Intelligence Artificielle et Interface Homme machine

Session4 : Exercice pratique. Partie 1

aivancity
PARIS-CACHAN

Session4 : Exercice pratique. Partie 1

Sommaire

Objectif

Mettre en oeuvre les principes de conceptions abordés dans les sessions précédentes pour mettre en place une application intelligentes d'extractions d'informations.

- Diagramme des cas d'utilisation
- Conception UX (Maquettage, Persona)
- Analyse des risque.
- Mis en place de cas de test et matrice RTM.
- Implémentation.
- Validation des exigences

Le rendu final de l'application comptera pour la note de contrôle continu et les étudiants s'organiseront par groupe de trois (03). Ce rendu sera fait sous forme de dépot git (Git repository) mis au format **.zip** et à déposer sur Blackboard.

Exigences fonctionnelles

1. L'application devra permettre de charger (upload) de type PDF ou docx.
2. L'application devra produire automatiquement un résumé des documents chargés
3. Sous forme de chat interactif, L'utilisateur pourra poser des questions relatives au(x) document(s) chargé(s).
4. A chaque réponse de chat, il sera possible d'afficher les extraits des documents pertinents, utilisés par le chatbot pour répondre à la question.(Les sources)
5. Les documents chargés devront être indexés et stockés de manière permanente sur le système de fichiers.
6. **Bonus** : Pour les documents écrits dans une langue autre que le français, l'application devra fournir un en français de la première page à l'utilisateur.

Exigences techniques

- L'application devra implémentée en python.
- L'approche RAG (Retrieval Augmentated Generation) sera utliisee pour l'enregistrement(indexation) et l'interrogation des documents) et le framework python LlamaIndex sera utilisé
- L'application utilisera l'API de OpenAI(fournie) pour l'accès aux systèmes d'IA :
  1. Modèle d'embedding **text-embedding-ada-02**
  2. Modèle de LLM **gpt-3.5-turbo** pour
- L'interface sera implémentée avec Streamlit
- Les templates devront être réalisés avec Figma

Retrieval-Augmented Generation (RAG)

Principe

Concept de base

Le RAG combine les techniques de récupération d'informations et de génération de texte pour produire des réponses plus pertinentes et contextuellement appropriées.

- **Récupération** : Identification des documents ou passages pertinents dans une base de données.
- **Augmentation** : Utilisation des informations récupérées pour enrichir le contexte de la génération de réponses.

Processus

1. Interrogation de la base de données avec une requête utilisateur.
2. Récupération des passages les plus pertinents.
3. Combinaison des passages récupérés avec la requête initiale.
4. Génération de la réponse finale en utilisant un modèle de langage avancé

Retrieval-Augmented Generation (RAG)

Principe

Concept de base

Le RAG combine les techniques de récupération d'informations et de génération de texte pour produire des réponses plus pertinentes et contextuellement appropriées.

- **Récupération** : Identification des documents ou passages pertinents dans une base de données.
- **Augmentation** : Utilisation des informations récupérées pour enrichir le contexte de la génération de réponses.

Processus

1. Interrogation de la base de données avec une requête utilisateur.
2. Récupération des passages les plus pertinents.
3. Combinaison des passages récupérés avec la requête initiale.
4. Génération de la réponse finale en utilisant un modèle de langage avancé

Retrieval-Augmented Generation (RAG)

Principe

Concept de base

Le RAG combine les techniques de récupération d'informations et de génération de texte pour produire des réponses plus pertinentes et contextuellement appropriées.

- **Récupération** : Identification des documents ou passages pertinents dans une base de données.
- **Augmentation** : Utilisation des informations récupérées pour enrichir le contexte de la génération de réponses.

Processus

1. Interrogation de la base de données avec une requête utilisateur.
2. Récupération des passages les plus pertinents.
3. Combinaison des passages récupérés avec la requête initiale.
4. Génération de la réponse finale en utilisant un modèle de langage avancé

Retrieval-Augmented Generation (RAG)

Principe

Concept de base

Le RAG combine les techniques de récupération d'informations et de génération de texte pour produire des réponses plus pertinentes et contextuellement appropriées.

- **Récupération** : Identification des documents ou passages pertinents dans une base de données.
- **Augmentation** : Utilisation des informations récupérées pour enrichir le contexte de la génération de réponses.

