

ALBARICO Marjorie

M2 – Data

Email : albarico.marjorie00@gmail.com / marjorie.albarico@supdevinci-edu.fr

Projet 1 : Topic Modeling des Avis des Produits

(cf. [P1_Top_Modeling.ipynb](#))

Projet 2 : Système RAG avec Descriptions de Produits

Ce projet vise à créer une application qui utilise des descriptions de produits pour répondre de manière précise et fiable aux requêtes des utilisateurs.

L'objectif était de transformer un fichier JSONL contenant des métadonnées de produits en une base de données consultable, tout en utilisant des techniques avancées de traitement du langage naturel. Voici les étapes que j'ai suivies.

Étape 1 : Chargement des données

- **Données utilisées** : J'ai travaillé avec un fichier nommé meta.jsonl contenant des descriptions de produits.
- **Méthode** : J'ai chargé ce fichier avec **Pandas**, en utilisant une fonction spécifique pour les fichiers au format JSONL.
- **Défi rencontré** : Aucun problème majeur ici, mais j'ai vérifié la présence des colonnes nécessaires, notamment la colonne details.

```
# Fonction pour charger le fichier JSONL
def load_jsonl(file):
    # Charger le fichier JSONL dans un DataFrame pandas
    df = pd.read_json(file, lines=True)
    return df

# Charger et afficher les premières lignes du fichier
try:
    df = load_jsonl(uploaded_file)
except Exception as e:
    st.error(f"Erreur lors du chargement du fichier : {e}")
```

Étape 2 : Prétraitement et segmentation des textes

- **Objectif** : Diviser les descriptions longues en morceaux (ou "chunks") pour faciliter la génération des embeddings.
- **Méthode utilisée** : J'ai utilisé la classe `RecursiveCharacterTextSplitter` de `LangChain` pour diviser les textes.
- **Paramètres** :
 - Taille des morceaux : 512 caractères.
 - Chevauchement : 128 caractères.
- **Défi rencontré** : Au début, j'avais choisi d'utiliser la colonne « **descriptions** », mais cela m'a posé plusieurs problèmes. Certains textes étaient manquants, et les données de cette colonne ressemblaient davantage à des avis qu'à des descriptions détaillées des produits. Pour résoudre ce problème, j'ai opté pour la colonne « **détails** », qui est plus complète et contient des informations spécifiques et structurées sur les produits, ce qui correspondait mieux à mes besoins pour ce projet.

```
# Diviser les descriptions longues en morceaux de taille appropriée
chunk_size = 512
chunk_overlap = 128

# Vérifier si la colonne 'details' existe dans le DataFrame
if 'details' in df.columns:
    text_splitter = RecursiveCharacterTextSplitter
    (chunk_size=chunk_size, chunk_overlap=chunk_overlap)
    df['details_chunks'] = df['details'].apply(lambda x:
    text_splitter.split_text(str(x)) if pd.notnull(x) else [])
else:
    st.error("La colonne 'details' n'existe pas dans le fichier
    JSONL.")
```

Étape 3 : Génération d'embeddings

- **Objectif** : Convertir les morceaux de texte en représentations vectorielles.
- **Modèle choisi** : `sentence-transformers/all-MiniLM-L6-v2` via `HuggingFace`.
- **Défi rencontré** : Au début, j'ai eu des erreurs avec ce modèle, notamment des soucis liés aux dépendances. Cela m'a pris beaucoup de temps, mais j'ai réussi en utilisant la classe `HuggingFaceEmbeddings` de `LangChain`.
- **Résultat** : Les morceaux de texte ont été correctement transformés en vecteurs.

```
# Charger le modèle d'embedding
model = HuggingFaceEmbeddings(model_name="sentence-transformers/
all-MiniLM-L6-v2")
```

Étape 4 : Création d'une base vectorielle

- **Objectif** : Stocker les embeddings dans une base de données vectorielle pour permettre une recherche efficace.
- **Outil choisi** : J'ai utilisé **ChromaDB**.
- **Défi rencontré** : Initialement, j'ai rencontré des erreurs avec ChromaDB et la gestion des dépendances. Finalement, j'ai utilisé une intégration avec LangChain pour simplifier le processus.
- **Résultat** : J'ai créé une base vectorielle contenant tous les vecteurs générés.

```
vectorstore = FAISS.from_texts(
    texts=details_chunks_list,
    embedding=model)

retriever = vectorstore.as_retriever()
```

Étape 5 : Création d'un système de récupération

- **Objectif** : Configurer un système de récupération des documents pertinents en fonction des requêtes des utilisateurs. Cela permet de répondre uniquement à partir des données disponibles.
- **Méthode** : J'ai utilisé LangChain pour simplifier la gestion des embeddings et leur recherche dans une base de données vectorielle. À l'origine, j'ai rencontré des problèmes en utilisant certains outils comme ChromaDB avec RecursiveCharacterTextSplitter et all-MiniLM-L6-v2. Après plusieurs essais, j'ai décidé d'utiliser HuggingFace Embeddings et un stockage FAISS, qui ont fonctionné de manière fiable.

