

Implémentation d'un chatbot avec un système de RAG



Natural Language Processing
Janvier 2025

Antoine CHARLET, Ewan GOURONNEC,
Océane TENEBAY, Ayyoub ZAROU

Contents

- 1 Introduction 2
- 2 Dataset 2
 - 2.1 Données 2
 - 2.2 Traitement des données 2
- 3 Méthodologie 3
 - 3.1 Modèles 3
 - 3.1.1 Embedding 3
 - 3.1.2 LLM 3
 - 3.2 Mécanismes Avancés 3
- 4 Implémentation 4
 - 4.1 Données 4
 - 4.2 Modèles 4
 - 4.3 Mécanismes Avancés 5
 - 4.3.1 Hybrid Search 5
 - 4.3.2 Reranking 5
 - 4.4 Résultats 6
- 5 Conclusion 7

1 Introduction

L'objectif de ce projet est d'implémenter un chatbot basé sur un système RAG en exploitant des techniques avancées comme la recherche hybride (Hybrid Search) et le reranking. Ces approches permettent d'améliorer le RAG notamment en augmentant l'efficacité du retrieval et la pertinence des réponses.

Nous avons choisi d'appliquer cette méthodologie sur des données liées au cinéma, en utilisant la base de données MPST: Movie Plot Synopses with Tags. Notre chatbot sera évalué sur sa capacité à retrouver et à synthétiser des informations pertinentes à partir de cette base. Notre objectif est d'analyser les apports concrets de ces mécanismes de RAG avancés sur le retrieval et la génération dans le cas spécifique de notre chatbot.

2 Dataset

2.1 Données

Nous avons utilisé pour ce projet le dataset MPST: Movie Plot Synopses with Tags. Il contient des résumés de films (synopsis) et des tags décrivant leurs thèmes, genres, et autres caractéristiques spécifiques. Ce dataset est intéressant pour notre étude car les synopsis peuvent servir de base documentaire pour la génération de réponses de notre chatbot.

2.2 Traitement des données

Les données sont récupérées directement sur la plateforme Kaggle avec la bibliothèque kagglehub. Le dataset MPST est ensuite chargé sous forme d'un dataframe Pandas. Le fichier contient alors plusieurs informations sur les films, notamment le titre, les tags, et synopsis.

Nous ne conservons que les données de test afin d'alléger la quantité de données à traiter et nous supprimons les lignes contenant des valeurs manquantes. Ensuite, nous ajoutons les sources (un URL vers la page IMDb de chaque film est généré à partir de son identifiant.) et nous fusionnons les colonnes titre, tag et synopsis afin de transformer chaque ligne du dataframe en un document. Pour finir, ces documents sont segmentés afin de permettre un meilleur traitement.

3 Méthodologie

3.1 Modèles

3.1.1 Embedding

Dès le début de notre étude, nous comparons deux modèles : mxbai-embed-large et all-minilm. Ces deux modèles, visent à encoder des textes dans des représentations vectorielles pour des tâches de traitement automatique du langage naturel (NLP). Le modèle mxbai-embed-large est un modèle spécialisé conçu pour produire des embeddings denses et riches en informations sémantiques. Il est optimisé pour capturer des relations complexes dans des textes longs et contextuellement riches.

A contrario, le modèle all-MiniLM-L6-v2 est un modèle léger développé par Sentence Transformers. Il est conçu pour un encodage rapide et efficace des textes tout en maintenant une bonne qualité d'embedding. La taille des embedding est bien moindre (384 dimensions contre plus de 700 pour mxbai-embed-large) mais ce modèle a l'avantage d'être plus rapide.

3.1.2 LLM

Nous avons, à nouveaux, comparer deux modèles pour les LLMs : mistral et llama3.2. Llama 3.2 est un modèle polyvalent pour les tâches de traitement du langage naturel (NLP), comme la génération de texte, le résumé et la traduction automatique. Sa capacité à comprendre et à générer du texte semblable à celui d'un humain le rend populaire pour le développement de chatbots, la rédaction de contenu et la construction de systèmes d'IA conversationnels. Llama3.2 est intéressant pour créer un assistant conversationnel capable de répondre de manière précise et déterministe.

Mistral gère la génération de code et l'analyse de données à grande échelle, ce qui le rend idéal pour les développeurs travaillant sur des plateformes de codage pilotées par l'IA. Ses capacités de reconnaissance des formes lui permettent de s'attaquer à des tâches complexes. Mistral permet de répondre à des questions techniques ou complexes.

