

ECOLE CENTRALE CASABLANCA



TP

Retrieval Augmented Generation (RAG)

Réalisation et Rapport :

Ilyas OUHNINE

Fadoua BEN KABOUR

Marwane KASSA

Table des matières

1	Introduction	2
2	Configuration et Préparation de l'Environnement	2
2.1	Vérification de l'environnement	2
2.2	Chargement des variables d'environnement	2
3	Indexation des Documents	3
3.1	Ajout et indexation des fichiers	3
4	Recherche et Récupération d'Informations	5
5	Système de Question-Réponse	5
6	Évaluation des Performances	5
7	Configuration et Paramétrage du Modèle	6
7.1	Configuration du modèle LLM	6
7.2	Optimisation de l'indexation	6
7.3	Optimisation de la récupération des documents	7
8	Interface Utilisateur et Expérience	7
9	Conclusion et Perspectives	8

1 Introduction

Le projet RAG (Retrieval-Augmented Generation) vise à développer un système d'intelligence artificielle capable de récupérer et de générer des réponses précises à partir d'un corpus documentaire. Ce système repose sur la combinaison de l'indexation vectorielle et de la génération de texte via un grand modèle de langage (LLM).

Les technologies clés employées incluent :

- **Azure AI Inference** : pour l'inférence des embeddings et la génération des réponses.
- **LangChain** : pour la gestion des requêtes et des interactions avec l'IA.
- **ChromaDB** : base de données vectorielle performante permettant un accès rapide aux documents pertinents.
- **Flask** : framework léger permettant une interaction utilisateur via une interface web intuitive.

L'objectif principal du projet est d'améliorer la recherche documentaire en combinant des techniques avancées de recherche sémantique et de génération de texte.

2 Configuration et Préparation de l'Environnement

Avant d'exécuter le système, une vérification de l'environnement est nécessaire.

2.1 Vérification de l'environnement

La vérification de l'environnement est effectuée avec la commande suivante :

```
python check_environment.py --env
```

Cette commande permet de s'assurer que toutes les variables d'environnement essentielles (GITHUB_TOKEN, AZURE_INFERENCE_ENDPOINT, etc.) sont bien définies.

```
===== RAG System Environment Check =====

✓ GITHUB_TOKEN is set: ton_****_ici
✓ AZURE_INFERENCE_ENDPOINT is set: https://models.inference.ai.azure.com
✓ AZURE_INFERENCE_CHAT_MODEL is set: gpt-4o
✓ AZURE_INFERENCE_EMBEDDING_MODEL is set: text-embedding-3-small
! AZURE_OPENAI_API_KEY is not set (optional)
! AZURE_OPENAI_ENDPOINT is not set (optional)
! AZURE_OPENAI_DEPLOYMENT_NAME is not set (optional)
✓ Azure AI Inference SDK is installed
✓ LangChain is installed (version: 0.3.20)

===== Testing Azure AI Inference Connection =====

X Failed to connect to Azure AI Inference API: (unauthorized) Bad credentials
Code: unauthorized
Message: Bad credentials

===== Environment Check Complete =====
```

FIGURE 1 – Vérification de l'environnement

2.2 Chargement des variables d'environnement

Le script suivant est utilisé pour charger les variables depuis le fichier `.env` :

```
python load_env.py
```

Si toutes les clés API et URL sont correctement configurées, le système peut être lancé.

```
(venv) PS C:\Users\hp\Downloads\NLP-ECC\NLP-ECC> python load_env.py
>>
GitHub token loaded (length: 40)
Azure Inference endpoint loaded: https://models.inference.ai.azure.com
Chat model loaded: gpt-4o
Embedding model loaded: text-embedding-3-small

Environment variables loaded successfully.
You can now run the RAG system.
```

FIGURE 2 – Chargement des variables d'environnement

3 Indexation des Documents

Le système RAG indexe les documents afin de permettre une recherche efficace et rapide.

3.1 Ajout et indexation des fichiers

L'interface permet d'ajouter des fichiers et de les indexer via l'option **Index Documents**.

Follow these steps to make documents searchable:

1. Upload files to the data directory
2. Index the files to make them searchable

Step 1: Upload Files

Drag & Drop Files Here

Or click to select files

Supported formats: PDF, TXT, DOCX, CSV, MD

Upload File to Data Directory

Upload & Index Selected File

Files in Data Directory:

File Name	Type	Size	Indexable	Actions
Attestation sur l'honneur.pdf	pdf	285.35 KB	Yes	Delete

Refresh File List

FIGURE 3 – Ajout de fichiers pour l'indexation

Une fois les fichiers ajoutés, l'indexation est déclenchée.

