TD6: Mini-projet NLP avec Embeddings : Récupération Augmentée pour Répondre à des Questions sur des Données Textuelles

Description générale du projet :

Dans ce projet, les étudiants devront implémenter un système de Récupération Augmentée (Retrieval-Augmented Generation, RAG) en utilisant des embeddings pour encoder des documents et des requêtes. Le projet se concentrera sur deux tâches principales :

1. Encodage des articles ou des tweets à l'aide d'un modèle de langage pré-entraîné pour générer des vecteurs d'embeddings.
2. Création d'un pipeline de recherche où une requête utilisateur est encodée, comparée aux documents encodés, et renvoie les articles ou tweets les plus pertinents.

Pipeline à implémenter :

Le projet se décompose en deux grands pipelines que les étudiants devront construire :

1. **Pipeline d'encodage des documents :**
   - Charger un ensemble de documents (tweets ou articles de loi).
   - Utiliser un modèle d'embeddings open-source (par exemple, un modèle de BERT ou d'un autre modèle pré-entraîné disponible via une bibliothèque comme sentence-transformers).
   - Générer et stocker les vecteurs d'embeddings des documents dans un fichier JSON organisé.
2. **Pipeline d'encodage des requêtes et de recherche de correspondances :**
   - Permettre à l'utilisateur de saisir une requête.
   - Encoder cette requête en utilisant le même modèle d'embeddings.
   - Comparer le vecteur de la requête avec les vecteurs d'embeddings des documents.
   - Renvoyer les documents (tweets ou articles de loi) les plus proches en fonction de la similarité cosinus.

Détails Techniques :

Les étudiants doivent :

1. **Choisir un modèle d'embedding :** Ils doivent chercher et implémenter un modèle open-source pour générer les embeddings, comme ceux disponibles via sentence-transformers (ex. paraphrase-MiniLM-L6-v2 ou distilbert-base-nli-stsb-mean-tokens).
2. **Charger les données :** Les tweets ou articles de loi sont chargés depuis un fichier texte où chaque ligne contient un document (tweet ou article de loi).

3. **Stockage des embeddings :** Une fois que les embeddings des documents sont générés, ils doivent être sauvegardés sous format JSON, où chaque entrée contiendra :
    - L'index ou l'identifiant du document (ligne).
    - Le texte du document.
    - Le vecteur d'embedding correspondant.
4. **Recherche et Similarité :** Lorsqu'un utilisateur fait une requête, le script doit :
    - Encoder la requête en utilisant le même modèle d'embeddings.
    - Calculer la similarité cosinus entre l'embedding de la requête et ceux des documents.
    - Retourner les documents les plus pertinents.

Données :

Voici un exemple de jeu de données que vous pourriez utiliser :

1. Fichier d'articles de loi (exemple d'un 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.
Article 3 : Toute personne est présumée innocente jusqu'à ce que sa culpabilité soit légalement établie.
Article 4 : Toute personne a le droit à un procès équitable et public.
Article 5 : Nul ne peut être soumis à la torture ni à des traitements inhumains ou dégradants.
```

2. Fichier de tweets (exemple d'un fichier tweets.txt) :

```
Je suis très heureux de participer à cette conférence !
Le climat change, il est temps d'agir.
Les droits de l'homme doivent être protégés en toutes circonstances.
L'éducation est la clé de l'avenir de notre planète.
Nous devons tous lutter contre les inégalités sociales et économiques.
```

Jeu de Requêtes :

Voici un ensemble de requêtes possibles que les étudiants devront tester dans leur pipeline :

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

Exigences supplémentaires :

1. **Utilisation des bibliothèques Numpy et JSON :**
    - Les étudiants doivent utiliser Numpy pour la manipulation des vecteurs (embeddings) et le calcul de la similarité cosinus.
    - Les documents, embeddings, et résultats doivent être stockés et récupérés dans des fichiers JSON.

2. **Performance et Efficacité :**
    - L'algorithme doit être optimisé pour parcourir et comparer les vecteurs de manière efficace (ex. via des techniques comme l'utilisation de matrices avec Numpy).
3. **Interface utilisateur :**
    - Les étudiants doivent prévoir une interface simple en ligne de commande où l'utilisateur peut saisir une requête et obtenir une réponse.

Exemple de sortie attendu :

Si l'utilisateur saisit la requête : "Droit à la vie privée", le système renverrait une réponse similaire à :

```
Documents pertinents trouvés :

1. Article 1 : Toute personne a droit au respect de sa vie privée et familiale, de son domicile et de sa correspondance.

Score de similarité : 0.89
```

Structure du fichier JSON (pour les documents et leurs embeddings) :

Le fichier JSON pourrait ressembler à ceci :

```
[
  {
    "id": 1,
    "text": "Article 1 : Toute personne a droit au respect de sa vie privée et familiale, de son domicile et de sa correspondance.",
    "embedding": [0.21, 0.34, 0.56, ...]
  },
  {
    "id": 2,
    "text": "Article 2 : Nul ne peut être arbitrairement détenu ni emprisonné.",
    "embedding": [0.11, 0.24, 0.66, ...]
  }
]
```

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
```

Jeu de Requêtes

Voici quelques exemples de requêtes que vous pourriez proposer aux étudiants. Ils doivent être formulés sous forme de questions ou de phrases à rechercher dans les données.

Exemples de requêtes

1. **Quels sont les derniers films recommandés ?**
2. **Quelles sont les lois sur la liberté d'expression ?**
3. **Quels sont les tweets liés à la technologie ?**
4. **Comment l'intelligence artificielle est-elle perçue ?**
5. **Quelle est la loi sur les traitements inhumains ?**
6. **Quels sont les commentaires sur le football ?**

Instructions du mini-projet

- **Source de données** : Donnez aux étudiants un des fichiers de données mentionnés ci-dessus (`tweets.txt`, `lois.txt`, ou `donnees_melangees.txt`).
- **Pipeline d'encodage** : Ils doivent encoder les documents (les tweets ou articles) en vecteurs en utilisant un modèle d'embedding comme `BERT`, `SBERT`, ou `Word2Vec`.
- **Recherche** : Implémentez une méthode pour traiter une requête utilisateur, encoder cette requête, et utiliser un algorithme de similarité (comme la distance cosinus) pour retourner le document le plus pertinent.
- **Stockage des données** : Les vecteurs et les documents originaux doivent être stockés dans un fichier JSON pour une meilleure organisation.