CES Data Scientist, Télécom ParisTech

Index inversé sur HBase

Pierre Senellart (pierre.senellart@telecom-paristech.fr)

16 décembre 2014

Le but de ce TP est de créer à partir des articles de Simple English Wikipedia (stockés dans HBase) un index inversé, lui aussi stocké dans HBase. Dans ce TP, nous contruirons cet index de manière centralisé. Nous verrons dans une session ultérieure comment paralléliser la construction de l'index. Référez-vous au TP précédent pour la configuration de votre environnement.

Lisez l'ensemble du TP avant de commencer à coder. Construisez progressivement votre application, et testez au fur et à mesure que celle-ci fonctionne correctement. Pour évaluation de la session, vous devez envoyer à Pierre Senellart <pierre.senellart@telecom-paristech.fr> votre programme Python à la fin de la séance de TP.

Implémentez la construction d'un index inversé des articles de Wikipedia, qui doit avoir les fonctionnalités suivantes, pas nécessairement codées dans cet ordre :

— Chaque article de Simple English Wikipedia doit être traité tour à tour ; pour des raisons de performance, pendant le développement, limitez-vous aux 1000 premiers articles.

— Pour le découpage du terme en termes (*tokenization*), on pourra utiliser la commande suivante : `it = re.finditer(r"\w+",text,re.UNICODE)` qui met dans `it` un itérateur vers l'ensemble des suites de caractère alphanumériques d'une chaîne de caractères `text`.

— Les mots vides (stop words) ne sont pas ajoutés dans l'index ; se reporter à la liste de mots vides fourni sur le site pédagogique.

— Les termes sont racinisés avec l'algorithme de racinisation de Porter (déjà implémenté dans le module Python `stemming`, que vous pourrez installer).

— Chaque terme doit être accompagné de son score tf–idf.

— Vous devez créer une table HBase ayant pour nom votre nom de famille, afin d'y stocker l'index. **N'écrivez pas dans quelque autre table que ce soit, pour éviter tout conflit avec les autres apprenants.**

— Ne pas oublier de vider cette table chaque fois que vous redémarrez le programme (le plus simple est de la supprimer et de la recréer avec les méthodes HappyBase adéquates).

— Les mises à jour sur la base doivent se faire en *batch* pour des raisons de performance (voir la documentation de HappyBase).

— HappyBase stocke des chaînes binaires d'octets, tandis que Python s'attend à manipuler des chaînes de caractère. Pour passer de l'un à l'autre et réciproquement, vous pouvez utiliser `octets.decode('utf-8')` et `caracteres.encode('utf-8')`.

— Développez un petit programme annexe vous permettant de consulter les informations stockées dans l'index sous HBase.

Une fois le développement terminé, tentez d'enlever la limitation aux 1000 premiers articles. Que constatez-vous ?