Consignes du projet - étape 2

Dans ce document, je vous présente les consignes de la deuxième étape du projet.

- Après avoir aspiré les entités médicales de type noms de médicaments par substance active à partir du VIDAL et généré le dictionnaire « subst.dic », vous devrez écrire un **deuxième script Python** permettant d'**enrichir** ce dictionnaire « **subst.dic** » avec de nouvelles entités médicales de type noms de médicaments par nom commercial ou par substance active, à partir du fichier « **corpus-medical.txt** » donné en argument. **L'encodage UTF-8 sans BOM du fichier du corpus médical ne doit pas être modifié** et le dictionnaire « subst.dic » après enrichissement **doit conserver son encodage de départ**, à savoir l'« **UTF-16 LE avec BOM** » (UCS-2 LE BOM).
- Le dictionnaire « subst.dic » après enrichissement ne doit pas contenir de doublons et doit être trié par ordre croissant (a-z). Il contiendra donc toutes les entités médicales de type noms de médicaments par substance active issues du VIDAL selon l'intervalle choisi **+** les nouveaux noms de médicaments issus du corpus médical.
- Le script d'enrichissement **doit garder une trace** des noms de médicaments trouvés dans le fichier « corpus-medical.txt », en les stockant dans un autre fichier qui **doit** s'appeler « **subst_enri.dic** », en mettant ses entrées lexicales en minuscules. Cependant, **ce dictionnaire ne doit subir ni tri, ni suppression de doublons** et doit être encodé en « **UTF-16 LE avec BOM** » (UCS-2 LE BOM).
- Le contenu du fichier « **subst_enri.dic** » doit être affiché sur la console (invite/ligne de commandes) avec un compteur démarrant à 1.
- Générer un fichier nommé « infos2.txt » **sans doublons** contenant :
    - le nombre de médicaments **issus du corpus** pour chaque lettre de l'alphabet ;
    - et le nombre total de médicaments **issus du corpus**.
- Générer un fichier nommé « infos3.txt » **sans doublons** contenant :
    - le nombre de médicaments **issus de l'enrichissement** pour chaque lettre de l'alphabet ;
    - et le nombre total de médicaments **issus de l'enrichissement**.
- Construire un graphe d'extraction sous UNITEX, qui se base sur l'étiquette **<N+subst>** du dictionnaire « subst.dic », afin d'**extraire les occurrences de** « **posologies** » à partir du fichier

« corpus-medical.txt ». Le graphe d'extraction **doit s'appeler** « **posologie.grf** ». Le résultat de cette extraction sera placé par UNITEX dans le fichier « **concord.html** », qui se trouve dans le dossier « corpus-medical_snt ».

Remarque : Une « posologie » contient le nom du médicament, le dosage du médicament (50mg, 20 mg, 10 mg, etc.), la dose (usuelle ou maximale), le rythme d'administration (ou fréquence d'administration), l'heure-moment de prise du médicament (à 8 heures, le soir, le matin, etc.) et la durée de traitement (pendant un mois, de J1 à J7, etc.).

Par exemple :

SIMVASTATINE 20 mg : 1 cp/j à 8 heures pendant un mois

CYTARABINE 100 mg/m² de J1 à J7

ZOLPIDEM 10 mg 1 cp au coucher

METFORMINE 850 mg 3 fois par jour

SPECIAFOLDINE 5 mg : 1 cp matin

ALADACTONE 25 mg : 1 cp/jour le midi

INEXIUM 40 1 cp par jour le soir

Remarque : Une « posologie » n'est pas forcément précédée du token « posologie ». Il faut donc bien analyser le fichier « **corpus-medical.txt** » que je vous ai envoyé en pièce-jointe, afin de découvrir les différentes façons d'exprimer une « posologie ».

Remarque : Il est à noter que dans certains cas, le dosage de médicament n'est pas présent, par exemple, "METOPROLOL : ½ le matin, ½ le soir". Dans cet exemple, le dosage du METOPROLOL n'est pas précisé. Pourtant, il en existe différents dosages, comme le "METOPROLOL 100 mg", employé dans "METOPROLOL 100 mg : ½ le matin, ½ le soir" ou le "METOPROLOL 50 mg", employé dans "METOPROLOL 50 : 1/jour".

Remarque : Concernant la dose (ou dose journalière), elle peut être soit usuelle, soit maximale. La première peut être déduite à travers le dosage du médicament et le rythme d'administration. Par exemple, dans "METFORMINE 850 mg 3 fois par jour", on multiplie 850 mg par 3 fois, ce qui nous donne une dose usuelle de 2550 mg/jour. Pour la dose maximale d'un médicament, on peut donner l'exemple du Paracétamol dont la dose maximale journalière est de 4 g/jour.

