

Détermination du meilleur chemin dans un arbre recouvrant maximal

Professeur référent : Michel Chilowicz <chilowi@u-pem.fr>

Lorsque nous disposons d'un arbre recouvrant maximal (MST), nous souhaitons déterminer le meilleur chemin entre deux mots présents dans cet arbre. Nous réalisons pour cela un programme Java fin de réaliser cette tâche.

Parcours de l'arbre

Une fois que nous avons précalculé pour un modèle donné un MST ou lorsque nous disposons d'un MST pour une partie donnée, nous pouvons l'utiliser pour trouver le meilleur chemin entre deux mots présents dans cet arbre. Pour cela nous parcourons l'arbre pour trouver le chemin liant les deux mots. Nous avons la garantie qu'un tel chemin existe et que ce chemin est unique. D'autre part ce chemin fait partie des meilleurs chemins (pour l'ensemble de mots considérés dans l'arbre) : il n'existe pas d'autre chemin dont la similarité minimale soit supérieure à ce chemin.

Le parcours de l'arbre peut être réalisé au choix en largeur ou en profondeur.

Implantation

Vous devez réaliser un main lisant sur l'entrée standard l'arbre recouvrant maximal. Les deux arguments passés au *main* sont le mot de démarrage et le mot d'arrivée. Il sera retourné sur la sortie standard (en utilisant le format textuel de données que vous avez conçu pour représenter les MST) le chemin liant les deux mots indiqués. On retournera également le score associé à ce chemin, i.e. la similarité minimale entre deux mots consécutifs sur le chemin.

Cette implémentation devra être réalisée en langage Java.

Pour réaliser l'implémentation, vous devrez charger en mémoire l'arbre recouvrant maximal avec une structure de données adéquate puis vous devrez parcourir l'arbre pour trouver tous les noeuds présents sur ce chemin.