ReadMe-TP2

Mouad Belbey

Ce code a pour but de construire un système de recherche d'information (engin de recherche) simplifié. Ce système indexe un ensemble de textes (documents), construit une structure d'index et l'utilise pour retrouver les textes correspondant à une requête de l'utilisateur. L'engin de recherche fait deux traitements : Indexation des documents en mode hors ligne (offline) et traiter une requête de l'utilisateur en ligne. Pour cela plusieurs fonctions ont été créées :

StructureIndex.java : le but de cette fonction et de créer une structure de type liste chaînée de « i » index de documents (**NoeudDoc.java)**, chaque document contient une liste chainée et composée de nœuds (**NoeudMot.java)** qui stocke les mots et leurs fréquences. **Pour cette structure, on a choisi de trier les mots du document 1 fois à la fin de son traitement.**

StructureInverse.java : l'objectif de cette fonction est de permettre une recherche accélérée pour cela on crée une structure inverse de type liste chaînée de « i » index de mots (**NoeudMot.java)**, chaque mot contient une liste chainée et composée de nœuds (**NoeudDoc.java**) qui stocke les documents et leurs fréquences. **Pour la structure inverse, on a décidé de trier par insertion au fur et à mesure car avec la structure indexation, la liste des mots est déjà triée, donc avec la méthode trier par insertion ce sera plus efficace et prendra moins de temps.**

***Par défaut quand on lance le programme, il n'y a pas de documents indexés, donc le programme va ouvrir d'abord l'interface Administrateur pour indexer les docs choisis en premier avant de commencer la recherche !

Main.java : la classe main lance le code et affiche l'interface graphique pour Administrateur pour indexer les documents avant de faire des recherches.

InterfaceUser.java : Permet de trouver un mot, une chaine de mot ou le texte au complet. On peut afficher aussi le contenu du document trouvé en sélectionnant la ligne du tableau de résultat. Pour trier les résultats selon le score décroissant, on a appliqué la **méthode mergeSort** vu au cours (car c'est une méthode efficace et prend moins de temps !)

- Menu :
  - Pour utilisateur :
    - Recherche :
      - JLabel : Requête
      - JTextField : entrer les mots désirés.
      - JButton : recherche :
        - Deux textArea :
          - Le 1er cliquable affiche l'index du document et son score.
          - Le 2eme affiche le texte du document selectionné.
  - Pour administrateur :
    - Administrateur : InterfaceAdmin.java

InterfaceAdmin.java : Permet de indexer un ou plusieurs document de type Texte (.txt) avec FileChooser. On a limité les choix de FileChooser avec FileFilter pour éviter les erreurs de mal cliqué sur un fichier de type autre que .txt. Interface affiche aussi la **Structure Indexation** et la **Structure Inverse** une fois ces documents choisis sont traités. Ce fait **permet à l'administrateur de mieux gérer les documents traités ainsi de tester la fonction recherche plus facilement.**

- Menu :
    - Pour administrateur :
        - Administrateur :
            - JButton : Parcourir : affiche une page pour sélectionner les fichiers(.txt) à traiter.
            - Structure d'indexation : JTextArea qui affiche la structure d'index de documents, les mots et leurs fréquences.
            - Structure inverse : JTextArea qui affiche la structure inverse, l'index de mots, les documents et leurs fréquences.

ExtensionFilter.java : selon les consignes les seuls documents traiter sont de type (.txt). Cette classe permet d'ajouter une extension pour FileFilter