



---

## Embeddings et Services Web API : Récupération Augmentée pour Répondre à des Questions sur des Données Textuelles

### 1. Contexte Générale du TP

#### Description générale du projet

Dans ce projet, les étudiants implémenteront un système de Récupération Augmentée (Retrieval-Augmented Generation, RAG) sous forme de **services Web** en utilisant FastAPI. Ils vont encoder des documents textuels (articles ou tweets) et des requêtes en embeddings pour permettre une recherche efficace. Le projet se concentrera sur deux tâches principales, chacune implémentée sous forme de service API :

1. **Encodage des documents** : Charger des articles ou tweets, les encoder en embeddings via un modèle de langage pré-entraîné, et exposer ces embeddings via une API.
2. **Recherche de documents** : Implémenter un service de recherche pour une requête utilisateur, en calculant la similarité cosinus entre l'embedding de la requête et ceux des documents pour renvoyer les plus pertinents.

---

#### Détails des API à implémenter

Le projet se décompose en deux principaux services web API que les étudiants devront construire :

##### 1. Service d'encodage des documents

- **Route** : POST /encode\_documents
- **Description** : Ce service reçoit un ensemble de documents (tweets ou articles de loi) sous forme de texte, les encode en vecteurs d'embeddings via un modèle (par exemple, BERT ou SBERT), puis stocke ces vecteurs dans un fichier JSON pour être réutilisés dans la recherche.
- **Étapes de la route** :
  - Charger les documents envoyés via la requête.
  - Encoder chaque document en utilisant un modèle d'embeddings.

- Stocker les embeddings dans un fichier JSON (où chaque entrée contient l’ID du document, son texte et son embedding).
- **Exemple de requête :**

```
{
  "documents": [
    "Article 1 : Toute personne a droit au respect de sa vie privée...",
    "Le climat change, il est temps d'agir.",
    ...
  ]
}
```

- **Exemple de réponse :**

```
{
  "status": "success",
  "message": "Documents encoded and stored successfully"
}
```

## 2. Service de recherche de documents

- **Route :** POST /search\_documents
- **Description :** Ce service permet de saisir une requête, l'encode en un vecteur, puis calcule la similarité cosinus entre ce vecteur de requête et ceux des documents encodés pour renvoyer les documents les plus pertinents.
- **Étapes de la route :**
  - Recevoir la requête utilisateur.
  - Encoder la requête avec le même modèle que celui utilisé pour les documents.
  - Charger les embeddings des documents depuis le fichier JSON.
  - Calculer la similarité cosinus entre l’embedding de la requête et ceux des documents.
  - Retourner les documents les plus proches.
- **Exemple de requête :**

```
{
  "query": "Droit à la vie privée"
}
```

- **Exemple de réponse :**

```
{
  "relevant_documents": [
```

```
{
  "id": 1,
  "text": "Article 1 : Toute personne a droit au respect de sa vie privée...",
  "similarity_score": 0.89
},
...
]
```

---

### Instructions Techniques

Les étudiants devront :

1. **Choisir un modèle d'embedding** : Utiliser un modèle pré-entraîné comme ceux disponibles via sentence-transformers, par exemple paraphrase-MiniLM-L6-v2.
2. **Structure des données** : Charger des documents (articles de loi ou tweets) envoyés via des requêtes, les encoder, et les stocker dans un fichier JSON où chaque entrée contiendra :
  - L'identifiant du document,
  - Le texte du document,
  - Le vecteur d'embedding correspondant.
3. **Service de Recherche et Similarité** :
  - Le service de recherche doit encoder la requête de l'utilisateur avec le même modèle d'embedding.
  - Calculer la similarité cosinus avec les vecteurs d'embeddings des documents.
  - Retourner les documents les plus pertinents avec leur score de similarité.
4. **Utilisation des bibliothèques Numpy et JSON** :
  - Utiliser **Numpy** pour manipuler les vecteurs et calculer la similarité cosinus.
  - Stocker et lire les embeddings des documents depuis un fichier JSON.
5. **Déploiement d'une API en ligne de commande** :
  - Les étudiants doivent fournir une interface de commande simple pour permettre à l'utilisateur d'interagir avec les services API, notamment en saisissant une requête utilisateur pour la recherche.

## Exigences Supplémentaires

### 1. Optimisation de l'algorithme :

- Assurer que l'API de recherche soit performante même en cas d'un grand nombre de documents (e.g., en optimisant les calculs avec Numpy).

