

Examen du cours Recherche d'information / web sémantique
ENSEEIHT 3A Info – année 2014-2015
Mercredi 4 mars 2015

Les documents sont autorisés pour les 2 parties.

Partie « recherche d'information »

(durée indicative : 1h – barème 10 pts)

Exercice 1 (5 pts)

On considère un système de recherche d'information basé le modèle de langue mixte (JM) combinant le modèle de document et le modèle de collection. Les probabilités sont estimées en se basant sur la fréquence des termes dans le document (pour le modèle de document) et la fréquence des termes dans la collection (pour le modèle de collection).

Considérons la requête suivante : q (XML, document), dont les ICF (Inverse Collection Frequency, fréquence relative des termes dans la collection) sont : $10/10\ 000\ 000$ pour le terme "XML" et $10\ 000/10\ 000\ 000$ pour le terme "document".

Soit une collection de 4 documents : d_1 , d_2 , d_3 et d_4 . On suppose que le document d_1 contient 1000 termes dont le terme « XML » qui apparaît une seule fois. On suppose que le document d_2 contient également 1000 termes dont le terme « document » qui apparaît 1 fois. Les documents d_3 et d_4 ne contiennent pas ces deux termes.

Questions :

Donner l'ordre dans lequel les 4 documents seront renvoyés en réponse à la requête q

1. dans le cas d'un modèle de langue mixte JM avec $\lambda=0.5$
2. dans le cas du modèle probabiliste BIR
3. On suppose que la requête q est pondérée q (XML 3, document 1), donner l'ordre dans lequel les documents seront renvoyés dans le cas des deux modèles.

Exercice 2 (5pts)

Soit $C_0 = \{D_1, \dots, D_N\}$, une collection de N documents. On se propose de construire deux autres collections de la manière suivante :

- C_1 formée par suppression de D_1 de C_0 , C_1 a donc $N-1$ documents.
- C_2 formée en remplaçant D_1 de C_0 par un nouveau document D'_1 contenant tous les mots de D_1 mais ils apparaissent tous qu'une seule fois, C_2 a donc le même nombre de documents que C_0 .

Soient t un terme apparaissant plus d'une fois dans D_1 (i.e. $\text{freq}(t, D_1) > 1$) et $\text{IDF}(t, C_i)$ (*Inverse document frequency*) du terme t dans la collection C_i . Soit également $\theta(C_i)$ le modèle de langue de la collection estimée à partir de la collection C_i (estimé par maximum de vraisemblance)

Questions:

- 1- Comparer $\text{IDF}(t, C_1)$ et $\text{IDF}(t, C_0)$? Justifier votre réponse.
- 2- Même question que (a) en prenant $\text{IDF}(t, C_2)$ and $\text{IDF}(t, C_0)$? Justifier votre réponse.
- 3- Comparer $P(t|\theta(C_1))$ et $P(t|\theta(C_0))$? Justifier.

Partie « Web sémantique »

(durée indicative : 1h – barème 10 pts)

Exercice 1 : (5 points)

1. A qui servent les blank nodes en RDF ?

Ce sont des variables, des concepts « vides ».

2. Que « signifient » les 6 triplets RDF ci-dessous ?

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
@prefix foaf : <http://xmlns.com/foaf/0.1/Document> .
```

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
@prefix myVoc: <http://example.org/myVocabulary/> .
```

```
myVoc:doc1 dc:creator "Paul".
```

L'entité doc1 a pour créateur "Paul".

```
"Paul" rdf:type foaf:Person .
```

"Paul" est une personne au sens de foaf.

```
myVoc:doc2 dc:creator _:x .
```

L'entité doc2 a un créateur inconnu _x

```
myVoc:doc3 dc:creator _:x .
```

L'entité doc3 a le même créateur inconnu _x

```
myVoc:doc3 dc:title "La mauvaise réputation".
```

L'entité doc3 a pour titre « la mauvaise réputation »

```
"Paul" foaf:knows _:x .
```

Paul connaît l'inconnu _x.

L'ensemble signifie que Paul, qui est une personne, connaît l'auteur des documents doc1 et doc3, doc ayant pour titre « la mauvaise réputation ».

3. Quelle est la notation de RDF utilisée en question 2 ?

Quels sont les espaces de noms utilisés ?

Notation N3

myVoc, foaf, rdf et rdfs, dc

4. Le vocabulaire myVoc est-il une ontologie ? pourquoi ?

Non parce que on ne connaît ni le type de doc2 et doc3 ni le type de « la mauvaise réputation ».

