Examen d'ingénierie linguistique

Master 1 informatique

28 avril 2011

2 heures

Aucun document autorisé

Exercice 1 : Reconnaissance de séquences (3 points)

Un ingénieur cherche à extraire certaines informations dans les courriers électroniques que son entreprise reçoit. En particulier, il s'intéresse aux codes postaux (ex. *77454*) et aux adresses électroniques (ex. *mconstan@univ-mlv.fr*). Pour cela, il utilise un système de reconnaissance qui exploite des expressions rationnelles pour identifier les différentes séquences souhaitées.

1. Ecrire une expression rationnelle reconnaissant les codes postaux.
2. Ecrire une expression rationnelle reconnaissant les adresses électroniques.

Exercice 2 : Evaluation de moteurs de recherche (3 points)

On souhaite comparer la qualité de deux moteurs de recherche M1 et M2 pour répondre à une requête donnée. Les deux moteurs retournent une liste de 10 documents. Ils sont classés dans l'ordre décroissant de pertinence par rapport à la requête. La liste est ensuite manuellement vérifiée: v indique que le document est pertinent, x indique qu'il ne l'est pas. Les résultats sont donnés ci-dessous :

Rang	M1	M2
1	D1 (v)	D6 (v)
2	D2 (x)	D7 (x)
3	D3 (v)	D8 (v)
4	D4 (v)	D9 (x)
5	D5 (x)	D10 (x)
6	D6 (v)	D5 (x)
7	D7 (x)	D4 (v)
8	D8 (v)	D3 (v)
9	D9 (x)	D2 (x)
10	D10 (x)	D1 (v)

1. Calculer la précision de chacun des deux moteurs. *Rappel de cours* : la précision est la proportion de documents retournés qui sont pertinents par rapport à une requête.
2. Calculer la précision au rang 5 pour chacun des deux moteurs.
3. Calculer la précision moyenne pour chacun des deux moteurs. *Rappel de cours* : la précision moyenne est la moyenne des précisions à tous les rangs où le document est pertinent.

Exercice 3 : Classification naïve de Bayes (4 points)

Dans cet exercice, nous simulons un système de classification naïve de Bayes, utilisant deux catégories X et Y. La collection d'apprentissage est :

Document	Catégorie
aabbd	X
abddd	Y
abbbd	X

Rappel de cours: Pour maximiser la probabilité de classer un document *d* formé de la séquence $w_1w_2...w_n$ dans la catégorie *c,* on utilise la formule suivante :
$\max_c P(c|d) = \max_c P(c).P(w_1|c).P(w_2|c)...P(w_n|c)$

1. Simuler la phase d'apprentissage, i.e. estimer les probabilités de base.
2. Simuler la phase de classification sur le document *aabbcc*, i.e. indiquer la catégorie qui sera assignée à ce document (expliquer!).

Exercice 4 : Algorithme d'Earley (5 points)

Soit la grammaire:
P0 → E
E → aEb
E → c

Simuler l'application de cette grammaire sur les séquences *aacbb* et *aabb* avec l'algorithme d'Earley.

Exercice 5 : Alignement (4 points)

Implanter en Python un système d'alignement mot à mot en utilisant l'algorithme du Linking Algorithm vu en cours. Le système prend en entrée deux séquences de mots *e* et *f*. Il retourne une liste de paires de mots (e_i,f_j) où e_i est un mot de e et f_j est un mot de f. On suppose également que nous avons à disposition une classe *SimilarityMatrix* représentant la matrice de similarité des deux séquences de mots. Elle possède plusieurs méthodes:

- *sim*(i,j) qui retourne un réel indiquant la similarité entre les mots e_i et f_j. Plus ce nombre est grand, plus e_i et f_j sont similaires ;
- *remove*(i,j) qui supprime la i^e ligne et la j^e colonne de la matrice ;
- *rowIndexes*() et *columnIndexes*() qui retournent respectivement les indices des mots dans *e* et les indices des mots dans *f*, encore présents dans la matrice.

Le constructeur de cette classe est appellé en passant *e* et *f* en arguments.