Constitution et traitement de corpus

Nous allons voir comment constituer et traiter un corpus pour une chane de traitement complte de de Traitement Automatique des Langues. Nous allons reconstituer un corpus dont les éléments sont décrits dans un fichier JSON disponible à l'adresse suivante : `https://daniel.greyc.fr/public/corpus_daniel.tar.gz`. Pour l'instant nous n'avons besoin que du fichier `daniel.json`
Notre première tâche sera de récupérer un maximum de fichiers parmi ceux décrits dans le JSON. Nous cherchons uniquement le source HTML, nous n'avons besoin ni des images ni des CSS ou du JAVASCRIPT. Pour ce faire, nous disposons de l'url de chaque fichier. Plusieurs méthodes sont possibles, la plus simple étant probablement d'utiliser WGET sous UNIX. Des librairies sont également disponibles dans différents langages : URLLIB, cURL, URLCONNECTION...

(a) Nettoyage du corpus

Nous avons récupéré les sources HTML mais pour des tâches de TAL, nous sommes uniquement intéressés par la partie textuelle de ces pages. Pour extraire la partie purement textuelle des documents nous pouvons avoir recours à des expressions régulières mais ce n'est pas très efficace. Nous allons donc utiliser des outils spécifiques, à savoir des outils de nettoyage de page web (*boilerplate removal*). Vous choisirez au moins deux outils (de manière à pouvoir les comparer) parmi les outils disponibles en ligne. En voici une liste non exhaustive :

- JUSTTEXT (Python) : `http://corpus.tools/wiki/Justext` téléchargeable à l'url suivante : `http://corpus.tools/raw-attachment/wiki/Downloads/justext-1.2.tar.gz`
- BOILERPIPE (Java ou API) : `https://boilerpipe-web.appspot.com/`
- READABILITY (Java ou autres langages) : `https://github.com/karussell/snacktory`
- NCLEANER (Python) : `http://sourceforge.net/project/showfiles.php?group_id=209325&package_id=268780&release_id=586691`
- HTML2TEXT : outil intégré dans de nombreux systèmes UNIX ...

(b) L'évaluation : Qualité du nettoyage

Comment connaître la qualité des outils ? Une méthode d'évaluation intrinsèque consiste à vérifier que le contenu textuel est bien conservé et qu'un minimum de bruit (publicités...) est présent. Nous utiliserons une implantation en PYTHON fondée sur la distance d'édition qui est disponible à l'adresse suivante : `http://sourceforge.net/project/showfiles.php?group_id=209325&package_id=268779&release_id=586694`. Pour évaluer nous avons besoin d'une vérité de terrain. Cette vérité de terrain se trouve dans le dossier `files` de l'archive dont nous avons extrait le Json en début de TP. Nous allons comparer les outils choisis :

- Au niveau global : lequel est le plus performant au niveau du corpus ;
- Au niveau local : quels sont les différences (langues, sources) entre les outils.

Pour visualiser les résultats, vous générerez des tableaux HTML placés dans un dossier `resultats`. Les résultats globaux, par langue et par domaine (`bbc`, `express`...) devront également y être présentés. Pour chaque fichier traité il faudra faire apparaître les résultats (Rappel, Précision, F-Mesure) pour chacun des outils utilisés.

Une archive contenant le code, les fichiers Html extraits , leur version nettoyée ainsi que les résultats devra être fournie à la fin de la séance. En particulier il faudra pouvoir comparer les rsultats pour chaque langue et pour chaque document (variations de la classe donnée par l'API en fonction du type de nettoyage).