# Travail pratique 2

## Synonymes… BD

Pour le TP1, vous aviez créé un système qui faisait la comparaison des contextes de mots. Dans le TP2, vous devrez emmagasiner dans une BD les données qui vous ont permis d’effectuer ces comparaisons.

On veut pouvoir entraîner notre système sur un texte, avec une taille de fenêtre fixe. Stocker les résultats et recommencer. Chaque fois que vous fournissez un texte, la taille du vocabulaire change, donc votre matrice change de dimension. Vous devez donc vraisemblablement mettre à jour votre vocabulaire avant de calculer les cooccurrences.

Un point TRÈS important, la matrice que vous utilisiez dans le TP1 était assez creuse, c’est-à-dire que la vaste majorité des valeurs dans votre matrice sont 0 (99.9% en fait). Ne conservez donc pas ces valeurs!

#### Tables

Le nombre de tables et leur structure relève de votre choix, mais n’oubliez pas les contraintes là où elles sont nécessaires.

#### Type d’application

Vous devez créer une application de type ligne de commande. Remarquez qu’on lance une commande séparée pour chaque corpus individuel. Notez aussi que l’entraînement et la recherche de « synonymes » se fait en deux exécutions **distinctes**.

#### Arguments

Vous devez gérer des options. Elles peuvent être fournies dans n’importe quel ordre.

Options pour l’entraînement :

* -e : entraînement
* -t <*taille*> : taille de fenêtre. <*taille*> doit suivre -t, précédé d’un espace.
* --enc <*encodage*> : encodage de fichier. <*encodage*> doit suivre --enc, précédé d’un espace.
* --chemin <*chemin*> : chemin du corpus d’entraînement. <*chemin*> doit suivre --chemin, précédé d’un espace.

*Note : les options qui contiennent plus d’une lettre doivent être précédées de 2 tirets afin de respecter la syntaxe GNU/POSIX.*

Lorsqu’on fournit l’option -e, on doit fournir les options -t, --enc et --chemin, avec leurs arguments respectifs. Le système analysera le corpus et ajoutera les nouveaux mots de vocabulaire et les nouvelles cooccurrences pour cette taille de fenêtre dans la BD, suite à quoi il arrêtera son exécution.

*Notez bien que les cooccurrences pour la fenêtre de taille 5, par exemple, et celles pour la taille de fenêtre 7, par exemple, ne sont pas exactement les mêmes…*

Exemple d’appel pour l’entraînement :

Y:\Cooccurrences\src>mainBD.py -e -t 5 --enc utf-8 --chemin ..\textes\GerminalUTF8.txt

Options pour la recherche :

* -r : rechercher des synonymes
* -t <*taille*> : taille de fenêtre. <*taille*> doit suivre -t, précédé d’un espace.

Lorsqu’on fournit l’option -r, on fournit l’option -t, avec la taille, évidemment. Le système chargera le vocabulaire et les cooccurrences pour cette taille de fenêtre. Il demandera ensuite à l’utilisateur un mot, le nombre de résultats et la méthode de calcul, comme au TP1.

Exemple d’appel pour la recherche :

Y: \Cooccurrences\src>mainBD.py -r -t 5

Vous DEVEZ vous conformer aux consignes pour la ligne de commande. Je vous conseille de vous occuper de ces consignes avant tout le reste.

#### Connecteur BD

Vous devez utiliser cx\_Oracle.

#### Équipes

Vous devez vous mettre en équipes de 3 et utiliser Git pour communiquer le code entre vous.

#### À remettre

Les fichiers source Python, les scripts Oracle (création de table et autre).

Pour cette remise, vous n’utiliserez pas Léa, vous devez utiliser Git sur Bitbucket. Vous devez rendre votre repo accessible à l’utilisateur ppmontyCVM. Votre repo doit porter le nom suivant :

Cooccs\_NomPrenom1\_NomPrenom2\_NomPrenom3

Ce repo sera celui que vous allez utiliser jusqu’à la fin de la session. Je vous conseille donc de faire le ménage des fichiers inutiles sur votre repo et d’inclure, dans le répertoire racine, un README où doivent figurer les noms de tous les membres de votre équipe et toute autre information nécessaire à la mise en place de votre système (i.e. exécution des scripts Oracle, configuration du connecteur, recette pour gâteau au chocolat, etc.)