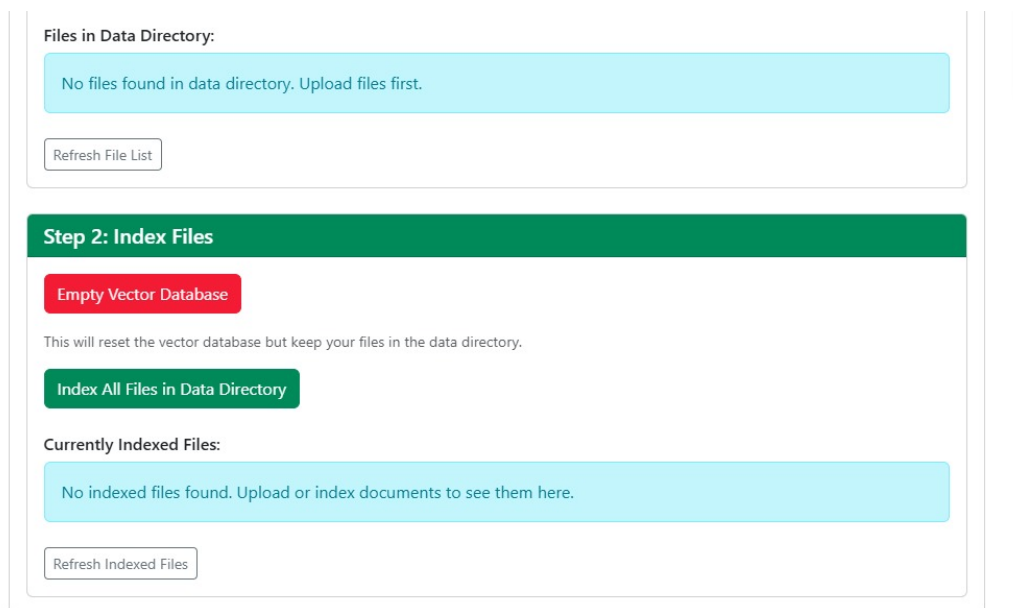


FIGURE 4 – Lancement de l'indexation des fichiers

Le système exécute l'indexation en plusieurs étapes :

- Chargement des fichiers (*loading*).
- Découpage en segments (*chunking*).
- Calcul des embeddings.
- Stockage dans la base vectorielle.

`python cli.py index`

Une confirmation d'indexation s'affiche à l'écran.