3.2 Mécanismes Avancés

Dans le cadre de notre étude, nous analysons l'impact de deux mécanismes avancés, la recherche hybride (Hybrid Search) et le reranking, sur les performances de notre chatbot. Ces méthodes sont utilisées dans des systèmes de récupération augmentée par génération (Retrieval-Augmented Generation, ou RAG) pour améliorer la pertinence et la qualité des réponses générées.

4 Implémentation

4.1 Données

Afin de séparer le traitement des données du corps de notre étude, nous avons créé un fichier `data.py` consacré à ce traitement. Il contient :

1. La fonction `load_movie_data()` qui est utilisée pour télécharger et charger le jeu de données en mémoire sous forme de `DataFrame`. (Le nombre total de films est affiché pour confirmer la réussite du chargement.)
2. La fonction `prepare_movie_data(movies)` se charge de nettoyer et structurer les données pour qu'elles soient exploitables sous forme de documents.
3. La fonction `split_movie_data(docs)` divise les documents en morceaux plus petits, facilitant leur traitement ultérieur.

Après utilisation des fonctions, nous obtenons un jeu de données contenant des synopsis de films pour une utilisation dans notre chatbot.

4.2 Modèles

Nous avons fait le choix d'utiliser FAISS pour créer une base de données vectorielles (Vector Store) utilisant les embeddings des modèles `mxbai-embed-large` et `all-minilm` (surnommé respectivement Large et Mini dans le code). Ils sont ensuite sauvegardés localement afin de gagner du temps pour les exécutions futures. Cela permet de rechercher des documents en fonction de leurs similarités vectorielles avec le prompt saisi.

Le fichier `models.py` contient une classe "Models" permettant de choisir un modèle d'embedding (`mxbai-embed-large` ou `all-minilm`) ainsi qu'un LLM (`llama3.2` ou `mistral`). Pour toutes les étapes suivantes, notre pipeline question-réponse se compose d'une `retrieval chain` permettant de chercher les informations pertinentes dans le vector store et du prompt suivant pour évaluer les réponses des LLMs :

"What's a good movie about an epic viking?"

On constate que Llama Mini (soit `all-minilm` pour les embedding et `llama3.2` pour le llm) est le plus rapide à fournir une réponse, qui cependant est moins pertinente. La réponse de Llama Large (soit `mxbai-embed-large` pour les embedding et `llama3.2` pour le llm), en revanche, concorde davantage à la demande de l'utilisateur. Quant au modèle Mistral, il est le plus lent à répondre, mais retourne deux suggestions de films. Mistral semble aussi halluciner une partie de la question de l'utilisateur.

Réponse Llama : "Based on your interest in an epic Viking movie, I would recommend "Beowulf". It is a legendary tale about a Geatish warrior who travels to Denmark to slay Grendel, a monstrous creature that has been terrorizing King Hrothgar's mead hall"

Réponse Mistral : "The movie that fits your description of an epic Viking story with elements of magical realism, psychedelic or avant-garde themes is 'Beowulf'.there isn't a movie titled as you described ("Alfred the Great)"

Nous avons ensuite comparé la similarité entre le prompt et les réponses de chacun des modèles en utilisant comme métrique le score BLEU, ainsi qu'une analyse humaine. On en a déduit que le modèle qui fournissait la réponse la plus pertinente est Llama Large, avec un score BLEU de 10.7%, contre 2.7% pour Mistral Large et 1.5% pour Llama Mini. Ces scores restent toutefois très faibles. Cependant, il faut considérer que ces scores représentent le niveau de correspondance entre la réponse générée et les documents récupérés (ce score n'est pas toujours pertinent.), l'analyse humaine est donc notre plus grand facteur de décision. Dans la suite de notre étude, nous avons décidé d'utiliser les embedding de mxbai-embed-large et le llm llama3.2.

4.3 Mécanismes Avancés

4.3.1 Hybrid Search

Pour implémenter Hybrid Search, nous avons combiné le retriever formé avec la base de données vectorielles FAISS avec le retriever BM25Retriever. BM25 (Best Matching 25) est un algorithme de récupération d'informations, généralement utilisé pour évaluer la pertinence d'un document par rapport à une requête. Cette approche prend en compte la longueur du document dans son évaluation de la pertinence, ce qui permet d'éviter que les documents les plus longs soient décrits à tort comme étant les plus pertinents.

Ces deux retrievers sont combinés en un, sous forme d'instance de la classe EnsembleRetriever, dans laquelle les poids de chaque retriever sont égaux. Nous avons fait le choix d'attribuer à chaque retriever le même poids. Cependant, nous pourrions pousser l'étude en modifiant les valeurs des poids et en essayant de trouver la meilleure combinaison.