Étape 6 : Intégration avec un modèle LLM

- **Objectif** : Générer une réponse basée uniquement sur les documents pertinents.
- **Structuration du prompt** : J'ai conçu un prompt pour guider le modèle et m'assurer qu'il utilise uniquement les informations issues des documents fournis.

```
# Structurer le prompt pour le LLM
prompt_template = """
You are a helpful assistant designed to answer user queries using
only the information extracted from a set of product descriptions.
Your responses should strictly rely on the provided documents. If
you cannot find an appropriate answer in the documents, respond
with "I do not know."

Please make sure to include the exact passages or document details
from which you derive your answers.
Do not generate any information beyond what is provided in the
data, and avoid sharing sensitive, inappropriate, or potentially
incorrect information.

User query: {query}
Relevant documents: {documents}

Your response should be based solely on the relevant documents and
passages provided.
"""
```

Étape 7 : Recherche de Documents Pertinents

- **Objectif** : Localiser les documents les plus pertinents dans la base vectorielle en fonction d'une requête utilisateur.
- **Approche**
 1. **Transformation de la requête utilisateur en embedding** :
La requête est transformée en embedding à l'aide du même modèle utilisé pour les documents (sentence-transformers/all-MiniLM-L6-v2), garantissant une compatibilité dans l'espace vectoriel.
 2. **Recherche dans la base vectorielle** :
Le retriever utilise la méthode de similarité vectorielle de ChromaDB (ou FAISS dans ce cas) pour identifier les documents les plus proches.

```
# Demander à l'utilisateur de saisir une requête
query = st.text_input("Enter your query:", "Products from China?")
```

Étape 8 : Chaîne de récupération

- **Objectif** : Combiner les descriptions des documents pertinents avec la question utilisateur, puis passer l'ensemble dans un modèle LLM pour générer une réponse.

```

if query:
    # Effectuer la récupération des documents pertinents
    results = retriever.get_relevant_documents(query)
    relevant_documents = "\n".join([result.page_content for result
    in results])

    # Appeler le modèle avec les paramètres choisis
    response = query_gpt3_turbo_dynamic(query, relevant_documents,
    temperature, top_p, max_tokens)

    # Afficher la réponse générée
    st.write("Response from GPT-3.5 Turbo:")
    st.write(response)

```

Étape 9 : Exécution de Requêtes Utilisateur

- **Objectif :** Tester le pipeline RAG avec des questions utilisateur typiques pour évaluer la qualité des réponses générées.
- **Résultat attendu :** Les réponses doivent être strictement basées sur les documents extraits et pertinentes à la requête.
- **Exemples de Requêtes :**
 - « Products from China ? »

```

Response from GPT-3.5 Turbo:

Based on the information provided in the documents, the following products are from China:

1. Manufacturer: Reiko
   - Country of Origin: China
   - Date First Available: August 24, 2012
2. Manufacturer: GUANGZHOU WENYI COMMUNICATION EQUIPMENT CO.,LTD
   - Country of Origin: China
   - Date First Available: February 28, 2017
3. Manufacturer: Seidio
   - Country of Origin: China
   - Date First Available: December 6, 2013
4. Manufacturer: Shenzhen Putuo Technology Co.,Ltd
   - Country of Origin: China
   - Date First Available: October 18, 2021

Therefore, the products from the manufacturers Reiko, GUANGZHOU WENYI COMMUNICATION EQUIPMENT CO.,LTD, Seidio, and Shenzhen Putuo Technology Co.,Ltd are from China.

```

- « What is the product description? »

Response from GPT-3.5 Turbo:

Based on the provided product descriptions, here are some details extracted from the documents:

1. Product from Unicorns Pro:
 - Package Dimensions: 7.7 x 3.6 x 1.3 inches
 - Item Weight: 4.8 ounces
 - Other display features: Wireless
 - Form Factor: Bumper
 - What's in the box: Kickstand, Belt Clip
 - Manufacturer: Unicorns Pro
 - Date First Available: March 24, 2020
 - Material: Polycarbonate, Thermoplastic Polyurethane
 - Brand: Unicorns Pro
 - Fastener Type: Clip On
2. Product from Joyleop:
 - Product Dimensions: 5.91 x 2.76 x 0.59 inches
 - Item Weight: 2.08 ounces
 - Special features: Shock-Absorbent
 - Other display features: Wireless
 - Form Factor: Case
 - Manufacturer: Joyleop
 - Date First Available: March 9, 2019
 - Brand: Joyleop
 - Material: Silicone
 - Pattern: Animal
 - Special Feature: Shock-Absorbent
3. Product from Misisi:
 - Product Dimensions: 15.99 x 8.99 x 1 inches
 - Item Weight: 0.023 ounces
 - Is Discontinued By Manufacturer: Yes
 - Date First Available: November 19, 2019

○ « Who is Donald Trump ? »

Response from GPT-3.5 Turbo:

I do not know.