Remarque : Il est à noter également que les termes "dose" et "dosage" sont également employés pour désigner la quantité d'un médicament par kilo de poids. Le dosage de l'Adénosine, par

exemple, est fixé à 0,1 mg par kilo de poids. Donc, pour un bébé de 6 mois qui pèse 7 kg, la dose de ce médicament est de 0,7 mg.

- Écrire un troisième script permettant d'appeler UNITEX pour exploiter votre graphe, à partir de l'emplacement **C:\.....\Unitex-GramLab\App>**
    a. **Pour appeler UNITEX, vous devrez utiliser le script du cours 8, (Lancer UNITEX à partir d'un script Python).** Ce troisième script Python « unitex.py » **doit exploiter** les **ressources** suivantes :
        I. le dossier « **corpus-medical_snt** » créé automatiquement à chaque lancement du script « unitex.py » ;
        II. le fichier : « **corpus-medical.txt** » ;
        III. le fichier : « **corpus-medical.snt** » ;
        IV. le fichier : « **Norm.txt** » (facultatif) ;
        V. le fichier : « **Alphabet.txt** » ;
        VI. le fichier : « **subst.dic** » ;
        VII. le fichier : « **subst.bin** » ;
        VIII. le fichier : « **Dela_fr.bin** » ;
        IX. le fichier : « **Dela_fr.inf** » ;
        X. le fichier : « **posologie.grf** » ;
        XI. le fichier : « **posologie.fst2** » ;
        XII. le fichier : « **concord.ind** ».

Remarque : Lors de la phase d'extraction, il est **nécessaire** d'utiliser comme ressource supplémentaire le dictionnaire système « **Dela_fr.bin** » fourni par UNITEX, afin de pouvoir exploiter les masques lexicaux comme **<PREP>, <DET> ou <PREPDET>**, etc. **Vérifiez aussi que vous avez bien « Dela_fr.inf » à côté du « Dela_fr.bin », afin que ce dernier puisse être exploité.**

- Écrire un quatrième script permettant d'injecter le contenu du fichier « concord.html » dans une base de données **SQLite** nommée « **extraction.db** », en utilisant le module « sqlite3 » de Python. Pour parcourir les données de votre base de données, utilisez « sqlitebrowser ».
- La table « EXTRACTION » de votre base de données contiendra : l'ID (clé primaire) et la POSOLOGIE.

- Pour lancer votre application d'extraction d'information, placez vos 4 scripts (aspirer.py, enrichir.py, unitex.py et sqlite.py) dans l'emplacement **C:\.....\Unitex-GramLab\App>**

Pour l'**évaluation de votre travail**, vous **devrez m'envoyer par mail** :

- **Le script d'aspiration** : « aspirer.py » doit générer « subst.dic » et « infos.txt ». Ce script prend deux arguments :
  I. l'intervalle des pages à traiter, en respectant le format : **B-H, E-S** ou **A-W**, etc. ;
  II. le port http utilisé dans le fichier de configuration du serveur « Apache ».

- **Le script d'enrichissement** : « enrichir.py » doit enrichir le DELAF « subst.dic » à partir du fichier « corpus-medical.txt » donné en argument. Ce script doit générer 4 fichiers :
  I. « subst.dic » (dictionnaire enrichi) ;
  II. « subt_enri.dic » (trace d'enrichissement) ;
  III. « infos2.txt » ;
  IV. « infos3.txt ».
- **Le script SQLite** : « sqlite.py » doit enregistrer les posologies contenues dans le fichier « concord.html » dans la base de données SQLite nommée « extraction.db ». Ce script prend en argument le fichier « concord.html » et génère la BDD « extraction.db ».

- **Le script Python qui appelle UNITEX** : « unitex.py » doit exploiter plusieurs ressources, comme le graphe « posologie.grf **»** et le DELAF « subst.dic ».

- **Le graphe d'extraction** : « posologie.grf » doit extraire à partir du fichier « corpus-medical.txt » les posologies, en s'appuyant sur les DELAF « Dela_fr » et « subst ».

Pour résumer, vous devrez m'envoyer **5 fichiers** :

- les 4 scripts **Python** ;
- et le **graphe d'extraction** au format **.grf**.

La date limite d'envoi de votre projet par mail est fixée au jeudi 18 mars à 23h59.

Cdt,
N.Z