

Présentation du système

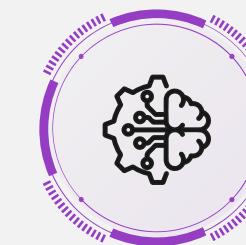
Présenter par :
Imrane



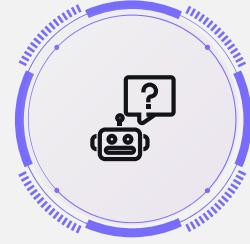
Sommaire



I. Introduction



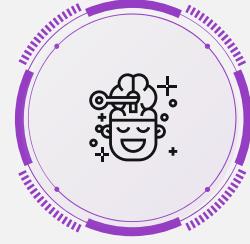
II. Pré-traitement et découpage des supports



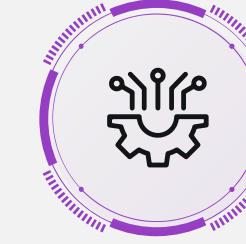
III. Vectorisation et indexation



IV. Formatage du prompt et modèle de langue utiliser



V. Exemple



VI. REX et Conclusion



I. Introduction

- Objectif : Crée un agent conversationnel qui répond à des questions d'un étudiant de première année en BUT SD
- Document à disposition
- Selon le principe du RAG (Retrieval Augmented Generation)



II. Pré-traitement et découpage des supports



Traitement python

Nous avons transformé chaque type de fichier (pdf, csv, ipynb,...) en texte.



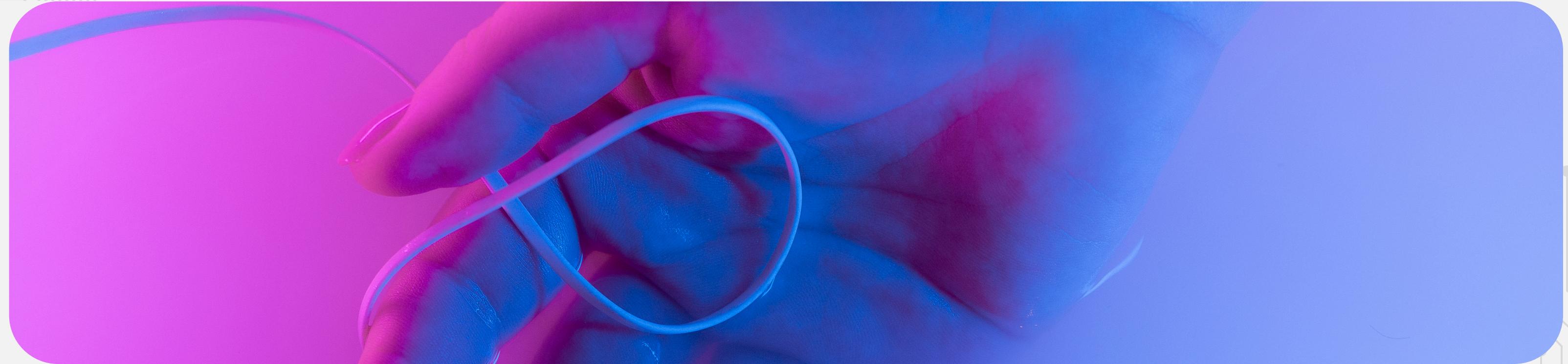
Création du document json

Le Script python stock tout les fichiers convertis en texte dans un document json.



Création des Chunks

Segmenter les documents en chunks (500 caractères)



