Mathias Barbosa

David Lasgleizes

Justine Ribas

Maël Thire

Chatbot de recommandation de films



Natural Language Processing - 2023/2024

Sommaire :

[**Introduction 3**](#_paa6f6ath7m3)

[**La base de données 3**](#_q9lv4nqe6een)

[**Notre solution 3**](#_ejs2k0gjt7qb)

[La description de notre solution 3](#_rlv939akrqxz)

[**Les étapes de traitement de la solution 3**](#_rilfyhxfjfcd)

[Le pre processing des données 3](#_6584eeqze2lv)

[Le calcul de similarité 3](#_o6979n6ikg)

[Le chatbot intégré avec un LLM 3](#_o9dmpg7j26pe)

[**Les résultats 3**](#_h4lrw9yxjgkp)

[Démonstration 3](#_x4b40m99rore)

[Les améliorations possibles 3](#_x5aeeupnn29o)

[**Notre méthodologie de travail 3**](#_2cqr9q54gr8f)

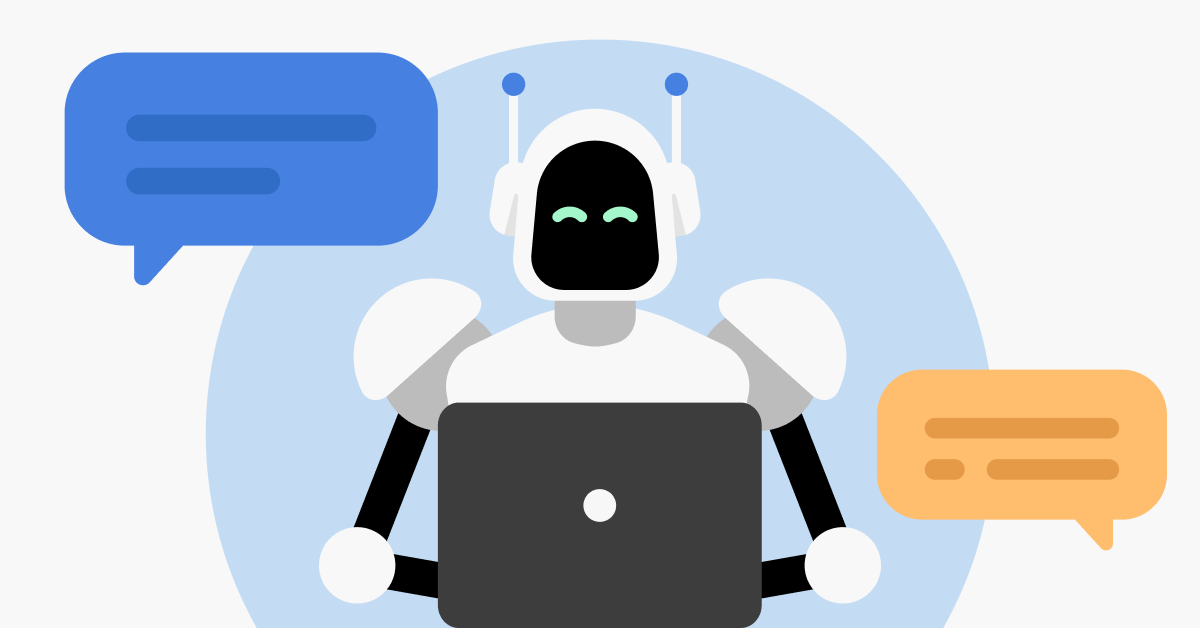
[Les outils utilisés 3](#_nx8gjrweu49f)

[La répartition du travail 3](#_yxtggarur0ic)

[**Conclusion 3**](#_6m6dz526s80a)

# 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.





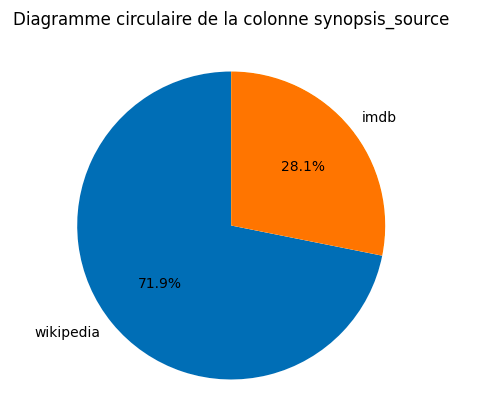
# 

# La base de données

La base de données que nous avons utilisée est “MPST: Movie Plot Synopsis with Tags” sur Kaggle : [lien](https://www.kaggle.com/datasets/cryptexcode/mpst-movie-plot-synopses-with-tags). 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 :

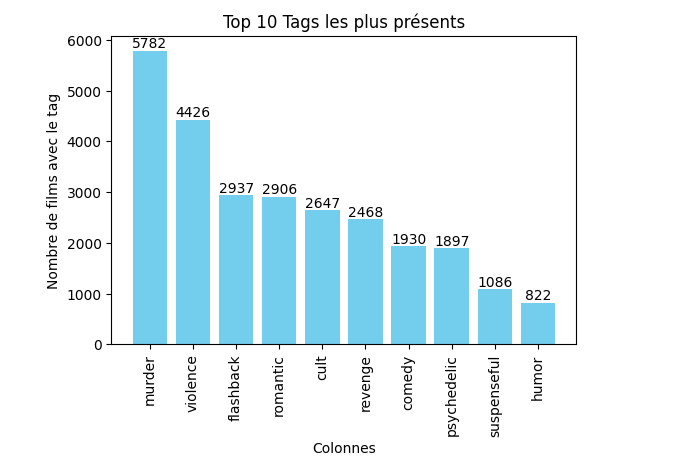
* l’identifiant du film
* le titre du film
* le synopsis du film
* une liste de tags
* un split de train, test, validation
* la source de donnée du synopsis du film

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. 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 :



# Notre solution

## 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 :



## 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 lorsqu’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 les 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 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”.

## 

## 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 résumé de ces cas de test dans une vidéo de démonstration présente dans le rendu final

### Les améliorations possibles

# Notre méthodologie de travail

### Les outils utilisés

Pour pouvoir travailler ensemble sur ce projet, nous avons majoritairement utilisé l'outil de travail Google Colab. En effet, cet outil nous permettait de travailler à plusieurs sur le même notebook et ainsi de vérifier que notre contribution personnelle fonctionnait correctement avec le reste du script.

### La répartition du travail

Pour avancer efficacement dans ce projet, nous avons décidé de découper le travail en sous tâches puis de les assigner aux membres du groupe. Pour la partie preprocessing, le traitement des tags et du synopsis a été réalisé par Maël et Justine. L’enrichissement de la base de données a été effectuée par David qui a également travaillé sur le calcul de similarité des films. Enfin, l’importation du LLM et le développement du ChatBot ont été réalisés par Mathias.

# Conclusion