

# Indexation et Recherche d'Information

## Correcteur orthographique

LO 17

### 1 Objectif

Vous devez réaliser une application **java** qui permettra d'associer un lemme à chacun des mots d'une phrase saisie au clavier. Pour cette première étape, vous définirez manuellement un fichier d'une ou deux dizaines de mots et de lemmes judicieusement choisis pour tester vos algorithmes.

L'analyseur doit :

1. saisir la phrase ;
2. extraire les mots un à un jusqu'à la fin de la phrase (on utilisera la classe **StringTokenizer**) et, pour chaque mot, appeler une méthode d'un objet de type « lexique » qui réalisera les fonctions suivantes :
  - (a) le convertir en minuscules (**nomdunestring.toLowerCase()**) (cette méthode peut faire partie de la classe appelante) ;
  - (b) tester si le mot existe dans le lexique et si oui retourner le lemme ;
  - (c) si non compter le nombre de lettres communes entre le début du mot et le début d'un mot du lexique (algorithme de recherche par préfixe vu en cours) et si ce nombre est suffisamment grand :
    - i. stocker le lemme dans une liste de lemmes candidats ;
    - ii. retourner la sous-liste des meilleurs lemmes candidats ;
  - (d) si non test de Levenshtein et si le test est positif :
    - i. stocker le lemme dans une liste de lemmes candidats ;
    - ii. retourner la sous-liste des meilleurs lemmes candidats ;
  - (e) si non indiquer qu'aucun mot n'a été trouvé.

Lorsque vos algorithmes ont été testés, appliquez votre programme au lexique complet des lemmes du corpus.

### 2 Outils et méthodes

Pour vous aider en ce qui concerne les entrées/sortie en **java** vous trouverez les classes **Cat.java** et **Saisie.java** sur le site de l'UV.

On vous recommande de commencer par construire un objet « lexique » qui lit un fichier dictionnaire et qui possède une méthode permettant d'accéder au lemme d'un mot donné en entrée. Puis vous pourrez spécialiser cet objet pour qu'il réalise l'ensemble des fonctions souhaitées.