### 2. Documentation API :

- Inclure une documentation minimale via FastAPI pour chaque route, permettant d'afficher les détails de chaque endpoint.

## Jeu de Données et Requêtes

Les étudiants peuvent utiliser des fichiers d'articles ou de tweets. Exemple de données possibles :

- **Exemple d'articles** (fichier articles\_law.txt) :

Article 1 : Toute personne a droit au respect de sa vie privée et familiale, de son domicile et de sa correspondance.  
Article 2 : Nul ne peut être arbitrairement détenu ni emprisonné.

- **Exemple de tweets** (fichier tweets.txt) :

Je suis très heureux de participer à cette conférence !  
Le climat change, il est temps d'agir.

## Exemple de Requêtes pour Test

- "Qu'est-ce qu'un procès équitable ?"
- "Les droits de l'homme en France"
- "Changement climatique"
- "Droit à la vie privée"

## Annexe :

### Option 1: Jeu de Données des Tweets

Vous pouvez créer un jeu de données synthétiques de tweets sur différents sujets. Chaque ligne du fichier représente un tweet. Voici un exemple :

#### Contenu du fichier tweets.txt

```
Je suis tellement content de mon nouveau téléphone ! #tech
Le match de ce soir était incroyable ! #football
J'adore cette nouvelle série sur Netflix. #divertissement
Il pleut encore aujourd'hui, quel temps maussade. #météo
L'intelligence artificielle va changer le monde. #IA
Le réchauffement climatique est un problème sérieux. #environnement
Je viens de finir de lire un livre fascinant sur l'économie. #lecture
```

La conférence sur la tech d'aujourd'hui était inspirante ! #tech  
Le dernier film de science-fiction que j'ai vu était génial ! #cinéma  
Le projet sur lequel je travaille avance bien. #travail

## Option 2: Jeu de Données Législatif

Vous pourriez utiliser des extraits d'articles de lois, par exemple, des articles de la loi pénale ou civile. Chaque ligne du fichier représente un article de loi.

### Contenu du fichier `lois.txt`

Article 1: Nul ne peut être condamné pour une action ou une omission qui, au moment où elle a été commise, ne constituait pas une infraction pénale selon le droit national ou international.  
Article 2: Toute personne accusée d'une infraction pénale est présumée innocente jusqu'à ce que sa culpabilité ait été légalement établie.  
Article 3: Toute personne privée de liberté doit être traitée avec humanité et avec le respect de la dignité inhérente à la personne humaine.  
Article 4: Aucune peine privative de liberté ne peut être infligée pour inexécution d'une obligation contractuelle.  
Article 5: Toute personne a droit à un recours effectif devant une instance nationale pour les actes violant les droits fondamentaux.  
Article 6: Le droit à la vie est protégé par la loi. Nul ne peut être intentionnellement privé de la vie, sauf en exécution d'une condamnation à mort.  
Article 7: La torture et les peines ou traitements inhumains ou dégradants sont interdits en toutes circonstances.  
Article 8: Toute personne a droit au respect de sa vie privée et familiale, de son domicile et de sa correspondance.  
Article 9: Toute personne a droit à la liberté d'expression, sous réserve de ne pas porter atteinte à l'ordre public.  
Article 10: La liberté de pensée, de conscience et de religion est protégée par la loi.

## Option 3: Mélange Articles et Tweets

Vous pouvez aussi mélanger des articles de lois et des tweets dans un même fichier pour augmenter la complexité du projet. Cela mettra les étudiants face à un défi supplémentaire pour gérer des données hétérogènes.

### Contenu du fichier `donnees_melangees.txt`

Je suis tellement content de mon nouveau téléphone ! #tech  
Article 1: Nul ne peut être condamné pour une action ou une omission qui, au moment où elle a été commise, ne constituait pas une infraction pénale.  
Le match de ce soir était incroyable ! #football  
Article 3: Toute personne privée de liberté doit être traitée avec humanité et dignité.  
L'intelligence artificielle va changer le monde. #IA  
Le réchauffement climatique est un problème sérieux. #environnement  
Article 5: Toute personne a droit à un recours effectif devant une instance nationale pour les actes violant les droits fondamentaux.  
La conférence sur la tech d'aujourd'hui était inspirante ! #tech  
Le dernier film de science-fiction que j'ai vu était génial ! #cinéma