

Guide du VGI

Notes : Il faut charger la librairie Qt et l'installer pour exécuter les logiciels

Construction de la base de données

J'ai utilisé le logiciel MRI de DOAN Tuan Anh pour construire la base de données. Les fichiers sortis du logiciel MRI sont situés dans deux répertoires MS et SA avec les fichiers suivants :

- `igroups.data` : contenir un seul chiffre. C'est la quantité de clusters
- `image.data` : contenir les chemins des images. Chaque image est un sac de mots.
- `index.data` : chaque ligne est correspondante avec un mots dans la dictionnaire et chaque chiffre dans la ligne est l'indice de sac de mots.
- `wgroups.data` : chaque ligne est correspondante avec un cluster et chaque chiffre dans la ligne est l'indice de mots dans la dictionnaire.
- `n` fichiers `.data` (`n` est la quantité des images dans la base d'images) : contenir les données suivantes :
 - la première ligne : la quantité des mots visuels
 - la deuxième ligne : la quantité des mots visuels dans le sac de mots
 - `n` prochaines lignes (`n` est la quantité des mots visuels) : chaque ligne contient 8 chiffre. Le premier chiffre est l'indice de mots dans la dictionnaire. Le deuxième est la valeur `tf_idf`. Le troisième est la quantité de régions ou contient ce mot. Cinq chiffres dernier ont pour but de dessiner cette région.
 - la dernière ligne : le premier chiffre est la quantité de cluster qui cette image appartient, les autres chiffres sont les indices de clusters.

Mon modèle a besoin des fichiers suivants :

- Tout d'abord, une base d'images ou contient des images. J'utilise un fichier `db.conf` pour configurer la base d'images. Ce fichier contient des informations suivantes : le premier chiffre est la quantité des images dans la base d'images, et les chemins des images.
- Le fichier `igroups.data`
- Et `n` fichiers `.data` (`n` est la quantité des images)

Les fichiers de configuration

Le fichier `vgi`

Mon modèle utilise un fichier de configuration `vgi` pour stocker les chemins nécessaires des fichiers précédents. Il y a deux versions de fichier `vgi`. La version 1.0 ne supporte pas les sous-clusters. La version 2.0 les supporte.

Le contenu du fichier `vgi` de la version 1.0 est comme suivant :

```

vgi1.0
//
// Fichier de configuration pour VGI
//
// chemin du fichier de configuration de la base d'images
image /home/manleviet/TPE/Database/BW/db.conf
// chemin du repertoire de MS
databaseMS /home/manleviet/TPE/Database/database01/MS
// chemin du repertoire de SA
databaseSA /home/manleviet/TPE/Database/database01/SA
// dans les repertoire MS et SA, on doit contenir les fichiers suivants :
// 1. igroups.data
// 2. les fichiers data des images

```

Il contient trois chemins : un chemin du fichier db.conf, un chemin du répertoire MS et du répertoire SA.

Le contenu du fichier vgi de la version 2.0 est comme suivant :

```

vgi2.0
//
// Fichier de configuration pour VGI
//
// chemin du fichier de configuration de la base d'images
image /home/manleviet/TPE/Database/BW/db.conf
// chemin du repertoire de MS
databaseMS /home/manleviet/TPE/Database/database01/MS
// chemin du repertoire de SA
databaseSA /home/manleviet/TPE/Database/database01/SA
// dans les repertoire MS et SA, on doit contenir les fichiers suivants :
// 1. igroups.data
// 2. les fichiers data des images
cluster 0 /home/manleviet/TPE/Database/cluster0/MS-C0
/home/manleviet/TPE/Database/cluster0/SA-C0
cluster 1 /home/manleviet/TPE/Database/cluster1/MS-C1
/home/manleviet/TPE/Database/cluster1/SA-C1
cluster 2 /home/manleviet/TPE/Database/cluster2/MS-C2
/home/manleviet/TPE/Database/cluster2/SA-C2
cluster 3 /home/manleviet/TPE/Database/cluster3/MS-C3
/home/manleviet/TPE/Database/cluster3/SA-C3
cluster 4 /home/manleviet/TPE/Database/cluster4/MS-C4
/home/manleviet/TPE/Database/cluster4/SA-C4
cluster 5 /home/manleviet/TPE/Database/cluster5/MS-C5
/home/manleviet/TPE/Database/cluster5/SA-C5
cluster 8 /home/manleviet/TPE/Database/cluster8/MS-C8
/home/manleviet/TPE/Database/cluster8/SA-C8
cluster 9 /home/manleviet/TPE/Database/cluster9/MS-C9
/home/manleviet/TPE/Database/cluster9/SA-C9
cluster 10 /home/manleviet/TPE/Database/cluster10/MS-C10
/home/manleviet/TPE/Database/cluster10/SA-C10
cluster 11 /home/manleviet/TPE/Database/cluster11/MS-C11
/home/manleviet/TPE/Database/cluster11/SA-C11

```

Il y a encore les lignes **cluster**. Chaque ligne comprend une indice du cluster, un chemin du répertoire MS de ce cluster, un chemin du répertoire SA de ce cluster.

