Résolution d'Entités Utilisant des Signatures Probabilistes - Psig

Ce projet effectue la résolution d'entités en utilisant des **signatures probabilistes**. Le processus inclut la préparation des données, le mapping, la normalisation et l'application d'un modèle de résolution d'entités basé sur des signatures probabilistes.

Structure du Projet

Structure du Répertoire

arduino
Copy code

```
project-root/
│
├── main.py
├── constants.py
├── Dockerfile
├── requirements.txt
├── README.md
│
├── prepare.py
├── mapping.py
├── describe.py
├── normalize.py
├── psig_model.py
│
├── input_data/
├── output_data/
└── Dictionaries/
```

Description des Fichiers

- **main.py**: Orchestre l'ensemble du pipeline en exécutant séquentiellement les étapes et en gérant les chemins d'entrée/sortie.
- **constants.py**: Stocke toutes les constantes utilisées dans le projet, y compris les chemins de fichiers et les seuils par défaut.

- **Dockerfile**: Définit la configuration du conteneur Docker pour le déploiement du projet.
- **requirements.txt**: Liste des dépendances Python nécessaires pour exécuter le projet.
- **README.md**: Fichier de documentation fournissant des instructions et des détails sur le projet.
- **prepare.py**: Traite les DataFrames en supprimant les colonnes avec un grand nombre de valeurs NaN et en filtrant les colonnes en fonction des mappings.
- **mapping.py**: Gère les tâches de mapping des données, appliquant des mappings prédéfinis aux colonnes du DataFrame.
- **describe.py**: Fournit des capacités de description des données, y compris des résumés statistiques et la génération de tables synthétiques.
- **normalize.py**: Normalise les adresses en utilisant la géocodification et pré-traite les colonnes de chaînes dans les DataFrames.
- **psig_model.py**: Contient la classe PsigModel pour la résolution d'entités utilisant des signatures probabilistes.
- **input_data/**: Répertoire contenant les fichiers CSV d'entrée à traiter.
- **output_data/**: Répertoire où les fichiers de sortie générés par le pipeline sont sauvegardés.
- **Dictionaries/**: Répertoire contenant les fichiers JSON utilisés pour les mappings de données.

Configuration et Prérequis

Configuration de l'Environnement

- **Python 3.8 ou supérieur** doit être installé.
- Installez les dépendances nécessaires listées dans `requirements.txt`.

Installation des Dépendances

1. Naviguez vers le répertoire du projet.

Exécutez la commande suivante pour installer les dépendances :

```
pip install -r requirements.txt
```

2.

Instructions d'Utilisation

Exécution du Pipeline

1. Assurez-vous que les fichiers CSV d'entrée sont placés dans le répertoire `input_data/`.

2. Ajustez les mappings et les paramètres dans les fichiers de dictionnaire respectifs situés dans `Dictionaries/`.
3. Ouvrez un terminal et naviguez vers le répertoire du projet.

Exécutez la commande suivante pour lancer le pipeline :

```
python main.py
```

4. Suivez les sorties du terminal pour surveiller le progrès et la complétion de chaque étape du pipeline.
5. Les fichiers de sortie seront enregistrés dans le répertoire `output_data/` tel que spécifié dans `constants.py`.