SAÉ105 : Traiter des données

Description générale de la SAÉ

Les métiers R&T font souvent appel à des compétences particulières en informatique pour résoudre les problématiques liées aux systèmes d'information. Ces problématiques sont souvent reliées aux données générées par ces mêmes systèmes. En effet, les systèmes d'information génèrent une quantité gigantesque de données pouvant être traitées avec des objectifs très variés comme le nettoyage des données, l'extraction d'information et/ou l'archivage des données.
D'une manière générale le traitement de données se fait en plusieurs étapes en commençant par la collecte des données, le nettoyage des données, l'extraction d'informations et enfin l'analyse et l'évaluation des résultats. Ces traitements peuvent être récurrents d'où l'intérêt de développer des modèles permettant d'automatiser ces tâches.
Dans cette SAÉ, vous allez être amenés à développer des scripts permettant de réaliser les différentes étapes composant le pipeline complet de l'analyse de données textuelles.

Contexte

La SAÉ se déroulera comme un projet qui a pour objectif d'exploiter les connaissances acquises dans les ressources R107 et R108 pour le développement de scripts permettant l'analyse de données textuelles. Il sera réalisé en plusieurs étapes qui consistent en :

- La collecte de données (lecture, réorganisation et sauvegarde) textuelle.
- Nettoyage des données (extraction des données pertinentes en lien avec le projet).
- Traitement des données pour répondre à la problématique (représentation vectorielle du texte, études statistiques, …)
- Création d'une représentation graphique des différentes études statistiques faites sur les données textuelles
- Interprétation et évaluation des résultats obtenus.

Livrables

Projet individuel de développement informatique avec des livrables finaux :

- Les programmes réalisés (regroupe tous les codes que vous auriez réalisés),
- Un rapport de quelques pages retraçant les différentes étapes du projet et incluant un petit résumé en anglais,
- Une petite présentation orale que vous devriez préparer pour la dernière séance de TP,

Organisation

La SAÉ se déroulera comme suit :

- Un projet individuel de développement informatique.

- Des livrables à fournir pour chaque étape tout au long du projet.
- Un livrable final à rendre en fin du projet, un rapport et une petite présentation.

Dans ce cadre :

- Les séances de travail encadrées (TD/TP) seront consacrées à l'évaluation des avancées, la résolution des problèmes persistants et l'introduction des nouvelles tâches devront être réalisées pour la prochaine séance, et de connaissances supplémentaires si nécessaire.
- Les séances de travail non-encadrés (TP) seront consacrées à la réalisation du travail demandé, et l'implémentation des fonctionnalités qui devront être livrées pour chaque étape du projet.

Définitions

Fouille de données textuelles

Les données textuelles sont présentes sous des formes diverses, allant des textes élaborés au tags (mots-clés) et en incluant les transcriptions des SMS, des tweets et des publications sur les réseaux sociaux. Toutes ces données sont destinées à être lues et comprises par des humains. Cette diversité de sources affecte l'uniformité des textes et creuse le fossé sémantique (la différence de langage utilisé dans un texte élaboré, un SMS et/ou une publication sur les réseaux sociaux) rendant l'interprétation de ces données par un ordinateur plus difficile et complexe. Les méthodes automatiques d'analyse de données ne sont pas encore capables de combler le fossé entre l'interprétation humaine et l'interprétation par un ordinateur. Cependant, il est souvent possible d'automatiser l'extraction, le nettoyage, l'analyse et l'organisation de ces données suivant un objectif précis.

Collecte et pré-traitement des données textuelles

La première étape d'une analyse de données textuelles est la collecte de ces données. En effet, diverses sources de données textuelles existent. Donc, la collecte de données passe d'abord par l'identification des sources (possibilité de regrouper des données issues de plusieurs source). L'identification de la source de données à utiliser peut conditionner les pré-traitements qui devront être effectuées sur ces données (sachant que la compatibilité des données avec l'objectif de l'analyse réduit les pré-traitements). Une fois les sources identifiées, les contenus de ces sources doivent être récupérés et enregistrés comme des données brutes qui devront subir un pré-traitement avant qu'elles soient utilisable. Ces prétraitements peuvent aller de la simple suppression des autres contenus (exemple : scripts, menus… pour les pages web) à l'uniformisation du codage (élimination des caractères spéciaux comme les symboles, les sauts de ligne …). Cette étape de collecte et de pré-traitement nous produit un corpus de textes uniformisés et prêts à être analysés dans le cadre d'un objectif prédéfini.

Analyse des données textuelles

Différents types d'analyse sont possible sur un corpus de textes. Ces analyses peuvent y aller de simples analyses statistiques à des modèles plus complexe permettant l'interprétation et la compréhension du contexte. Ces analyses se déroulent sur plusieurs étapes incluant l'extraction d'entités primaires (mots, mots composés…), l'extraction d'entités nommées (noms propres), la représentation vectorielle des textes …
Dans ce projet, nous allons nous focaliser sur des analyses basiques (études statistiques). Ces études vont inclure le comptage de caractères, de mots et de lignes dans chaque texte du corpus. Ensuite, nous allons faire des statistiques sur l'apparition de mots-clés (recherche de chaînes de caractères, combinaison entre plusieurs chaînes de caractères …).
Dans cette deuxième partie du projet, vous devrez développer différents scripts permettant de faire sur analyses sur un corpus de textes.