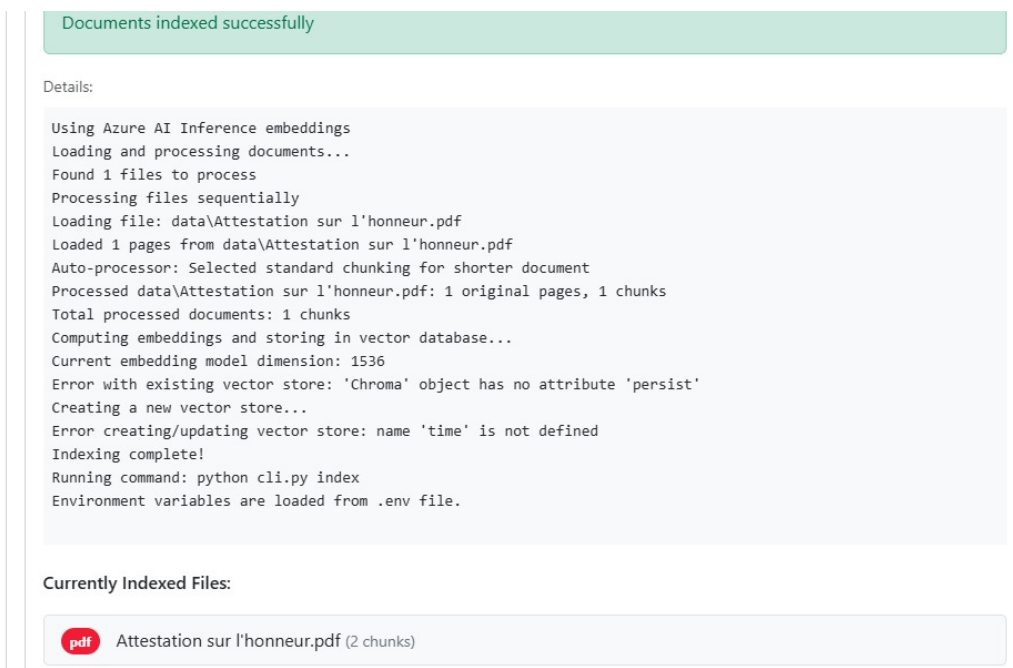


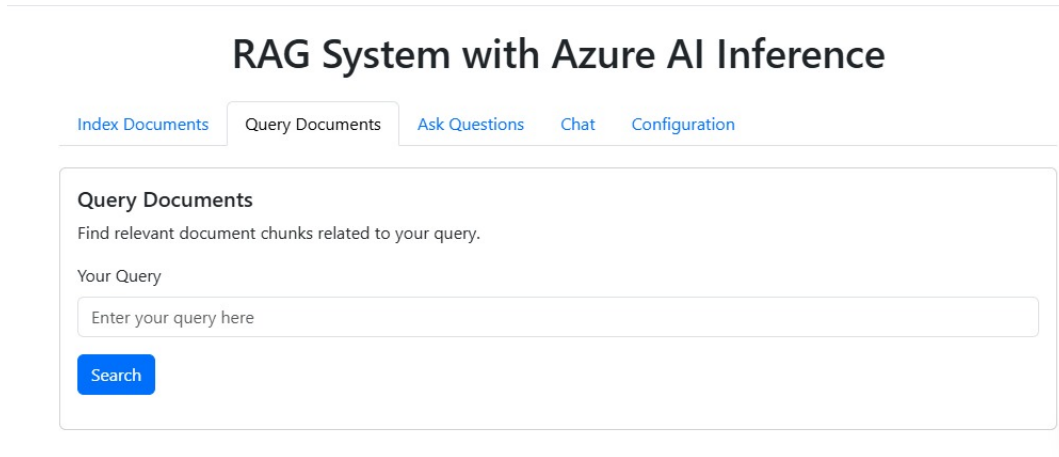
FIGURE 5 – Processus d'indexation des documents

4 Recherche et Récupération d'Informations

L'utilisateur peut rechercher des documents via :

```
python cli.py query
```

L'interface web permet également d'effectuer des requêtes.



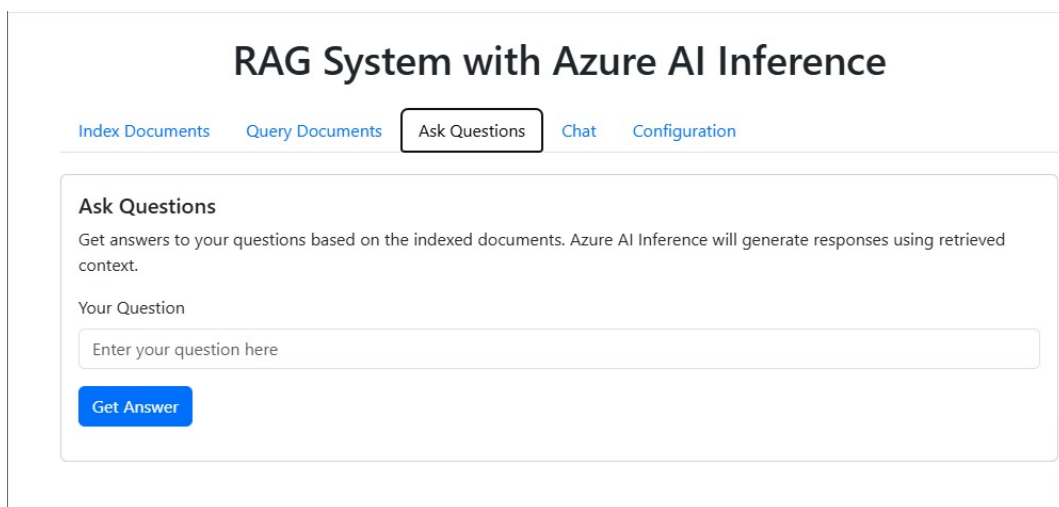
The screenshot shows a web interface titled "RAG System with Azure AI Inference". It has a navigation bar with five tabs: "Index Documents", "Query Documents", "Ask Questions", "Chat", and "Configuration". The "Query Documents" tab is currently selected. Below the navigation bar, there is a section titled "Query Documents" with the instruction "Find relevant document chunks related to your query." Below this, there is a label "Your Query" followed by a text input field containing the placeholder text "Enter your query here". At the bottom of this section is a blue button labeled "Search".