J'ai implémenté le logiciel confdb (le code source dans le répertoire ConfDatabase.zip) qui permet de créer le fichier db.conf et le logiciel vgiinput (le code source dans le répertoire VGIInput.zip) qui permet de créer le fichier vgi de version 1.0.

Le fichier vml

Puisque l'initialisation du graphe d'images est trop lent (il perd beaucoup de temps pour créer les liens), j'ai proposé de créer autre type de fichier de configuration sous forme xml qui stocke les définitions des images, des clusters, des liens.

Autres outils

Visual Word Viewer (VisualWordViewer.zip)

C'est un logiciel pour voir les mots visuels sur les images. Tout d'abord, on charge une image. Ensuite, on charge le fichier data correspondant. Le logiciel va dessiner les mots visuels sur l'image. L'utilisateur peut enregistrer l'image sur le fichier. On peut changer la couleur dessinée.

Draw Visual Word (DrawVisualWord.zip)

C'est un logiciel qui peut traverser un répertoire des images pour dessiner les mots visuels sur les images.

Le logiciel VGI

Les fichiers dans le répertoire Config

- data01.vgi : un fichier de configuration de version 1.0
- data02.vgi : un fichier de configuration de version 2.0
- data01.vml : un fichier de configuration sous forme de vml
- data01complet.tlp
- data01simple.tlp
- data02simple.tlp : trois fichiers tlp pour visualiser le graphe sur le logiciel Tulip

Le répertoire images

Il contient des images nécessaires

- images.png : l'image des clusters (maintenant, le modèle n'utilise pas cette image)
- transparent.png : l'image du font. Le modèle l'utilise pour dessiner les trois première images du cluster

Les fonctionnalités

- charger des données en se basant sur un fichier de configuration vgi (version 1.0 ou version 2.0)
- enregistrer le graphe sur le fichier de configuration vml
- enregistrer le graphe sur le fichier tlp (fichier du logiciel Tulip). On peut enregistrer sous deux types de graphes : le graphe complet (les images et les liens entre les images) et le graphe simple (les clusters, les images et les liens entre clusters et images)

- voir le graphe sous forme un arbre (View TreeGraph)
- changer la façons de voir la relations entre des images (deux boutons radios Images et Clusters)
- afficher trois première images du cluster
- les liens entre les clusters ont la différente taille
- peut changer le focus dans le premier niveau du graphe
- les liens entre les clusters et les images ont la différente couleur, cad chaque cluster ont la couleur propre
- supporter seulement huit niveaux de la mise en page spirale
- supporter les sous-clusters pas dynamique

Les fonctionnalités prévues

- charger des données en se basant sur un fichier de configuration vml
- quand on clique sur une image ou un cluster, cette image ou ce cluster doit indisponible
- résoudre l'événement clique et double-clique
- voir une image avec la taille originale, cad quand on clique le bouton droit, l'image original est affiche
- utiliser les processus légers pour charger les images
- changer la taille de la fenêtre
- quand la quantité des images est très grand, il est difficile a voir les images. On peut créer les sous-clusters dynamiquement

Les classes

J'ai organisé le logiciel en quatre modules :

1. **GraphImage** : Ce sont les classes qui servent à créer le graphe des images. Il comprend trois classes : *graphimage*, *image* et *line*. La classe *graphimage* est un graphe des images. Elle contient des images, des clusters, des sous-clusters, des liens entre eux.
2. **IO** : Ce module comprend des classes qui servent à lire des fichiers de configuration et à enregistrer le graphe sur des fichiers tlp ou vml. La classe *graphtlp* a pour but d'enregistrer le graphe sur un fichier tlp. La classe *graphxml* a pour but d'enregistrer le graphe sur un fichier vml et de lire un fichier vml (dans le futur). La classe *loaddatabaseconfig* a pour but de lire un fichier vgi.
3. **UI** : Il contient quatre classes d'interface : *mainwindow*, *choosefiledialog*, *savefiledialog* et *graphtreeview*.
4. **Scene** : Il comprend des classes qui servent à afficher le graphe d'images sur l'écran. La classes *graphscene* est un scene qui contient les niveaux de graphe. La classes *levelscene* est un niveau qui contient les images du niveau. La classe *imagescene* est une image.

De plus, j'ai une classes *util* qui comprend des fonctions utiles et statiques.