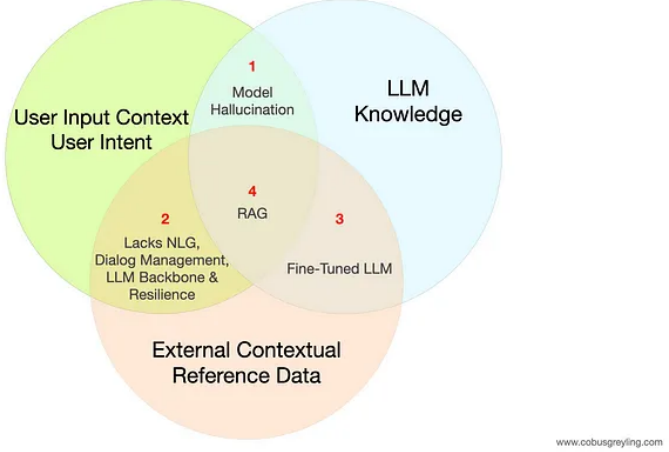
Processus

1. Interrogation de la base de données avec une requête utilisateur.
2. Récupération des passages les plus pertinents.
3. Combinaison des passages récupérés avec la requête initiale.
4. Génération de la réponse finale en utilisant un modèle de langage avancé (LLM).

Retrieval-Augmented Generation RAG

Avantages

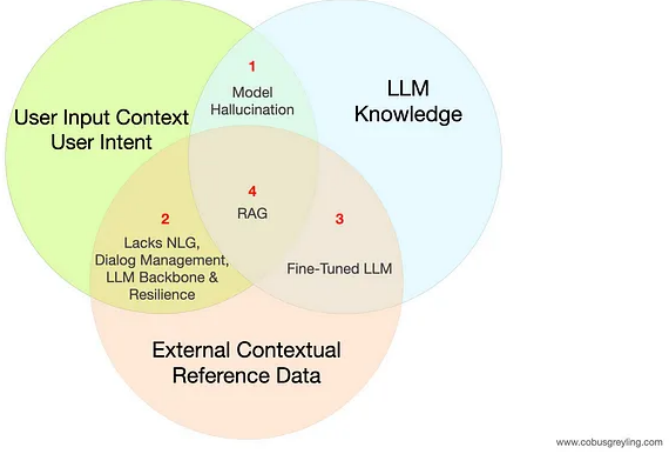
- Amélioration de la pertinence des réponses.
- Réduction des hallucinations génératives.
- Meilleure exploitation des données disponibles.



Retrieval-Augmented Generation RAG

Avantages

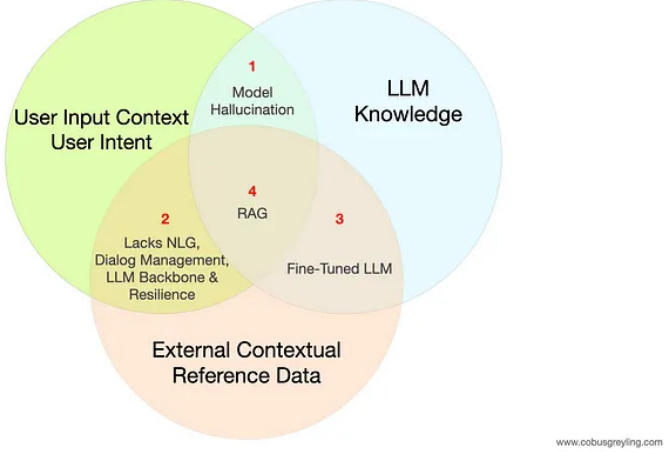
- Amélioration de la pertinence des réponses.
- Réduction des hallucinations génératives.
- Meilleure exploitation des données disponibles.