4.3.2 Reranking

Nous avons ensuite implémenté le Reranking avec Flashrank. Pour commencer, le compresseur Flashrank effectue un second traitement sur les documents pour réévaluer leur pertinence. Ces documents sont ensuite réorganisés de manière à ce que les plus pertinents apparaissent en tête de liste. Finalement, nous avons combiné l'Hybrid Search et le Reranking.

Réponse Sans Mécanisme : "Based on your interest in an epic Viking movie, I would recommend "Beowulf". It is a legendary tale about a Geatish warrior who travels to Denmark to slay Grendel, a monstrous creature that has been terrorizing King Hrothgar's mead hall"

Réponse Hybrid Search : "'Based on your question "What's a good movie about an epic Viking?", I would recommend the title "Beowulf". It is a legendary story about a Geatish warrior who travels to Denmark with his band of soldiers to slay Grendel, a hideously malformed troll-like creature."

Réponse Reranking : "'Based on your question "What's a good movie about an epic Viking?", I would recommend "Beowulf". It is a legendary story of a Geatish warrior who battles a monstrous troll-like creature and has elements of action, adventure, and romance."

Réponse Hybrid Search + Reranking : "I think you may be confusing this with another movie, as Alfred the Great is not about an epic Viking. However, I can suggest some alternative movies that might fit what you're looking for: The 13th Warrior (1999) - a historical action-adventure film based on the novel "Eaters of the Dead" by Michael Crichton, which tells the story of a Viking warrior who joins a group of warriors to fight against a deadly threat."

4.4 Résultats

On observe des résultats bien plus concluants lors de l'utilisation de mécanismes RAG avancés que sans. Après Reranking, les réponses données sont tout à fait pertinentes, avec un "relevance score" de 99% pour les meilleures suggestions. Lorsque l'on combine Reranking et Hybrid Search, les scores se rapprochent encore plus de 100%, ce qui est notamment dû à l'évaluation de pertinence des réponses effectuée par le retriever BM25.

Au vu des résultats de notre étude, nous avons appliqué les mécanismes de RAG avancés à notre chatbot. Nous avons notamment appliqué l'Hybrid Search. Le chatbot nous donne les résultats suivants :

Bienvenue dans le chatbot ! Tapez 'quit' pour sortir.

Vous : Give me an action movie

Bot: I don't know what specific action movie you're looking for, but I can suggest some popular ones:

- * Die Hard
- * The Matrix
- * Mad Max: Fury Road
- * John Wick
- * Taken

If you have any specific preferences (e.g. blaxploitation, 80s/90s action movies), let me know and I can try to give you a more tailored recommendation!

Vous : What is Die Hard about ?

Bot: I don't know the exact plot of Die Hard with a Vengeance, but I can tell you that it's the third film in the Die Hard series. However, I do know that the first two films are:

- * Die Hard (1988) - The original film where John McClane (played by Bruce Willis) finds himself trapped in a skyscraper during Christmas Eve and must fight against a group of terrorists.
- * Die Hard 2 (1990) - The sequel to the original, where John McClane is again forced to confront a group of terrorists at Washington Dulles International Airport.

If you're looking for information on the plot of Die Hard with a Vengeance, I can try to help you find it.

Vous : I want a more recent movie

Bot: Based on the context of the two previous titles, I'd like to suggest some action-packed movies from the past few years that might interest you. Keep in mind that these are just a few examples, and there are many great action movies out there.

Here are a few recommendations:

1. ****John Wick: Chapter 3 - Parabellum**** (2019) - an adrenaline-fueled sequel to the John Wick franchise.
2. ****Bad Boys for Life**** (2020) - an action-packed buddy cop movie with Will Smith and Martin Lawrence.
3. ****Extraction**** (2020) - a high-octane Netflix original film starring Chris Hemsworth as a black-market mercenary.
4. ****The Old Guard**** (2020) - a Netflix original movie based on the comic book series, starring Charlize Theron and Kiki Layne.
5. ****F9: The Fast Saga**** (2021) - the ninth installment in the Fast and Furious franchise, with plenty of high-speed action.

These movies offer a mix of intense action sequences, thrilling chases, and memorable characters. Of course, there are many more great action movies out there, but I hope this gives you a good starting point!

Would you like more recommendations or have any specific preferences (e.g., superhero movies, martial arts films)?

Vous : quit

Au revoir !

Actin
Go to

5 Conclusion

Ce projet nous a permis d'acquérir une meilleure compréhension des mécanismes de Retrieval-Augmented Generation (RAG). Notre étude, présentée dans ce rapport, montre les avantages liés à l'implémentation de ces techniques avancées dans la création d'un chatbot. En exploitant des mécanismes tels que l'Hybrid Search et le Reranking, nous avons pu améliorer significativement la pertinence et l'efficacité des réponses générées par notre chatbot.