

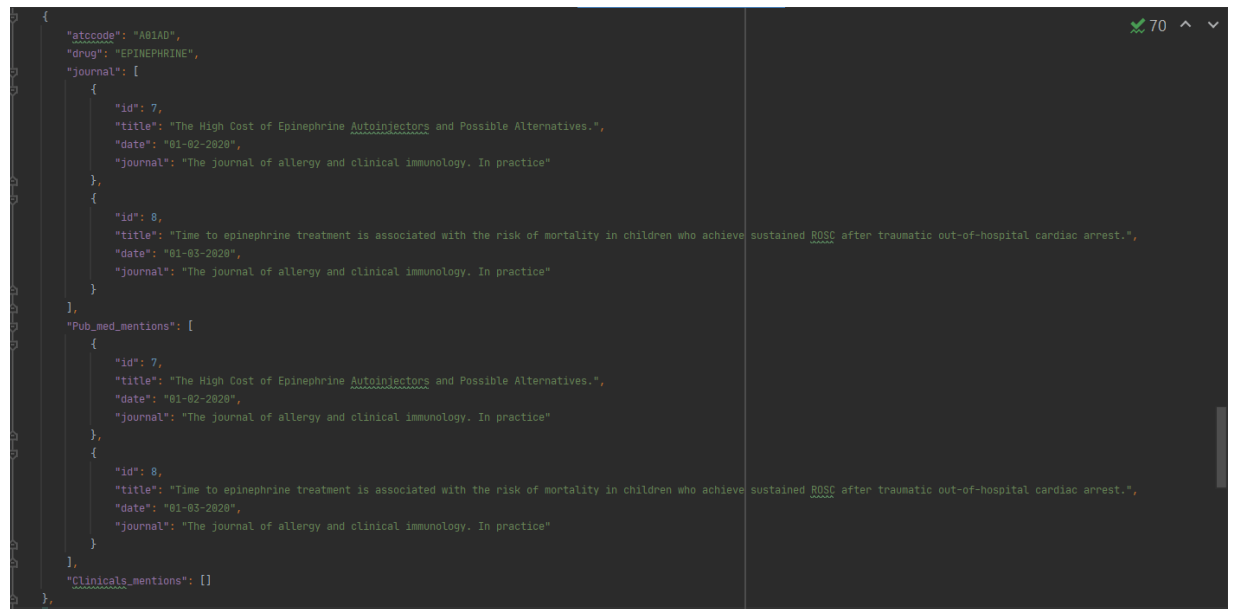
Compte Rendu

Oussama Derouiche

15/06/2020

1 Choix du type de format de json du resultat

Comme vous pouvez remarquer dans le fichier output-pipeline.json le format choisi pour le resultat est la suivante: Nous avons choisi cette format pour les



```
{
  "atccode": "A81AD",
  "drug": "EPINEPHRINE",
  "journal": [
    {
      "id": 7,
      "title": "The High Cost of Epinephrine Autoinjectors and Possible Alternatives.",
      "date": "01-02-2020",
      "journal": "The journal of allergy and clinical immunology. In practice"
    },
    {
      "id": 8,
      "title": "Time to epinephrine treatment is associated with the risk of mortality in children who achieve sustained ROSC after traumatic out-of-hospital cardiac arrest.",
      "date": "01-03-2020",
      "journal": "The journal of allergy and clinical immunology. In practice"
    }
  ],
  "Pub_med_mentions": [
    {
      "id": 7,
      "title": "The High Cost of Epinephrine Autoinjectors and Possible Alternatives.",
      "date": "01-02-2020",
      "journal": "The journal of allergy and clinical immunology. In practice"
    },
    {
      "id": 8,
      "title": "Time to epinephrine treatment is associated with the risk of mortality in children who achieve sustained ROSC after traumatic out-of-hospital cardiac arrest.",
      "date": "01-03-2020",
      "journal": "The journal of allergy and clinical immunology. In practice"
    }
  ],
  "Clinicals_mentions": []
}
```

Figure 1: format json

raisons suivantes :

- Elle répond à notre bien notre besoin de connaître les relations entre les médicaments, la publication médicale, les journaux et les essais cliniques.
- Dans un journal, on peut mentionner un médicament dans une publication médical ou un essai clinique.

- N'oublions pas que le résultat peut être transmis dans un entrepôt de données ou une base de données ce format permettra de garder les liens entre les différentes tables par exemple.
- Dans un journal, on doit spécifier le type de l'article dans lequel le médicament a été mentionné. (l'id de l'article et le type peuvent former une clé primaire pour une mention-journal)

2 Explication de quelques points dans notre pipeline ETL

- les fichiers logs suivent le fonctionnement de notre pipeline.
- les fichiers dans ./input/input-description sont des fichiers json qui décrivent notre source de données. Avant de commencer, il faut toujours analyser les données voir les colonnes importantes pour notre traitement, les liens entre eux, leurs types et les transformations nécessaires pour résoudre notre problème. Avec ce format, on peut ajouter une description dans la liste de la colonne [type, dans la combinaison unique ou non, et d'autres caractéristiques et selon la description de la colonne, on effectue et on automatise notre pipeline, c'est très important surtout pour la phase d'extraction
- Dans la phase de l'extraction, il faut toujours extraire les données utiles, pour cela, il faut valider des données (la fonction set-columns) avant de commencer la phase de transformation. Dans des cas plus compliqués stocker des données inutiles et les traiter ça coûte très cher pour que dans la transformation doit prendre des données propres et effectuer des jobs clairs, et moins cher côté traitement.

3 Réponse pour la partie 6 " pour aller plus loin"

Deux problèmes majeurs qui vont nous rencontrer lorsque on va évoluer votre code afin qu'il puisse gérer de grosses volumétries de données :

- Stockage: Dans ce cas les fichiers CSV, Excel et json vont être difficiles à gérer. Même le résultat va coûter cher lors du stockage.
- Processing : le processing actuel ne va pas répondre convenablement pour des données de plus grande taille

Pour cela, il faut, de mon point de vue migrer notre pipeline vers les services Cloud. Puisque j'ai travaillé avec GCP et leurs services pour la création des data pipelines. Bigquery Dataflow Apache beam m'a permis d'améliorer le processus avec leurs fonctionnalités :Bigquery: un

entrepôt de données qui peut être utilisé en complément de MapReduce et te permet de lui interpréter des données énormes à travers de requêtes SQL. Dataflow: qui te permet l'orchestration de traitement des données par flux et par lot sans serveur, à la fois rapide, unifié et économique. On peut lancer des job de façon parallèle et un service qui gère automatique des clusters de serveurs. Apache Beam: Apache Beam pour définir et exécuter des flux de données, y compris ETL, traitement par lots et en flux à l'aide des SDKs (python, java)