M2 GIL - Projet Big Data & Fouille de Données

26 novembre 2019

Contexte général

On s'intéresse ici à l'extraction de connaissances à partir de données non structurées et semi-structurées. Le format des données non structuré est du texte ou des documents XML qu'il faut pré-traiter. L'application de la méthode retenue se fera sur plusieurs corpus de textes en entrée (au format XML, résumés d'articles scientifiques en anglais) à partir desquels on souhaite obtenir des associations entre concepts ou termes biomédicaux.

Le projet se décompose en deux parties :

1. Pour la partie Big Data il s'agira de constituer des collections et de comparer différents SGBD entre eux sur les critères de performances (p.ex. temps de chargement, export, requêtes) ;

2. Pour la partie Fouille de données, il s'agira d'extraire des connaissances à partir des données générées dans la partie Big Data.

Modalités de rendu

Chacune des étapes (mots clés, constitution des corpus, annotation des résumés et titres avec TreeTagger, extraction d'entités nommées (term list) avec YaTea, extraction des règles d'association (avec Close ou Apriori ou autre algo de fouille sous Weka) etc...) seront détaillées dans un rapport PDF soigné.

Les modules développés, les résultats intermédiaires (collections JSON, tableau comparatif entre les SGBD ...etc.), sorties de Yatea, fichiers de règles d'associations...etc. seront également à remettre.

Remise : soutenance (15 min par groupe) le **13 janvier 2020** ; rapport finalisé et autres livrables à remettre le **24 janvier** au plus tard (lien FileX ou DropBox actif 4 semaines).

**Pour les étudiants en alternance : seules les questions en italique seront à traiter (1.1.1, 1.1.2 et Étape 5).

1 Big Data : import/export, requêtage de données

Étape 1 : Constitution des collections au format XML

Les fichiers XML devant être traités sont des extractions au format XML MEDLINE, de la banque de données des articles scientifiques en santé de la NLM (National Library of Medicine : http://www.ncbi.nlm.nih.gov/pubmed).
Les requêtes devant être lancées par chaque groupe sont fournies en annexe (chaque binôme/trinôme choisira 1 ensemble de mots clés, les 2 mots clés devront être traités, afin de constituer 2 corpus de textes).
NB : les fichiers XML résultant des requêtes sont de taille conséquente. Il est recommandé de prévoir une limite supérieure du nombre d'articles traités (p. ex. au plus 50 000 articles pour les premiers tests est un bon départ).

1.1 Intégration et gestion dans une base XML native

Pour chaque corpus de documents (c-à-d. pour chaque fichier XML correspondant à chacune des requêtes), il est demandé de l'intégrer dans une base XML native (au choix ExistDB ou BaseX). À partir des 2 collections (1 collection par mot-clé requête incluant toutes informations du fichier XML), générez les collections suivantes :

1. *Une collection par mot-clé requête contenant les "PMID", "MeSH Descriptor"**.*
2. *Une collection par mot-clé requête contenant les "PMID", "MeSH Descriptor/Qualifier"**.*
3. Une collection par mot-clé requête contenant les "PMID", "Title", "Abstract"
4. Une collection par mot-clé requête contenant les "PMID", "Title", "Abstract", "MeSH Descriptor".
5. Une collection par mot-clé requête contenant les "PMID", "Title", "Abstract", "MeSH Descriptor/Qualifier".

1.2 Intégration et gestion dans une base relationnelle/objet

Mêmes questions qu'en 1.1 mais avec Oracle en utilisant le type XML.