Retrieval-Augmented Generation RAG

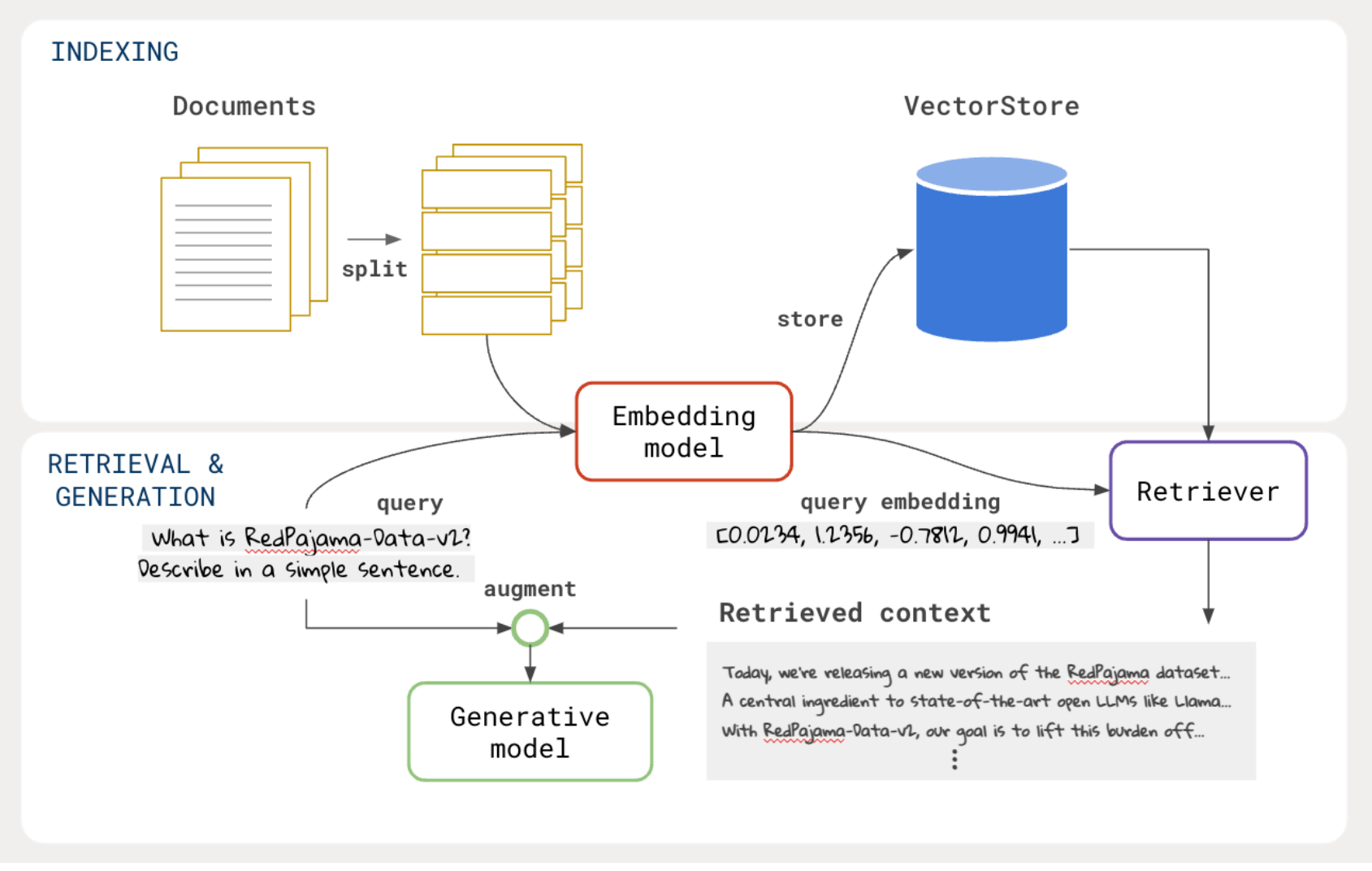
Avantages

- Amélioration de la pertinence des réponses.
- Réduction des hallucinations génératives.
- Meilleure exploitation des données disponibles.



Retrieval-Augmented Generation RAG

Etapes



Création de la base de données

Étape 1

- **Indexation** ou découpage du document en unités d'informations appelées chunks (Phrases ou paragraphes).
- Création de vecteur embeddings encodant le sens de chacun des chunks (Des phrases sémantiquement proches ont des vecteurs de représentation proches dans l'espace d'embedding.
- Utilisation d'un modèle d'embedding (transformeur encodeur) pour construire les vecteurs.
- Stockage persistent des index pour exploitation ultérieure.

Création de la base de données

Étape 1

- **Indexation** ou découpage du document en unités d'informations appelées chunks (Phrases ou paragraphes).
- Création de vecteur embeddings encodant le sens de chacun des chunks (Des phrases sémantiquement proches ont des vecteurs de représentation proches dans l'espace d'embedding.
- Utilisation d'un modèle d'embedding (transformeur encodeur) pour construire les vecteurs.
- Stockage persistent des index pour exploitation ultérieure.