Étape 10 : Test de Différents Paramètres pour le Modèle LLM

- **Objectif** : Explorer l'impact des paramètres suivants sur la qualité des réponses générées :
 - **Temperature** : Contrôle la diversité et la créativité des réponses.
 - **Top-p** : Détermine la probabilité cumulée des options les plus probables.
 - **Max Tokens** : Définit la longueur maximale de la réponse.
- **Configuration Dynamique** : Les paramètres sont ajustables via l'interface Streamlit, permettant des tests en temps réel.

```
# Création d'une interface pour tester différents paramètres
temperature = st.slider("Temperature", 0.0, 1.0, 0.0, 0.1) #
Slider pour Temperature
top_p = st.slider("Top-p", 0.0, 1.0, 1.0, 0.1) # Slider pour Top-p
max_tokens = st.number_input("Max tokens", min_value=50,
max_value=1000, value=500, step=50) # Input pour max_tokens
```

Vue utilisateur :

Projet 2 : Création d'une architecture Retrieval Augmented Generation sur des descriptions de produits Amazon

Donnée utilisée : metadata.jsonl

Vector store contains 1000 vectors.

Temperature

0.00 0.40 1.00

Top-p

0.00 0.80 1.00

Max tokens

500 - +

Enter your query:

- Tests réalisés :

Temperature

0.00 0.80 1.00

Top-p

0.00 0.30 1.00

Max tokens

500 - +

Enter your query:

Products from China?

Response from GPT-3.5 Turbo:

Products from China are mentioned in the following document:

- Manufacturer: Reiko
- Country of Origin: China

This information is extracted from the document that includes details about a product with an orange color case from Reiko, where the country of origin is specified as China.

Cette réponse est concise mais omet des détails. Une réduction de top-p semble limiter les informations utilisées, ce qui peut ne pas être optimal pour ce type de tâche.

Temperature

0.00 0.80 1.00

Top-p

0.00 1.00 1.00

Max tokens

500 - +

Enter your query:

Products from China?

Response from GPT-3.5 Turbo:

Yes, there are products mentioned in the data that are from China. For example, the manufacturer of the product with the item model number 'PP-LGLS840ORG' is Reiko, and the Country of Origin for this product is China (as per the document details provided). Similarly, the manufacturer of the product with the item model number 'BD2-HR31PH5-GD' is Seidio, and the Country of Origin for this product is also China (as per the document details provided). Additionally, the manufacturer of the product with the item model number 'BYHA-001' is Shenzhen Putuo Technology Co.,Ltd, and the Country of Origin for this product is China (as per the document details provided).

La réponse est correcte mais un peu redondante. Cela peut être attribué à la hausse de temperature, qui rend les réponses moins structurées.

Temperature

0.20

0.001.00

Top-p

1.00

0.001.00

Max tokens

500

-+

Enter your query:

Products from China?

Response from GPT-3.5 Turbo:

Yes, there are products in the provided data that are from China. The manufacturers Reiko and Shenzhen Putuo Technology Co., Ltd have products listed with the country of origin being China.

- Reiko: "Country of Origin: China"
- Shenzhen Putuo Technology Co., Ltd: "Country of Origin: China"

These details are extracted from the following passages in the documents:

- Reiko: "{ 'Color': 'Orange', 'Whats in the box': 'Case', 'Manufacturer': 'Reiko', 'Country of Origin': 'China', 'Date First Available': 'August 24, 2012' }"
- Shenzhen Putuo Technology Co., Ltd: "{ 'Country of Origin': 'China', 'Date First Available': 'October 18, 2021', 'Brand': 'NonoUV', 'Connector Type': 'USB Type C', 'Compatible Devices': 'iPhone 14, iPhone 14 Pro, iPhone 14 Pro Max, iPhone 14 Plus, iPhone SE 2022, iPhone 13, iPhone 13 Mini, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 12, iPhone 12 Mini, iPhone 12 Pro, iPhone 12 Pro Max, iPhone SE (2020), iPhone 11, iPhone 11 Pro, iPhone 11 Pro Max, iPhone XR, iPhone XS Max, iPhone X, iPhone 8, iPhone 8 PlusiPhone 14, iPhone 14 Pro, iPhone 14 Pro Max, iPhone 14 Plus, iPhone SE 2022, iPhone 13, iPhone 13 Mini, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 12, iPhone 12 Mini, iPhone 12 Pr... See more', 'Compatible Phone Models': 'iPhone 14, iPhone 14 Pro, iPhone 14 Pro Max, iPhone 14 Plus, iPhone SE 2022, iPhone 13, iPhone 13 Mini, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 12, iPhone 12 Mini, iPhone 12 Pro, iPhone 12 Pro Max, iPhone', '13, iPhone 13 Mini, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 12, iPhone 12 Mini, iPhone 12 Pro, iPhone 12 Pro Max, iPhone SE (2020), iPhone 11, iPhone 11 Pro, iPhone 11 Pro Max, iPhone XR, iPhone XS Max, iPhone X

Cette réponse est légèrement moins exhaustive, probablement à cause de la légère augmentation de température, qui favorise des réponses plus variées.

Temperature
0.00

0.00 1.00

Top-p
1.00

0.00 1.00

Max tokens
500 - +

Enter your query:
Products from China?

Response from GPT-3.5 Turbo:

Based on the information provided in the documents, the following products are from China:

1. Manufacturer: Reiko
 - Country of Origin: China
 - Date First Available: August 24, 2012
2. Manufacturer: GUANGZHOU WENYI COMMUNICATION EQUIPMENT CO.,LTD
 - Country of Origin: China
 - Date First Available: February 28, 2017
3. Manufacturer: Seidio
 - Country of Origin: China
 - Date First Available: December 6, 2013
4. Manufacturer: Shenzhen Putuo Technology Co.,Ltd
 - Country of Origin: China
 - Date First Available: October 18, 2021

Therefore, the products from the manufacturers Reiko, GUANGZHOU WENYI COMMUNICATION EQUIPMENT CO.,LTD, Seidio, and Shenzhen Putuo Technology Co.,Ltd are from China.

La réponse est claire, précise et exhaustive. Elle utilise des informations directement extraites des documents.

Temperature
0.40

0.00 1.00

Top-p
0.80

0.00 1.00

Max tokens
500 - +

Enter your query:
who is donald trump?

Response from GPT-3.5 Turbo:

I do not know.

- **Conclusion**

1. **Performance des paramètres :**

- Une température basse (0) combinée à top-p élevé (1.0) donne des réponses précises et complètes.
- Augmenter la température ou réduire top-p introduit plus de variabilité ou limite les informations, ce qui peut être moins adapté aux cas nécessitant des réponses exhaustives.

2. Améliorations possibles :

- Tester d'autres modèles LLM (par ex. LLaMA).
- Augmenter la taille du contexte fourni au modèle.
- Affiner la segmentation des chunks pour optimiser la pertinence des résultats.

3. Progrès réalisé :

- L'intégration des tests dynamiques des paramètres via Streamlit permet d'explorer différentes configurations du modèle en temps réel.

Ce projet de création d'une architecture RAG (Retrieval Augmented Generation) représente une étape importante dans ma découverte de l'intelligence artificielle et de son fonctionnement. Bien que le sujet ne soit pas celui que je maîtrise le mieux, ce fut une expérience enrichissante, me permettant de plonger dans des concepts avancés tout en consolidant des bases essentielles.

La mise en œuvre du système, depuis l'importation des données jusqu'à l'intégration d'un modèle de langage dans une interface interactive, m'a permis de comprendre les différentes étapes nécessaires pour créer un outil capable de répondre à des requêtes de manière informée. Même si ce projet est destiné aux débutants, il met en lumière la complexité et la rigueur nécessaires pour traiter des données, concevoir une base vectorielle, et ajuster des modèles d'IA pour fournir des réponses pertinentes.

Ce travail m'a également permis de réaliser l'importance des paramètres tels que la température, le top-p, et les max tokens dans l'affinement des résultats générés par un modèle LLM. À travers des essais, des erreurs et des ajustements, j'ai compris que ces outils ne sont pas "magiques" : ils nécessitent un réel travail de réflexion pour être bien utilisés.

En tant qu'étudiante, ce projet m'a donné une première vision concrète de la manière dont des systèmes comme ChatGPT ou d'autres assistants intelligents fonctionnent au quotidien. On les utilise souvent sans trop se poser de questions, mais construire même une version simplifiée de ces outils a été extrêmement gratifiant. J'ai aussi appris qu'il est important de rester patient et curieux, surtout lorsqu'on s'attaque à un domaine nouveau et parfois intimidant.