FIGURE 6 – Interface de recherche des documents indexés

5 Système de Question-Réponse

Le système intègre un modèle LLM (*GPT-4o*) qui génère des réponses à partir des documents indexés.

```
python cli.py ask
```



The screenshot shows a web interface titled "RAG System with Azure AI Inference". It has a navigation bar with five tabs: "Index Documents", "Query Documents", "Ask Questions", "Chat", and "Configuration". The "Ask Questions" tab is currently selected. Below the navigation bar, there is a section titled "Ask Questions" with the instruction "Get answers to your questions based on the indexed documents. Azure AI Inference will generate responses using retrieved context." Below this, there is a label "Your Question" followed by a text input field containing the placeholder text "Enter your question here". At the bottom of this section is a blue button labeled "Get Answer".

FIGURE 7 – Interface web pour poser des questions

6 Évaluation des Performances

L'évaluation du système repose sur plusieurs métriques :

- **Précision des réponses** : mesurée en comparant les réponses du LLM à des réponses attendues.

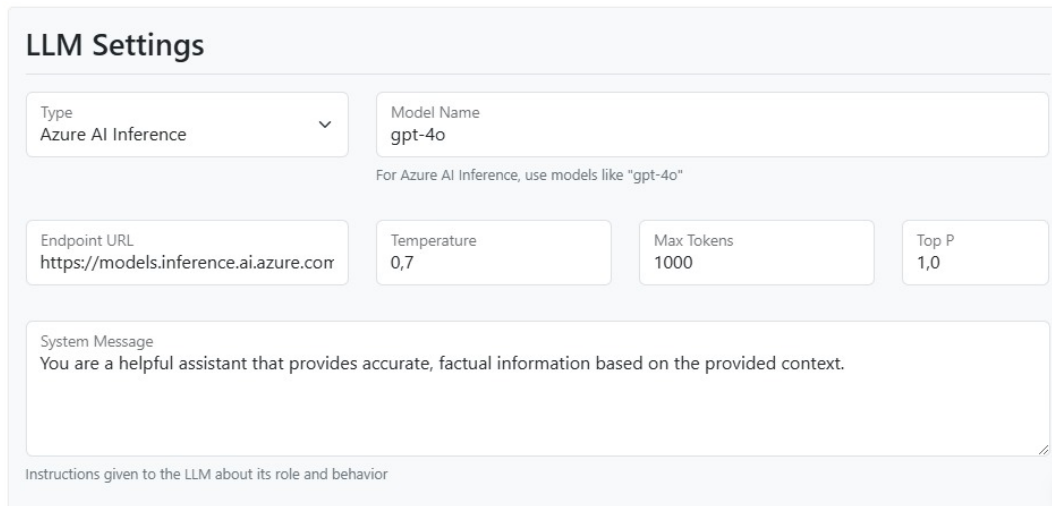
- **Temps d'exécution** : analyse du temps moyen de récupération d'un document.
- **Score de similarité** : basé sur la distance cosinus entre la requête et les documents indexés.

`python cli.py evaluate`

7 Configuration et Paramétrage du Modèle

7.1 Configuration du modèle LLM

Le système repose sur **Azure AI Inference** pour exécuter les requêtes sur un LLM.

