Deuxième partie : Tokenisation et découpage en phrase

Le but de cette séance est de manipuler un tokeniseur permettant de découper un corpus en *token* par rapport à un lexique de référence puis d'écrire un programme permettant de détecter les fins de phrases grâce à des heuristiques simples, et d'effectuer des normalisations sur le texte ainsi découpé.

Mise en pratique

- Partie 1 : Tokenisation
  - Nous allons utiliser comme lexique de référence pour la tokenisation, un lexique d'environ 90K mots extrait de la ressource *LeFFF*. Ce lexique est au format suivant:
    `indice \tab mot \tab nb-occ \tab pos1 \tab lemme1 \tab pos2 \tab lemme2 \tab ....`
  - avec :
    - *indice* : indice numérique entre 000000 et 999999 identifiant le mot de manière unique
    - *mot* : la forme devant être reconnue comme un *token* atomique par le tokeniseur
    - *nb-occ* : le nombre d'occurrence de la forme dans un grand corpus de texte représentatif de la langue française
    - *pos1 lemme1* : couple d'informations regroupant une catégorie morphosyntaxique possible pour la forme et le lemme associé
  - A chaque forme peut correspondre plusieurs catégories morphosyntaxiques, par exemple le mot *cours* peut être un nom avec comme lemme *cour* (*la cour de l'école*), ou bien *cours* (*un cours d'informatique*), ou encore un verbe avec comme lemme le verbe à l'infinitif (*courir*).
    Par exemple, pour les mots *une* et *cours*, nous aurons :
    `440231 une 119173 det un nc une pro un`
  - `094571 cours 5233 nc cour nc cours v courir`
  -
  - Vous pourrez télécharger le fichier sur le lien suivant : https://pageperso.lis-lab.fr/frederic.bechet/M1_TAL_TP_data/lex_lefff_pos_lemme_freq_90k.txt
  - Vous utiliserez dans un premier temps la commande *tokenize* qui vous est fournie sous la forme d'un programme en langage C que vous compilerez et utiliserez dans votre console Unix. Cette commande prend en entrée un corpus au format *plain text*, un lexique tel que *lex_lefff_pos_lemme_freq_90k.txt* et différentes options dont voici la liste :
    - *-code* : affiche les codes plutôt que les mots en sortie du tokeniseur

        - *-oov* : n'affiche que les mots qui ne font pas partie du lexique (*OOV=Out-Of-Vocabulary*)
        - *-no_oov* : ignore en sortie les mots qui ne font pas partie du lexique
        - *-line* : affiche un mot par ligne en sortie, ainsi que les sauts de ligne
    - Question 1 : en utilisant le tokeniseur et le lexique *lex_lefff_pos_lemme_freq_90k.txt*, calculez le nombre de tokens (incluant les mots hors vocabulaires) contenus dans la concaténation des corpus du répertoire corpus_topic
    - Question 2 : calculez maintenant le taux de mots hors-vocabulaire (tokens n'appartenant pas au lexique) sur toutes les occurrences de tokens de ces corpus
    - Question 3 : sortez la liste triée par fréquence décroissante de ces formes hors-vocabulaire. Décrivez en quelques phrases quelles sont les principales catégories de mots hors-vocabulaire.
    - Question 4 : en utilisant les informations morphosyntaxiques disponibles dans le lexique, sortez les listes classées par fréquence décroissante des formes verbales (étiquette *v*), nominales (étiquette *nc*) et noms propres (étiquettes *np*) de la concaténation des corpus du répertoire corpus_topic
- Partie 2 : Découpage en phrase
    - Vous allez maintenant écrire un programme permettant de découper en *phrases* le texte produit en sortie du tokeniseur. Pour cela vous définirez certaines heuristiques simples permettant de détecter une fin de phrase. Par exemple, l'heuristique la plus simple est de considérer que si après tokénisation on trouve un token "." (point) suivi par un saut de ligne ou un mot commençant par une majuscule, alors c'est une fin de phrase.
    - Question 1 : En utilisant le langage de programmation de votre choix, ainsi que le format de sortie du tokeniseur de votre choix, écrivez un programme permettant de définir de telles heuristiques et de rajouter des balises *<s>* (début de phrase) et *</s>* (fin de phrase) pour structurer les sorties du tokeniseur. En examinant les sorties de votre programme, rajoutez des heuristiques pour le rendre plus performant. Donnez la liste des heuristiques utilisées.
    - Question 2 : Combien de phrases obtenez vous sur l'ensemble des corpus du répertoire corpus_topic ? En examinant les sorties de votre programmes, quels sont à votre avis les problèmes restant pour le découpage des corpus en *phrases* ?
- Partie 3 : Normalisation des corpus
    - Vous avez pu voir en examinant les sorties du tokeniseur qu'il peut y avoir de nombreux problèmes de normalisation dans un corpus : le même mot peut être écris tout en minuscule si ce n'est pas un nom propre, ou bien avec uniquement la première lettre en majuscule s'il se trouve en début de phrase, ou bien entièrement en majuscule, par exemple dans un titre. Afin de généraliser les analyses, il est intéressant de *normaliser* les corpus en

revenant pour chaque mot à sa forme *canonique*, c'est à dire celle que l'on trouve dans un lexique de référence.
    - Enlever directement les majuscules en passant toutes les formes en minuscule n'est pas possible à cause de noms propres et des sigles. On peut là aussi utiliser des heuristiques simples pour faire cette tâche. Par exemple, une fois le découpage en phrase effectué, si un mot commence une phrase, que sa première lettre est une majuscule, et que la forme entièrement en minuscule existe dans le lexique, alors on peut remplacer l'occurrence dans le lexique par la forme en minuscule. De même, dans une phrase où tous les mots sont en majuscule, on peut remplacer les mots par leurs formes en minuscule si elles existentent dans le lexique.
    - Question 1 : Ecrivez un programme qui prend en entrée la sortie de votre programme de découpage en phrase et supprime les majuscules de début de phrase en utilisant l'heuristique précédente.
    - Question 2 : Recalculez le taux de mots hors vocabulaire après nettoyage des majuscules de début de phrase. Qu'en concluez vous ?
    - Question 3 : Quelles sont à votre avis les principaux problèmes restant, outre les majuscules de début de phrase, pour la normalisation des corpus ?
- BONUS : Intégration de tous les traitements
    - Récrivez le tokeniseur ainsi que le programme de découpage en phrase ainsi que le nettoyage au sein d'un programme unique dans le langage de votre choix.
    -