5. Peut-on écrire une requête SPARQL qui retourne tous les documents de ce vocabulaire et leurs auteurs ? expliquer pourquoi. Si ce n'est pas le cas, rajouter les triplets nécessaires et écrire une requête qui retourne tous les documents dont Paul connaît l'auteur.

Pas vraiment

Non car on ne sait pas que doc2 et doc 3 sont de type document, ou que Paul est un auteur.

- mais oui en qqsorte en utilisant l'expressivité / la sémantique de dc. On peut « deviner » que les entités reliées par L'entité doc3 a le même créateur inconnu _x

dc :creator.

Exercice 2 : (5 ou 6 points)

Soit l'ontologie suivante écrite avec la notation OWL/Turtle

```
:Entité          rdf:type          owl:Class .
:Fonction         rdfs:subClassOf  :Entité .
:Sexe :Genre      rdfs:subClassOf  :Entité.
:Discipline       rdfs:subClassOf  :Entité.
:Cours            rdfs:subClassOf  :Entité.
:CoursUniversitaire rdfs:subClassOf :Cours.
:Personne         rdfs:subClassOf  :Entité.
:Homme rdfs:subClassOf :Personne.
:Femme rdfs:subClassOf :Personne.
:feminin rdf:type :Genre.
:masculin rdf:type :Genre.
:lea rdf:type :Personne .
:coursOntoN7 rdf:type :CoursUniversitaire.
:coursWebSemantique rdf:type :CoursUniversitaire.
:Genre owl:equivalentClass [
    rdf:type owl:Class ;
    owl:oneOf ( :masculin :feminin )
].
:féminin owl:differentFrom :masculin
:aPourGenre
    rdfs:domain :Personne ;
    rdfs:range :Genre .
:Personne owl:equivalentClass [
    rdf:type owl:Restriction ;
    owl:onProperty :aPourGenre ;
    owl:someValuesFrom :Genre
] .
:lea :aPourGenre :féminin.
:Femme owl:equivalentClass [
    rdf:type owl:Restriction ;
    owl:onProperty :aPourGenre ;
    owl:hasValue :féminin
] .
:Homme owl:equivalentClass [
    rdf:type owl:Restriction ;
    owl:onProperty :aPourGenre ;
    owl:hasValue :masculin
] .
```

1. Donner la liste des classes, des propriétés (ObjectProperties) et des instances de cette ontologie.

Classes : Entité, Personne, fonction, homme, femme, genre, Cours, Discipline, :CoursUniversitaire, ObjectProperties : aPourGenre

Instances : masculin, féminin, :lea, :coursOntoN7, :coursWebSemantique

2. Si on lance un raisonneur, quelles classes va-t-il attribuer à l'instance :léa ? pourquoi ?

Personne, Femme et Entité

3. Créer une relation :enseigneCours entre qui indique qu'une personne enseigne un cours ; par analogie avec la définition de :Homme, créer une classe :Enseignant qui serait définie comme une personne qui enseigne au moins un cours, représenter que :léa enseigne le cours Web sémantique et le cours d'ontologie N7.

```
:enseigneCours
  rdfs:domain :Personne ;
  rdfs:range :Cours .
:Enseignant rdfs :subClassof :Personne
  owl:equivalentClass [
    rdf:type owl:Restriction ;
    owl:onProperty :enseigneCours ;
    owl:someValueFrom :Cours
  ] .
:lea :enseigneCours :coursWebSémantique
:lea :enseigneCours :coursOntoN7
```

4. Ecrire une requête SPARQL qui retourne tous les enseignants et leurs cours. Cette requête va-t-elle retourner :léa et les cours qu'elle enseigne ? pourquoi ?

```
SELECT ?ens ?cours DISTINCT {
  ?ens rdf:type :Enseignant .
  ?cours rdf:type :Cours .
  ?ens :enseigneCours ?cours .
}
Non car :lea n'est pas de type :Enseignant
```

5. Si on relance le raisonneur, quelles classes va-t-il attribuer à :léa ? quelles réponses retournera la même requête SPARQL ? qu'en concluez-vous au sujet de SPARQL ?

Il va rajouter la classe :Enseignant car :lea est un des arguments d'une relation :enseigneCours.

La requête Sparql pourra alors retourner

:lea :coursOntoN7

:lea :coursWebSemantique

Pour obtenir plus de réponses à des requêtes SPARQL, il vaut mieux avoir lancé le raisonneur avant d'exécuter ces requêtes.