INALCO - Licence LMFA - TNM L3, Algorithmique et structures de données

Caroline Koudoro-Parfait

17 avril 2023

Nom de l'étudiant :
n° d'étudiant :

Projet

1 Modalités :

Le projet :

- est individuel,
- sera constitué :
  - d'un programme python commenté dans ses grandes lignes,
  - d'un répertoire qui comprend les entrées et d'un autre qui contient vos sorties,
  - d'un fichier texte qui répondra aux questions posées dans l'énoncé.
- Vous devez vous appuyer sur les exercices effectués les semaines précédentes.
- il doit être envoyé par email le lundi 17 avril 2023 avant 17h45 au plus tard, à caroline.parfait@sorbonne-universite.fr.

1.1 Constitution du Corpus

Pour effectuer ce projet, vous devez disposer d'un corpus téléchargeable sur le drive : `https://drive.google.com/drive/folders/1WMsh2cRJ0R91yP1oXULIXPO0KvQTk3Hg?usp=sharing`.

Tesseract et Kraken sont des outils de **Reconnaissance Optique de caractères (OCR)**. Ils permettent de récupérer le texte figurant sur des images (photographies, scanner).

- Décrivez, en quelques lignes, l'architecture de dossier de ce corpus.

- Décrivez les données que contiennent les fichiers de ce corpus.
- Précisez quelle architecture de dossier vous pensez appliquer pour utiliser ce corpus dans votre programme de manière optimisée.

1.2 Lecture du Corpus

Écrire une fonction qui permet de lire les fichiers de votre corpus.

1.3 Reconnaissance des entités nommées de la classe PERS

En utilisant Spacy : `https://spacy.io/`.

Vous écrirez un **programme optimisé** qui permet de :

- Proposez un tableau qui donne le nombre de phrases et le nombre de tokens pour le texte de référence (REF) et les texte générés avec kraken et Tesseract.
- récupérer uniquement les entités nommées de la classe Personne (PERS),
- compter le nombre d'entités PERS récupérées et calculer la fréquence d'apparation de chacune des entités PERS récupérées.
- stocker les entités et leur contexte droit et gauche (les 150 mots d'avant et d'après chaque entité) dans un fichier json et csv, selon la structure de données que vous jugez la plus pertinente pour chacun des deux types de sorties.

Expliquez, en quelques lignes, votre choix concernant la structure de données.

2 Comparaisons automatiques : Similarités et Distances avec scikit-learn

Documentation de scikit-learn : `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html`

- Vous utiliserez au minimum 2 métriques,
- Vous élaborerez ce programme pour stocker au format json :
  - le nom des textes comparés,
  - les noms des métriques utilisées,
  - les résultats correspondants, et non pas les tableaux de résultats.
- Vous représenterez les résultats obtenus sous forme de graphiques. La visualisation des résultats doit permettre de pouvoir identifier (vous comparez le texte de référence avec la version OCR) :

  - quelle est la meilleure version des textes entre Kraken et Tesseract selon les résultats des métriques utilisées,
  - quelle est la meilleure version des sorties de Reconnaissance d'entités nommées PERS entre Kraken et Tesseract selon les résultats des métriques utilisées,

- Les graphiques doivent :
  - comporter une légende,
  - avoir pour nom le nom du fichier du texte comparé à la référence, auquel vous ajouterez une mention précisant sans ambiguïté la nature de l'image,
  - être stockés au format image png.

  Pour ce faire, vous pouvez utiliser la bibliothèque :

  - Matplotlib (`https://python.doctor/page-creer-graphiques-scientifiques-python-apprend`