JUNIA ISEN

CIR 3

Théorie des langages

TP N° 1

Nacim Ihaddadene / Lise-Marie Veillon

Objectif :

Réaliser trois programmes :

- Un correcteur d'orthographe
- Un moteur de suggestion
- Une carte des probabilités d'adjacence entre les lettres dans la langue française

Le développement de ce TP se fera en langage C++ :

- Utiliser les concepts orientés objet : Les classes, les constructeurs, …
- Utiliser les nouveaux mots clés du C++ : auto, nouvelle forme du for, etc…
- Faire appel aux librairies STL <vector>, <map>, <fstream>, …
- Utiliser la classe std::string et les fonctions de manipulation des chaines de caractères.

Liste des mots français :

La première partie du TP consiste à écrire les procédures permettant le chargement de l'ensemble des mots à partir d'un fichier et de la stocker en mémoire dans un format adapté pour l'étape suivante. La structure de données std ::vector permettra par exemple un accès facilité.

- Télécharger le fichier de mots dela-fr-public.zip à partir de Teams.
- Charger (avec un programme C++) l'ensemble des mots de ce fichier (ainsi que les autres informations disponibles dans chaque ligne).

Une ligne du fichier se présente sous la forme :

```
irai,aller.V+z1:F1s
```

qui veut dire que la racine du mot « irai » est « aller » et qu'il s'agit d'un verbe (V) conjugué au Futur (F) à la première personne (1) du Singulier (s).

Réalisation d'un correcteur d'orthographe.

Pour ce premier exercice, le travail consiste à écrire un programme en C++ qui permet de proposer des suggestions de correction à un mot lexicalement erroné (erreur d'orthographe).

Pour cela nous nous baserons sur le fichier de lexique français et une fonction mathématique permettant de calculer l'écart entre deux mots.

La distance de Levenshtein est une distance mathématique donnant une mesure de la similarité entre deux chaînes de caractères (voir les détails https://fr.wikipedia.org/wiki/Distance_de_Levenshtein)

Le code en C++ de cette fonction est disponible ici :

https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance#C.2B.2B

- Intégrer et tester cette fonction dans le programme.

Pour un mot saisi par l'utilisateur, le programme précisera si ce mot est correct ou proposera une liste de mots proches en utilisant la fonction de distance.

- Ecrire l'interface utilisateur qui permet de saisir en boucle un mot depuis le clavier.
- Comparer le mot saisi à tous les mots de la liste (avec la fonction distance).
- Si le mot est correct, afficher les informations disponible (racine et autres informations)
- Etablir un seuil pour la distance pour ne choisir que des mots proches.

Réalisation d'un outil de suggestion de mots pour la saisie.

Proposer les mots dont la partie qui vient d'être saisie est préfixe, dans la limite de 10 mots.

Exemple : Pour « Bon », proposer « Bonjour », « Bonsoir », « Bonheur », « Bonus », …

Réalisation d'une carte de probabilité :

Cette carte permet de voir la probabilité qu'une lettre suive une autre dans un mot de la langue française. A cela s'ajoute les probabilités de trouver cette lettre en début ou en fin de mot. Cette carte est utilisée dans plein de domaines tels que la reconnaissance de caractère ou la linguistique.

Il s'agit donc d'une matrice de 26 (lettres) x 28 (lettres + début + fin) à remplir en analysant tous les mots du fichier. Pour chaque mot, parcourir toutes les lettres en mettant à jour la cellule adéquate. A la sortie, il faut exporter la matrice dans un fichier CSV et visualiser sur un tableur (Excel).