Étape 1

- Écrire un premier script Python permettant d'extraire (web scraping) les entités médicales de type **noms de médicaments par substance active** de A à Z, à partir du contenu des 26 pages HTML du dossier « VIDAL » que je vous ai mis en pièce-jointe.

- Générer en sortie un dictionnaire au format **.dic** (format DELAF vu en cours 4) encodé en **UTF-16 LE avec BOM** (UCS-2 LE BOM).

- Ce dictionnaire **doit s'appeler** « **subst.dic** » et doit donc contenir les noms de médicaments par substance active des 26 pages HTML du dossier « VIDAL ».

- Chaque entrée lexicale de ce dictionnaire doit être suivie par les informations (codes) **,.N+subst**

- L'information **N** est de type grammatical et l'information **subst** est de type sémantique.

- Vous devez donc obtenir une sortie ayant le format **DELAF** d'UNITEX suivant :

  abacavir,.N+subst
  abatacept,.N+subst
  abciximab,.N+subst
  abiratérone,.N+subst
  …….

- L'extraction **doit être faite en local** sur votre machine. Pour ce faire, vous devrez installer une plate-forme de développement Web, comme par exemple : **XAMPP**, qui contient, entre autres, un serveur Web Apache.

Remarque : L'encodage UTF-8 sans BOM des pages HTML du dossier « VIDAL » ne doit pas être modifié.

- Ensuite, donner la possibilité à l'utilisateur de déterminer l'intervalle des pages à traiter, en respectant le format : B-H, E-S, A-W, ou A-Z etc. Cet intervalle **est le premier argument du premier script Python** « **extraire.py** ».

- Donner également la possibilité à l'utilisateur de saisir le « port http », qui est précisé dans le « fichier de configuration (httpd.conf) du serveur Web Apache ». Ce port est le deuxième argument du **premier script Python** « **extraire.py** ».
- Autrement dit, une fois que vous avez choisi votre port **manuellement** dans ce fichier de configuration, vous le mettez ensuite comme deuxième argument à votre script « **extraire.py** ». Ce script doit exploiter ce port pour accéder à l'URL des 26 pages HTML du dossier VIDAL, qui seront accessibles en local.
- Générer un fichier nommé « **infos1.txt** » contenant :
    - le nombre d'entités médicales de type noms de médicaments par substance active du dictionnaire « subst.dic » généré préalablement, pour chaque lettre de l'alphabet ;
    - et le nombre total d'entités médicales de type noms de médicaments par substance active de ce dictionnaire.

Remarque : Le port http par défaut est le port « **80** ».

Remarque : Ce premier script python « **extraire.py** » **doit impérativement avoir 2 arguments : l'intervalle d'extraction et le port http.**

Remarque : Votre script python **ne doit pas modifier (écrire dans) le fichier de configuration (httpd.conf) Apache durant l'étape d'extraction.**

Étape 2

- Après avoir extrait les entités médicales de type noms de médicaments par substance active à partir du dossier « VIDAL » et généré le dictionnaire « subst.dic », vous devrez écrire un **deuxième script Python** « **enrichir.py** », permettant d'alimenter et d'**enrichir** le dictionnaire « **subst.dic** » (généré dans l'étape précédente) 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 dossier « VIDAL » selon l'intervalle choisi **+** les nouveaux noms de médicaments issus du corpus médical « **corpus-medical.txt** ».

- 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_corpus.dic** », en mettant ses entrées lexicales en minuscules. Cependant, **ce dictionnaire doit subir ni tri, ni suppression de doublons** et doit être encodé en **« UTF-16 LE avec BOM »** (UCS-2 LE BOM).
- 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 conservés pour **l'enrichissement** pour chaque lettre de l'alphabet ;
    - et le nombre total de médicaments conservés pour **l'enrichissement**.

Étape 3

- Construire un graphe d'extraction (.grf) sous UNITEX, qui se base **impérativement** 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 » généré par UNITEX.

  Remarque : Une « posologie » contient généralement le nom du médicament, le dosage du médicament (**50 mg**, **20 mg**, **10 mg**, **500**, **400**, **0,4 ml**, **0.4 ml**, **0,4**, **4000 UI**, etc.), le rythme ou fréquence d'administration (2 fois par jour, 3 fois par jour, 4 fois par jour, 1 le matin et 1 le soir (donc 2 fois par jour), etc.), l'heure-moment de prise du médicament (à 8 heures, à 20h00, le soir, le matin, trois fois par jour (donc le matin, le midi et le soir) etc.) et la durée de traitement (pendant un mois, pendant encore 21 jours, de J1 à J7, etc.).

  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".

Exemples d'extraction de « posologies » à partir du corpus médical « corpus-medical.txt » :

TOPALGIC 100 mg 1 amp, 3 fois par jour, pendant 5 jours

INNOHEP 3 500 unités : 1 injection par jour pendant encore 21 jours

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

CYTARABINE 100 mg/m² de J1 à J7

PLAVIX 75 mg : 1 cp/jour

ZOLPIDEM 10 mg 1 cp au coucher

METFORMINE 850 mg 3 fois par jour

SPECIAFOLDINE 5 mg : 1 cp matin – 1 cp soir pendant un mois

ALADACTONE 25 mg : 1 cp/jour le midi

INEXIUM 40 1 cp par jour le soir

TEGRETOL 200 mg : 1 cp 2 fois par jour

PAROXETINE 20 mg : 1 fois par jour

EQUANIL 400 : 3 fois / jour

KEPPRA 500 : 2/jour

CRESTOR 10 mg : 1 comprimé par jour le soir

LOVENOX 0,4 : 20h

LOVENOX 4000 UI : 1/jour

LOVENOX 0,4 : 19h

LOVENOX 0,4 ml : 1 injection/jour le soir

LOVENOX 0.4 1 inj/jour à midi

Remarque : Dans certains cas, l'heure-moment de prise du médicament et la durée de traitement ne sont pas précisés, comme dans la posologie suivante :

PLAVIX 75 mg : 1 cp/jour

- 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 dédié au lancement d'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** » ;
        V. le fichier : « **Alphabet.txt** » (préciser dans quelle phase du script « **unitex.py** » ce fichier « Alphabet.txt » doit être utilisé et expliquer à quoi sert ce fichier TXT, en donnant des exemples précis. Cette réponse doit être écrite sous forme de commentaires dans le script « **unitex.py** ».) ;
        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, utiliser « DB Browser for SQLite ».
- 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 (**extraire.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'extraction** : « **extraire.py** » doit générer « subst.dic » et « infos1.txt ». Ce script prend deux arguments :
  I. l'intervalle des pages à traiter, en respectant le format : **B-H, E-S, A-W**, ou **A-Z**, 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 à partir du fichier « corpus-medical.txt ») ;
  II. « subst_corpus.dic » ;
  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.bin » et « subst.bin ». Le résultat doit contenir au minimum **1000 extractions correctes**.

Pour résumer, vous devrez m'envoyer **7 fichiers** :

- les 4 scripts **Python** ;
- Le fichier **« concord.html »** ;
- La base de données « **extraction.db** » ;
- et le **graphe d'extraction** au format **.grf**.