Création de la base de données

Étape 1

- **Indexation** ou découpage du document en unités d'informations appelées chunks (Phrases ou paragraphes).
- Création de vecteur embeddings encodant le sens de chacun des chunks (Des phrases sémantiquement proches ont des vecteurs de représentation proches dans l'espace d'embedding.
- Utilisation d'un modèle d'embedding (transformeur encodeur) pour construire les vecteurs.
- Stockage persistent des index pour exploitation ultérieure.

Création de la base de données

Étape 1

- **Indexation** ou découpage du document en unités d'informations appelées chunks (Phrases ou paragraphes).
- Création de vecteur embeddings encodant le sens de chacun des chunks (Des phrases sémantiquement proches ont des vecteurs de représentation proches dans l'espace d'embedding.
- Utilisation d'un modèle d'embedding (transformeur encodeur) pour construire les vecteurs.
- Stockage persistent des index pour exploitation ultérieure.

Récupération des Informations

Étape 2

- La première étape du processus RAG consiste à interroger une base de données ou une collection de documents.
- L'objectif est d'identifier les passages ou documents les plus pertinents en réponse à une requête utilisateur.
- Cette étape utilise des techniques avancées de recherche d'informations pour optimiser la pertinence.

Exemple

- Recherche de publications scientifiques pertinentes pour une question de recherche.
- Identification des sections de manuels techniques en réponse à une question spécifique.

Récupération des Informations

Étape 2

- La première étape du processus RAG consiste à interroger une base de données ou une collection de documents.
- L'objectif est d'identifier les passages ou documents les plus pertinents en réponse à une requête utilisateur.
- Cette étape utilise des techniques avancées de recherche d'informations pour optimiser la pertinence.

Exemple

- Recherche de publications scientifiques pertinentes pour une question de recherche.
- Identification des sections de manuels techniques en réponse à une question spécifique.

Récupération des Informations

Étape 2

- La première étape du processus RAG consiste à interroger une base de données ou une collection de documents.
- L'objectif est d'identifier les passages ou documents les plus pertinents en réponse à une requête utilisateur.
- Cette étape utilise des techniques avancées de recherche d'informations pour optimiser la pertinence.

Exemple

- Recherche de publications scientifiques pertinentes pour une question de recherche.
- Identification des sections de manuels techniques en réponse à une question spécifique.

Augmentation du Contexte

Étape 3

- Les informations récupérées sont utilisées pour enrichir le contexte de la génération de réponses.
- Cette étape permet d'intégrer des faits et des données spécifiques dans le processus de génération.
- L'objectif est d'améliorer la qualité et la pertinence des réponses générées.

Exemple

- Intégration de résultats de recherche dans un résumé généré.
- Utilisation de données financières récupérées pour générer des analyses de marché.

Augmentation du Contexte

Étape 3

- Les informations récupérées sont utilisées pour enrichir le contexte de la génération de réponses.
- Cette étape permet d'intégrer des faits et des données spécifiques dans le processus de génération.
- L'objectif est d'améliorer la qualité et la pertinence des réponses générées.

Exemple

- Intégration de résultats de recherche dans un résumé généré.
- Utilisation de données financières récupérées pour générer des analyses de marché.

Augmentation du Contexte

Étape 3

- Les informations récupérées sont utilisées pour enrichir le contexte de la génération de réponses.
- Cette étape permet d'intégrer des faits et des données spécifiques dans le processus de génération.
- L'objectif est d'améliorer la qualité et la pertinence des réponses générées.

Exemple

- Intégration de résultats de recherche dans un résumé généré.
- Utilisation de données financières récupérées pour générer des analyses de marché.

Génération de Réponses

Étape 4

- La dernière étape du processus RAG consiste à générer des réponses textuelles.
- Un modèle de langage avancé utilise le contexte enrichi pour produire une réponse pertinente et cohérente.
- Cette approche permet de combiner la précision des données récupérées avec la flexibilité de la génération de texte.

Génération de Réponses

Étape 4

- La dernière étape du processus RAG consiste à générer des réponses textuelles.
- Un modèle de langage avancé utilise le contexte enrichi pour produire une réponse pertinente et cohérente.
- Cette approche permet de combiner la précision des données récupérées avec la flexibilité de la génération de texte.

Génération de Réponses

Étape 4

- La dernière étape du processus RAG consiste à générer des réponses textuelles.
- Un modèle de langage avancé utilise le contexte enrichi pour produire une réponse pertinente et cohérente.
- Cette approche permet de combiner la précision des données récupérées avec la flexibilité de la génération de texte.

Streamlit

https://docs.streamlit.io/get-started/fundamentals/main-concepts

Merci !