


README :

Installation :

- 1) Installer eXistdb (nécessite JRE ou JDK = 7 ou 8, 512MB de RAM et min 200Mo sur le HDD)
- 2) Mettre les fichiers .xml de l'IDG dans eXistdb dans une base nommée "geobs" de préférence.
- 3) Installer Tulip >= 4.7

Utilisation :

Initialisation :

- 1) Démarrer Tulip.
- 2) Créer un nouveau graphe vide (Ctrl+N).
- 3) Double-cliquer sur le graphe vide qui est apparu dans la fenêtre des graphes en bas à gauche.
- 4) Choisir "Python Script view" dans le "Panel creation dialog".
- 5) Cliquer sur l'icône  puis charger le script "scriptTulip_init.py" dans le répertoire \Scripts Tulip du dépôt.
- 6) Indiquer le nom de la bdd utilisée dans eXistdb dans la variable nameBase du script (par défaut nameBase = "geobs").
- 7) Lancer exécution de scriptTulip_init.py en utilisant le bouton lecture en bas à droite (sous le "script output") ou via le raccourci Ctrl+Enter.

Les fiches de l'IDG s'affichent sous forme de points rouges. Vous pouvez les manipuler vis les options de Tulip ou via les autres scripts disponibles.

Pour utiliser les autres scripts, il suffit de charger le script de la même manière que pour init.py, de régler ses hypothétiques paramètres indiqués dans l'entête puis de lancer la lecture du script.

Autres scripts :

- scriptTulip_actionMode.py : permet de créer les graphes Mot-Clé, Acteur et Hybrid (Acteur/Mot-Clé) en fonction de la valeur de la variable actionMode à modifier dans le script ("keyword", "actor" ou "hybrid").
- scriptTulip_actionMode_hybrid_avanced.py : permet la création du graphe Hybrid (Acteur/Acteur). **Nécessite d'avoir exécuté le script actionMode avec le mode hybrid (actionMode = "hybrid")!**
- scriptTulip_similaritee.py : permet de créer le graphe Similarité. On peut faire varier le seuil en changeant la valeur de THRESHOLD entre 0 et 1. "CleanBeforeAction" permet de supprimer les sommets qui ne sont pas des fiches .xml du graphe avant d'opérer la création du graphe.
- scriptTulip_influence.py : permet de créer le graphe d'emprise. La valeur de THRESHOLD correspond à la surface de recouvrement minimum pour établir un lien entre deux fiches.

- scriptTulip_filter.py : filtre à appliquer **avant les autres scripts** et qui permet de filtrer les fiches traitées en fonction d'une surface ou d'une liste de mots-clés. On choisit le tri voulu via la valeur d'"actionMode" ("tags" ou "surface"). Il faut indiquer ensuite les mots clés dans refTags (sous la forme : ["tags1","tags2","tags3", ...]) ou la surface dans refBox (sous la forme : [Nord,Sud,Ouest,Est]). Les fiches possédant ces mots-clés ou s'appliquant sur cette surface seront conservées. Les autres seront supprimées.

Fichiers du dépôt :

- Queries : comprend toutes les requêtes xquery qui ont été utilisées.
- Scripts Tulip : comprend l'ensemble des scripts à exécuter dans Tulip.
- Fichiers Tlp : à ouvrir sur Tulip avec file->open. Contient le graphe déjà construit via les scripts python sur le jeu de données PIGMA.
- Parsing : comprend le script de parsing utilisé avant la phase de création des scripts graphe ainsi que le résultat du parsing. Le script établit un parsing des mots en majuscules des champs généalogie. Une exploitation plus efficace est disponible via le script genealogy.py dans le fichier Scripts Tulip qui s'applique sur un graphe Tulip et qui effectue en outre une analyse fréquentielle des mots pour déterminer les éléments importants pour chaque champ généalogie. Ce script est extrêmement lourd en calcul et peut prendre un temps considérable.
- Result : résultats des différentes requêtes xquery utilisées sur les données de PIGMA afin de déterminer la meilleure direction à prendre pour la création des scripts graphe.
- Code existant : code donné lors du début du projet.
- Documentation : documentations données lors du début du projet.