III. Vectorisation et indexation



Vectorisation

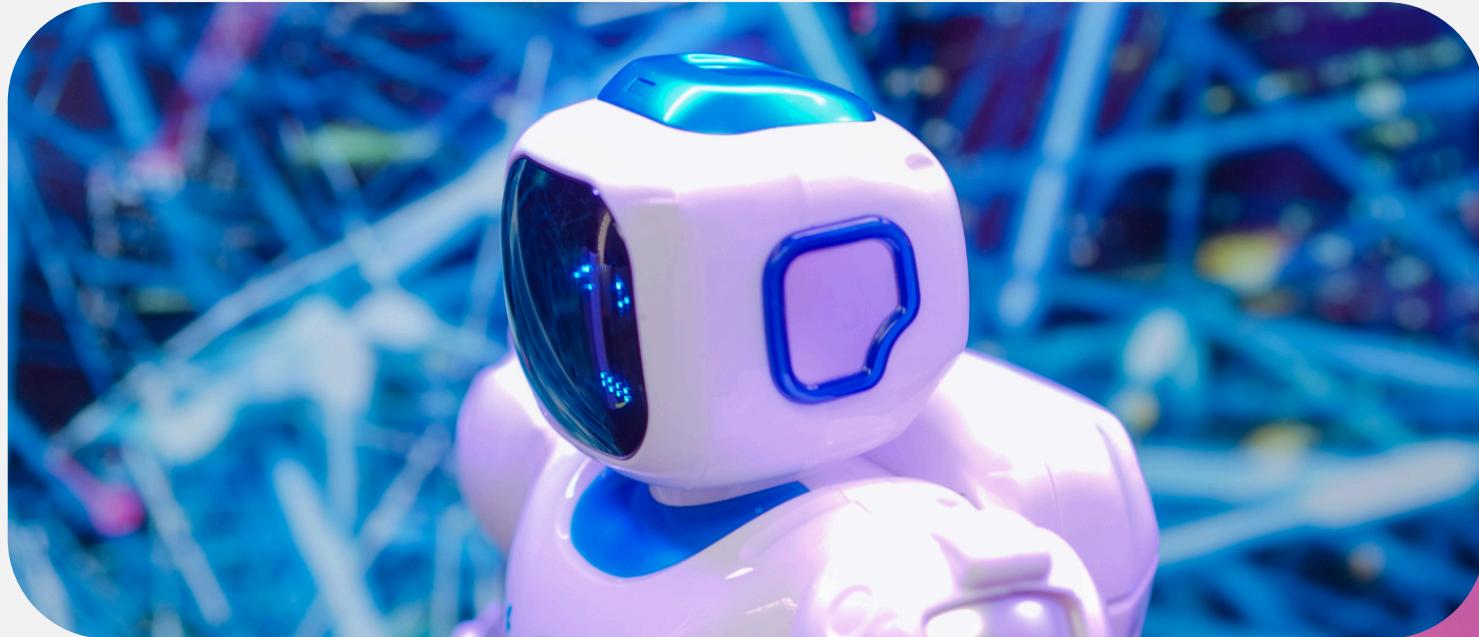
Transformer chaque chunk de texte en vecteur numérique (embedding) qui représente son sens



Indexation

Stocker les vecteurs dans une base de données vectorielle pour pouvoir faire des recherches par similarité (RAG)

IV. Formatage du prompt et modèle de langue utiliser



Modèle de langue utiliser

Nous avons utiliser le modèle Mistral 7b 0.3, qui est un modèle qui demande d'avoir un Token.

Performance du modèle et raisons de ce choix.