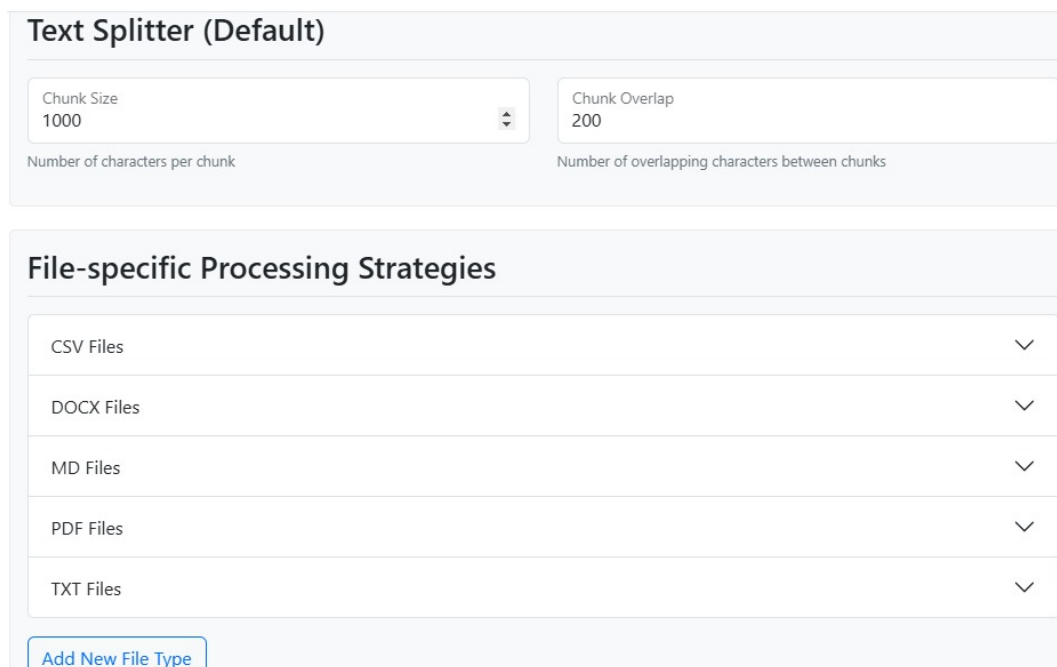


The screenshot shows the 'LLM Settings' configuration panel. It includes a dropdown for 'Type' set to 'Azure AI Inference', a text field for 'Model Name' set to 'gpt-4o', and a note: 'For Azure AI Inference, use models like "gpt-4o"'. Below these are four input fields: 'Endpoint URL' (https://models.inference.ai.azure.com), 'Temperature' (0,7), 'Max Tokens' (1000), and 'Top P' (1,0). A large text area for 'System Message' contains the text: 'You are a helpful assistant that provides accurate, factual information based on the provided context.' At the bottom, a small note reads: 'Instructions given to the LLM about its role and behavior'.

FIGURE 8 – Paramétrage du modèle LLM

7.2 Optimisation de l'indexation

Un découpage efficace des documents améliore la précision des résultats.



The screenshot displays two configuration panels. The top panel, 'Text Splitter (Default)', has two input fields: 'Chunk Size' (1000) with a note 'Number of characters per chunk', and 'Chunk Overlap' (200) with a note 'Number of overlapping characters between chunks'. The bottom panel, 'File-specific Processing Strategies', is a table with five rows for file types: CSV Files, DOCX Files, MD Files, PDF Files, and TXT Files, each with a dropdown arrow. At the bottom of this panel is a button labeled 'Add New File Type'.

FIGURE 9 – Configuration du découpage des documents

7.3 Optimisation de la récupération des documents

L'activation du traitement parallèle accélère les requêtes.

The image shows two configuration panels. The top panel, titled 'Parallel Processing', contains a toggle switch labeled 'Enable Parallel Processing' which is turned on, and a text input field for 'Max Workers' with the value '4'. Below the input field is the text 'Number of parallel processing threads'. The bottom panel, titled 'Retrieval Settings', contains a text input field for 'Top K Results' with the value '5'. Below this field is the text 'Number of most relevant document chunks to retrieve'.

FIGURE 10 – Traitement parallèle et réglages de la récupération

8 Interface Utilisateur et Expérience

L'interface web fournit :

- Une section **Index Documents** (ajout et indexation de fichiers).
- Un module de **Recherche de documents** (requêtes sur la base vectorielle).
- Un espace dédié au **Chatbot**.

RAG System Configuration

The image shows the 'RAG System Configuration' interface. At the top, there are four buttons: 'Back to Home' (grey), 'Go to Document Index' (blue), 'Restore Defaults' (green), and 'Reset Changes' (red). Below these buttons are two configuration sections. The first section, 'Data Paths', has two text input fields: 'Data Directory' with the value 'data' and 'Vector DB Directory' with the value 'vector_db_1742166892'. Below each field is a small explanatory text: 'Directory where document files are stored' and 'Directory where vector database is stored'. The second section, 'Embedding Model', has a dropdown menu for 'Type' and a text input field for 'Model Name'. Below the 'Model Name' field is a small note: 'For HuggingFace, use model ID like "sentence-transformers/all-mpnet-base-v2"'. At the bottom left of the 'Embedding Model' section are two buttons: 'Save Configuration' (blue) and 'Cancel' (white with a grey border).

FIGURE 11 – Interface de configuration du système

9 Conclusion et Perspectives

Le système RAG développé permet une récupération efficace des documents indexés tout en offrant des capacités avancées de génération de texte.

Améliorations futures :

- Optimisation du temps de requête et d'indexation.
- Intégration de nouveaux modèles LLM.
- Ajout d'un module de feedback utilisateur pour améliorer les résultats.