Étape 2 : Constitution des collections JSON

2.1 Intégration et gestion dans une base XML native

Pour chaque corpus de documents (c-à-d. pour chaque fichier XML correspondant à chacune des deux requêtes), il est demandé (i) de le transformer au format JSON puis (ii) de l'intégrer dans une base NoSQL orientée documents (MongoDB ou CouchDB). À partir des 2 collections (1 collection par mot-clé requête incluant toutes informations du fichier XML), générez les collections suivantes :

1. Une collection par mot-clé requête contenant les “PMID”, “MeSH Descriptor”.
2. Une collection par mot-clé requête contenant les “PMID”, “MeSH Descriptor/Qualifier”.
3. Une collection par mot-clé requête contenant les “PMID”, “Title”, “Abstract”
4. Une collection par mot-clé requête contenant les “PMID”, “Title”, “Abstract”, “MeSH Descriptor”.
5. Une collection par mot-clé requête contenant les “PMID”, “Title”, “Abstract”, “MeSH Descriptor/Qualifier”.
6. Par MeSH Descriptor inclus dans les 2 collections issues des requêtes, les collections suivantes :
   (a) “PMID”, “MeSH Descriptor”
   (b) “PMID”, “Title”, “Abstract”
   (c) “PMID”, “MeSH Descriptor/Qualifier”
   (d) “PMID”, “Title”, “Abstract”, “MeSH Descriptor”
   (e) “PMID”, “Title”, “Abstract”, “MeSH Descriptor/Qualifier”
7. Par couple “MeSH Descriptor/Qualifier” inclus dans les 2 collections issues des requêtes, les mêmes collections qu’en 6.

(Pour les questions 6 et 7, vous pouvez utiliser Map Reduce).

2.2 Intégration et gestion dans une base relationnelle

Mêmes questions qu’en 2.1 mais avec Oracle en utilisant le type JSON.

Étape 3 : Intégration dans une base orientée graphe

Transformez les données afin de les intégrer dans la base de données Neo4J (https://neo4j.com/developer/). Utilisez le langage Cyper pour les questions 1 à 7 du .

Étape 4 : Comparaison des méthodes d’intégration

Dressez un tableau comparatif entre les différents SGBD que vous avez utilisés pour réaliser les intégrations/transformations/exports.

2 Extraction de connaissances

Étape 5 : Fouille de données

À partir des collections 1, 2, 6.a, 6.c, 7.a et 7.c *extraire les règles d’association entre MeSH Descriptor et entre couples (MeSH Descriptor/Qualifier). Vous pouvez utiliser l’algorithme que vous souhaitez. ***

Étape 6 : Fouille de textes

Étiquetage de textes

Étiquetez les textes des collections contenant les "Title" et "Abstract" (avec Genia Tagger (http://www.nactem.ac.uk/GENIA/tagger/) ou Tree Tagger (http://www.cis.uni-muenchen.de/∼schmid/tools/TreeTagger/)).

Extraction de termes

Utilisez Yatea (https://metacpan.org/pod/distribution/Lingua-YaTeA/bin/yatea) pour extraire les termes candidats à partir des sorties de Tree Tagger ou Genia Tagger.

Règles d'associations

À partir des sorties de Yatea et des collections, extraire des associations :

1. entre termes candidats ;
2. entre termes candidats et MeSH Descriptors ;
3. entre termes candidats et couples MeSH Descriptor/Qualifier ;
4. entre lemmes des termes candidats (deuxième colonne dans le fichier `termList.txt`, ou troisième colonne du fichier `termCandidates.ttg`) ;
5. entre lemmes des termes candidats et MeSH Descriptors ;
6. entre lemmes des termes candidats et couples MeSH Descriptor/Qualifier.

Annexes :

Format des requêtes PubMed :

http ://www.ncbi.nlm.nih.gov/pubmed ?term=[terme]
Liste des éléments XML de MEDLINE :http://www.nlm.nih.gov/bsd/licensee/elements_alphabeti
Description des XML Element de MEDLINE http://www.nlm.nih.gov/bsd/licensee/elements_desc

Liste 1 :

myocardial infarction ; post-operative complications ;

Liste 2 :

micro rna ; escherichia coli ;

Liste 3 :

drug target ; adverse drug events ;

Liste 4 :

protein ligand ; hepatitis e virus ;

Liste 5 :

Zika ; Ebola ;

Liste 6 :

lyphoma ; drug target ;

Liste 7 :

Alzheimer ; rare diseases ;

Liste 8 :

thrombosis ; nosocomial infections ;

Stop words

http ://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T43/