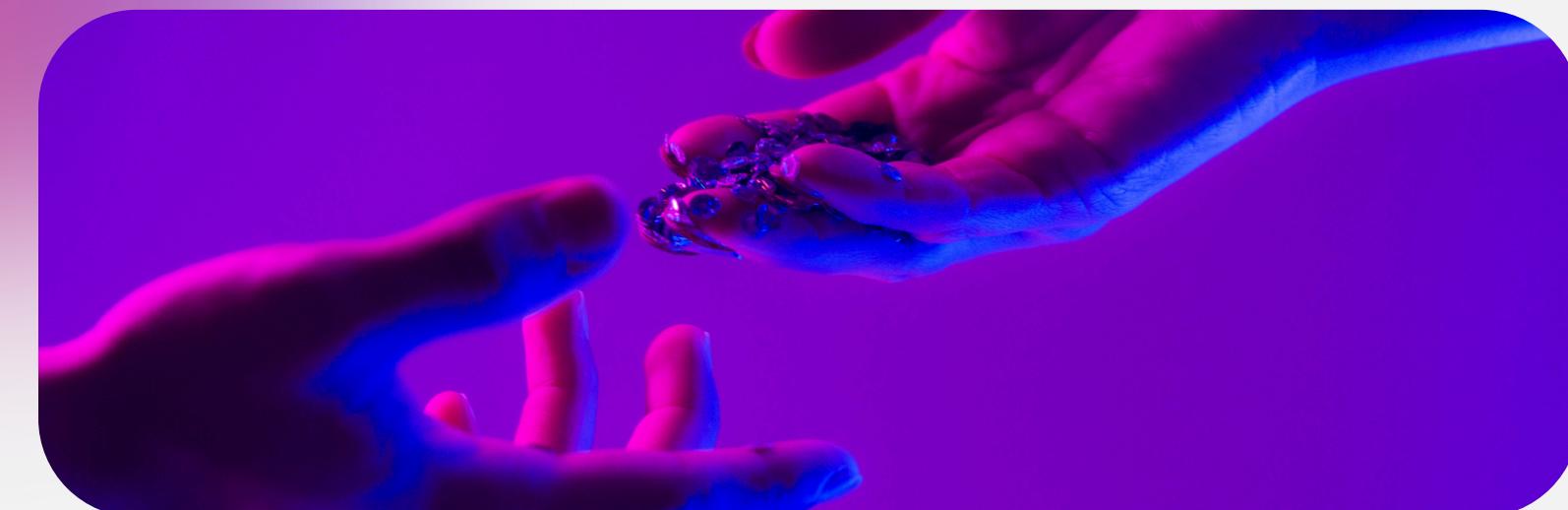
Formatage du prompt

Voici notre prompt :

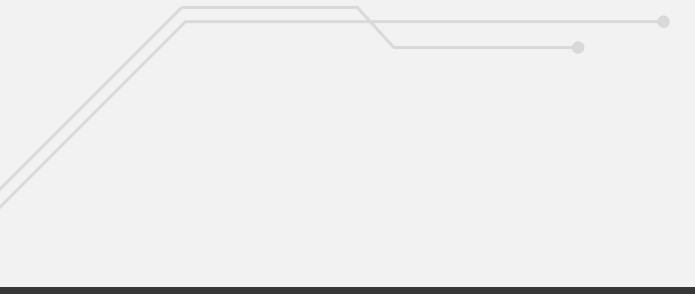
"Tu es un assistant pédagogique en Python. Voici des extraits de cours sur lesquelles tu te base : + {contexte}" +

"Voici la question de l'utilisateur : + {question}" +

"Réponds de manière claire, concise et pédagogique comme si tu étais face à un débutant. Commence ta réponse par : 'Voici la réponse à ta question'"



V. Exemple



```
Question : Comment créer une liste en Python ?  
Setting `pad_token_id` to `eos_token_id`:2 for open-end generation.
```

Réponse générée :

Voici la réponse à ta question :

Pour créer une liste en Python, tu peux utiliser des crochets `[]` autour des éléments séparés par des virgules. Par exemple, pour créer une liste de noms, tu peux écrire :

```
```python  
nom_de_la_liste = ['Alice', 'Bob', 'Charlie']
```
```

Tu peux également créer une liste vide en utilisant les crochets vides :

```
```python  
une_liste_vide = []
```
```

Tu peux également ajouter des éléments à une liste existante en utilisant l'opérateur `append()`. Par exemple :

```
```python  
une_liste = ['Alice']
une_liste.append('Bob')
print(une_liste) # ['Alice', 'Bob']
```
```

Tu peux également accéder aux éléments de la liste en utilisant des indices. Les indices commencent à 0. Par exemple :

```
```python  
une_liste = ['Alice', 'Bob', 'Charlie']
print(une_liste[0]) # Alice
print(une_liste[
```

## Explication

- Paramètre utilisé pour cette réponse (k=5 et 300 premiers tokens de chaque chunks)
- Question
- Réponse



# V. Exemple

```
Toi : Comment créer une liste en Python ?

Question : Comment créer une liste en Python ?
Setting `pad_token_id` to `eos_token_id`:2 for open-end generation.

Réponse générée :

Voici la réponse à ta question :

'Voici la réponse à ta question : Créer une liste en Python se fait en utilisant des crochets []. Voici un exemple :

```python  
liste_exemple = [1, "hello", 3.14, False]  
```  

Dans cette liste, on a une variable de type entier, une chaîne de caractères, un nombre à virgule flottante, et une variable booléenne. Les éléments sont séparés par des virgules.
=====
```

## Explication

- Paramètre utiliser pour cette réponse (k=3 et 500 premier token de chaque chunks)
- Question
- Réponse

# VI. REX et Conclusion

- REX de chacun
- Apprentissage
- Conclusion

Merci pour votre  
attention !

Questions/Réponses

