# Travail pratique 2

## Synonymes… BD

Dans le TP1, vous avez créé un système qui fait 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 dimensions. Vous devez prévoir mettre à jour votre vocabulaire avant de recalculer les cooccurrences.

Un point est 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 les valeurs nulles (signifiant l’absence de cooccurrences entre deux mots), elles prendront trop d’espace mémoire.

#### 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 (texte) individuel, mais que la base de données peut maintenant contenir les données cumulatives de plusieurs textes. Notez aussi que contrairement au TP1, l’entraînement et la recherche de « synonymes » se fait en deux exécutions **distinctes**. Vous devez aussi ajouter la possibilité de régénérer la base de données, également dans une exécution distincte.

#### Arguments

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

Options pour l’entraînement :

* -e : entraîner
* -t <*taille*> : taille de fenêtre. <*taille*> doit suivre -t, précédé d’un espace.
* --encodage <*encodage*> : encodage de fichier. <*encodage*> doit suivre --encodage, 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 leur argument respectif. Le système analysera le corpus et ajoutera les nouveaux mots de vocabulaire et les nouvelles cooccurrences pour cette taille de fenêtre dans la BD. Ensuite, il arrêtera son exécution.

*Notez bien que les cooccurrences pour la fenêtre de taille 5 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 --encodage 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 de fenêtre, évidemment. Le système chargera le vocabulaire et aussi les cooccurrences pour cette taille de fenêtre à partir de la BD. Il demandera ensuite à l’utilisateur un mot, le nombre de résultats à afficher et la méthode de calcul, comme au TP1.

Exemple d’appel pour la recherche :

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

Option pour la régénération de la BD :

* -b : régénérer la BD

L’option -b apparaît seule. Son effet est que toutes les tables de la base de données sont détruites et ensuite recréées, vides.

Exemple d'appel pour la régénération:

Y: \Cooccurrences\src>mainBD.py -b

Vous DEVEZ vous conformer aux consignes pour la ligne de commande. Je vous conseille de vous en occuper avant tout le reste. Des modules comme *argparse* existent pour le traitement d’arguments de ligne de commande.

#### Connecteur BD

Vous devez utiliser sqlite3.

#### Programmation

Voici une belle opportunité pour poser un défi à votre compréhension de l’héritage et du polymorphisme!

#### Équipes

Conservez les mêmes équipes qu'au TP1.

#### À remettre

Les fichiers source Python.

S.V.P. gardez le même dépôt que pour le TP1. Créez-y simplement un nouveau répertoire pour le TP2.

N’oubliez pas de faire le ménage des fichiers inutiles sur votre dépôt et d’inclure, dans le répertoire du TP2, un README où doit figurer toute information nécessaire à la mise en place de votre système. Comme pour le TP1, vous devez créer un tag nommé *TP2* lorsque vous avez terminé le travail. Tout DOIT se retrouver sur la branche « main ».