Utilisation des modèles développés et évaluation des résultats

Une fois que vous aurez développé les différents scripts permettant de réaliser la collecte, le pré-traitement et les analyses statistiques sur ces données textuelles, vous devrez dans cette dernière partie créer des scripts permettant d'enregistrer les données recolletées et les résultats des analyses effectuées. Vous devrez réaliser une représentation graphique de vos résultats en utilisant le module python **matplotlib** et enregistrer ces représentations sous forme d'images.

Cas pratique

Dans le cadre de cette SAÉ, vous allez réaliser un projet permettant d'analyser des tweets. Pour cela vous devrez développer plusieurs scripts permettant de réaliser les différentes étapes d'analyse de données textuelles définis précédemment. Les données que vous allez utiliser tout au long de cette SAÉ vous seront fournies en fichiers texte qui contiens des tweets (récupérés sur Twitter) sous la forme illustrée ci-après.


```
{
        "data": [
{
        "author_id":"2244994945",
        "created_at":"2020-02-14T19:00:55.000Z",
        "id":"1228393702244134912",
        "text": "What did the developer write in their card?\n"
        }
],}
```


Exemple de données que vous allez utiliser pour la SAÉ.

Problème

On suppose un répertoire contenant plusieurs fichiers contenant des données textuelles en format texte (fichier1.txt, fichier2.txt, … fichierN.txt). On veut analyser les données textuelles de ces fichiers pour compter le nombre d'apparition de chaque mot dans l'ensemble de ces fichiers et enregistrer le résultat de cette analyse dans un fichier (csv) qui doit contenir la fréquence d'apparition des 15 mots apparaissant le plus fréquemment. Pour cela, vous devez :

1. Ecrire un script (python ou shell) permettant d'initialiser votre environnement de travail en créant l'arborescence suivante :

```
src
`-- Prog
    `-- SAE105
        |-- Codes
        |-- Resultats
        `-- Donnees
             |-- Donnee_Brute
             `-- Donnee_pret
```

On considère comme mot toute chaîne de caractère composée de caractères appartenant à l'ensemble contenant les lettres (a – Z), les chiffres (0 – 9) et le caractère de soulignement (_). Cette chaîne doit être précédée immédiatement et suivie immédiatement par des caractères n'appartenant pas à cet ensemble.

2. Ecrire une fonction python *compte_lignes_mots(nom_fichier)* qui compte le nombre de lignes et de mots du fichier *nom_fichier*. La fonction prendra en entrée le fichier *nom_fichier* qu'on suppose existant et renverra une liste de deux éléments dont le premier est le nombre de lignes, le second le nombre de mots du fichier.

3. Ecrire une fonction python *compte_dans_fichiers(liste_fichiers)* qui compte les lignes et les mots dans chacun des fichiers de la liste *liste_fichiers*. La fonction prendra en entrée une liste de fichier *liste_fichiers* (on suppose que tous les fichiers de la liste existent) et renverra une liste dont chaque élément est une liste de deux éléments : Le premier élément est le nombre de lignes, le second le nombre de mots du fichier correspondant.
4. Ecrire une fonction python *mots_fichier(nom_fichier)* qui recense tous les mots utilisés dans le fichier *nom_fichier* et compte le nombre d'utilisations de chacun de ces mots. La fonction prendra en entrée le fichier *nom_fichier* (on suppose que le fichier existe) et renverra une liste dont chaque élément est une liste de deux éléments : le premier est le mot utilisé, le second le nombre d'utilisation de ce mot dans le fichier.
5. Ecrire une fonction python *mots_dans_fichiers(liste_fichiers)* qui recense tous les mots utilisés dans l'ensemble des fichiers de la liste *liste_fichiers* et compte le nombre d'utilisations de chacun de ces mots dans chacun des fichiers. La fonction prendra en entrée la liste de fichier *liste_fichiers* (on suppose que tous les fichiers de la liste existent) et renverra une liste dont chaque élément est une liste de plusieurs éléments : le premier est le mot utilisé, le deuxième élément est le nombre d'utilisation de ce mot dans le premier fichier, le troisième est le nombre d'utilisations de ce mot dans le deuxième fichier…et le dernier élément et le nombre d'utilisation de ce mot dans le dernier fichier.

6. Ecrire une fonction python *apparition_mots(liste_fichiers)* qui calcule la fréquence d'apparition des 15 mots apparaissant le plus fréquemment dans chacun des fichiers de la liste *liste_fichiers.* La fonction prendra en entrée la liste de fichiers liste *liste_fichiers*

(on suppose que tous les fichiers de cette liste existent) et renverra une liste dont chaque élément est une liste de 15 éléments : les éléments de cette liste sont les fréquences d'apparition des 15 mots apparaissant le plus fréquemment dans le fichier correspondant. La liste renvoyée par cette dernière fonction doit être enregistrée dans un fichier (csv : format de texte simple qui convient au stockage de données structurées simples). Dans notre cas, le fichier contiendra N lignes (chaque ligne correspondra à un fichier de la liste *liste_fichiers*). Chacune de ces lignes contiendra 15 champs séparés par des virgules (',') et chaque champ contiendra la fréquence d'apparition de l'un des 15 mots apparaissant le plus fréquemment dans fichier correspondant.