Roots : manager de corpus annotés

Introduction

Au sein de notre système, après avoir annoté les différents documents écrits et oraux, nous auront à la sortie du système d'annotation des fichiers de divers formats. Cette hétérogéneité conduit, non seulement à un travail long de conversion et d'alignement des différentes informations, mais aussi à une perte d'informations dans certaines étapes ce qui peut impliquer des erreurs dans les résultats.

Afin, de regler ce problème, nous utiliserons l'outil Roots *(Rich Object Oriented Transcription* System) qui sert à gérer ces différents données et de les homogéneiser .

1.Architecture de ROOTS

Roots est un outil basé sur des relations matricielles. Il a été implémenté, dans un premier temps en Perl, puis en C++. Cette dernière impléméntation de Roots a construit de lui un outil assez complet et apte à manipuler de différents données annotés nottament les informations écrites et orales.

1.1 Structuration d'un corpus dans Roots

En Roots, un corpus est représenté par N uttérances, chaque uttérance est constituée de K séquences et chaque séquence est composée de M items.

Les items peuvent représenter différent types de données: graphèmes, phonèmes, POS...etc. C'est ainsi que les séquences peuvent avoir différents niveaux d'annotation. Cependant, cette hétérogéneité des données nécessite de trouver des correspondances entre les divers items des différentes séquences. On doit aussi pouvoir établir des connexions entre des items qui ne sont pas en relation directe.

Pour ce faire, les correspondances seront représentées sous forme de matrices qui donneront par la suite un graphe dont les noeuds sont les items, ce qui facilitera de trouver les plus courts chemins entres les items de toutes les séquences .

Un corpus peut etre représenté ausssi par des *chunks* (divisions verticales) représentant des sous-corpus et des *layers* (divisions horizontales) représentant des couches d'informations où chaque couche représente des informations de meme type.

1.2 Fonctionnalités mises à disposition

Gràce à sa bibliothèque, Roots permet d'effectuer plusieurs opérations sur les éléments de l' hiérarchie de corpus. Par exemple, on peut ajouter et supprimer des uttérances et des séquences, ajouter de nouvelles relations entre les items et sauvegarder et charger le corpus.

Plusieurs, autres fonctionnalités très utiles s'ajoutent sur celles citées ci-dessus.

1.3 Stockage

Dans Roots, on ne stocke pas les informations en des bases de données, par contre, on les met dans des fichiers. Un corpus sera sauvegardé comme un fichier contenant tous les items et les relations qui les lient.

Afin de simplifier les opérations de lecture et d'écriture, si le corpus est sous forme de sous-corpus ou bien de couches d'information, un fichier est creé par subdivision. De cette façon là, on peut cibler les portions qui nous sont utiles du corpus tout entier.

Parmi les avantages d'une telle structuration de stockage, on peut effectuer un test d'un programme sur une portion du corpus, et puis lancer tout le corpus.

2.Autres Outils de traitement de données annotés

2.1 GATE (*General architecture for text engineering*)

2.1.1 Avantage

Gate permet de comparer les annotations et de mesurer la précision .

2.1.2 Inconvénient

Gate est basé sur un environnement de dévellopement integré, ce qui rend difficile l' intégration des outils extérieures .

2.2 NXT

2.2.1 Avantage

NXT permet de gérer des annotations linguistiques multiples .

2.2.2 Inconvénients

- NXT rassemble les annotations dans une base de données, et y permet l'accés par des requetes.
- NXT n'inclue pas l'établissement des relations entre les annotations .

2.3 UIMA

2.3.1 Avantage

Uima permet d'effectuer le traitement de grands corpus et de les annoter .

2.3.2 Inconvénient

Uima est destiné pour des développement industriels : il permet d'effectuer des traitements facilement et rapidement mais avec plus de cout en le comparant à Roots.