Examen d'ingénierie linguistique
Master 1 d'informatique
27 avril 2012

Tous documents autorisés
2h

Exercice 1 : Automates finis (5 points)

a) Quelle est la différence entre un automate fini et un transducteur ?
b) Lequel vous semble le plus adapté pour créer un étiqueteur morphosyntaxique, c'est-à-dire qui donne pour chaque mot d'un texte son étiquette grammaticale ?
c) Donnez une expression rationnelle et un automate fini permettant de reconnaître les adresses mail. Par exemple : plop.plip@essai.com, toto24@essai.com, ou encore 14bla-bla@essai.com sont considérés comme des adresses mail valides, mais pas plop?plip@essai.com (les seuls caractères spéciaux autorisés en plus des lettres et chiffres sont le tiret et le point), ni plop.plip@essai (il manque l'extension de domaine), ni, bien sûr plopessai.com (il manque l'arobase).

Exercice 2 : Classification supervisée de documents (4 points)

On souhaite réaliser une classification supervisée de documents par la méthode des centroïdes. Il existe deux catégories possibles : informatique (I) et zoologie (Z). Le corpus d'apprentissage comprend les textes suivants (classement indiqué entres parenthèses) :

- la souris mange le fromage (Z)
- le chat mange la souris (Z)
- le chat ne mange pas le fromage (Z)
- la souris est devant mon écran (I)
- je souris devant mon écran (I)
- je tape mon message dans le chat (I)

a) On va représenter chaque phrase par un vecteur, pour lequel chaque dimension correspond à un mot. Pour que toutes les phrases soient représentées par des vecteurs dans le même espace vectoriel, combien y a-t-il de dimensions dans cet espace vectoriel ?
b) Donnez la représentation vectorielle de chacune des 6 phrases dans l'espace vectoriel
c) Calculez les centroïdes de la catégorie I et de la catégorie Z.
d) Appliquez la méthode du centroïde pour classer les trois phrases suivantes dans la catégorie I ou Z :
   - le chat ne mange pas la souris
   - je tape la souris devant mon écran

Exercice 3 : Algorithme de tokenisation (5 points)

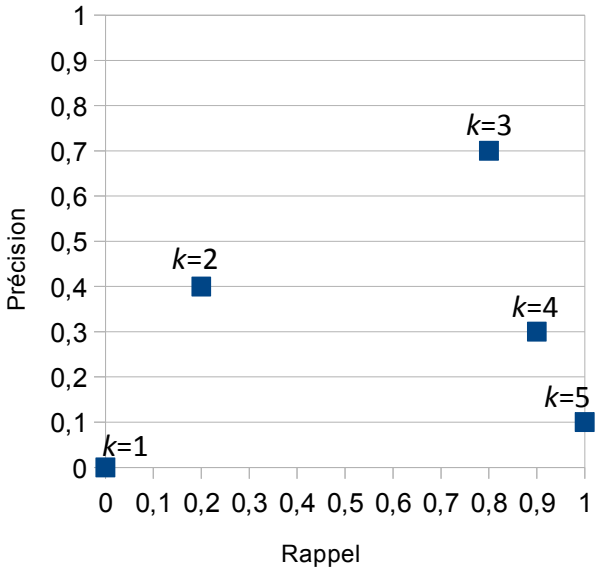
On souhaite tokeniser un texte qui ne contient aucun signe de ponctuation (les mots sont tous séparés par des espaces) en utilisant un dictionnaire. Le texte est contenu dans une chaîne de caractères *texte*. Le dictionnaire est un ensemble de mots, éventuellement composés (par exemple : « aujourd hui » ou « ver de terre ») stockés dans une liste *dico*. Le résultat de la tokenisation sera placé dans une liste *resultat*.

a) Quel est le résultat de la tokenisation de la phrase `"aujourd hui la taupe a mangé un ver de terre"` si le dico est : `["a", "a mangé", "aujourd hui", "de", "la", "taupe", "terre", "un", "ver", "ver de terre"]` ?
b) Écrivez en Python un algorithme qui prend en entrée une chaîne de caractères *texte* et une liste de chaînes de caractères *dico*, et renvoie le résultat de la tokenisation dans une liste de chaînes de caractères.
c) Quelle est la complexité de cet algorithme en fonction de : *n*, le nombre de caractères du texte *texte* ; *m*, le nombre de mots du dictionnaire *dico* ; *k*, le nombre maximum de caractères des éléments de *dico*.

Exercice 4 : Précision et rappel (3 points)

Vous avez à disposition un programme Python **EtiquetteNomPropre** qui prend en entrée un texte tokenisé *texte* (c'est-à-dire une liste dont la case numéro *i* contient le mot numéro *i* dans le texte ; la numérotation commence à 0), et un paramètre de réglage *k* entre 1 et 5. Il renvoie en sortie un tableau de booléens *resultats*, où la case numéro *i* contient True si le mot numéro *i* du texte est un nom propre, False sinon.

Pour évaluer le programme, un testeur a créé manuellement le tableau *resultatsReels1* correspondant aux noms propres d'un texte *Texte1*, et a calculé la précision et le rappel pour les noms propres dans les résultats obtenus par le programme **EtiquetteNomPropre**, pour chaque valeur de *k* possible, pour finalement obtenir le graphique ci-contre.



a) Pour quelle valeur de *k* obtient-on les résultats qui vous semblent les meilleurs ? Justifiez votre réponse en une phrase.

b) On vous signale que pour une certaine valeur de *k*, il serait possible d'améliorer les résultats avec un algorithme trivial. De quelle valeur de *k* s'agit-il ? Donnez les lignes de code nécessaires (en Python) pour implémenter cet algorithme trivial pour cette valeur de *k*.

Exercice 5 : Création d'un dictionnaire des synonymes (3 points)

Une entreprise souhaite créer automatiquement une base de dictionnaire des synonymes usuels en français (qui sera ensuite travaillé par des linguistes) à partir d'un corpus. Elle vous donne un mois pour réaliser et tester un tel programme, en vous fournissant :

- un corpus de sous-titres français, anglais et allemand 3000 films (des fichiers textes où chaque ligne correspond à une réplique, d'au moins 500 lignes par fichier).
- un dictionnaire des synonymes anglais électronique (c'est-à-dire un fichier où chaque ligne contient un mot anglais, suivi de la liste de ses synonymes, tous séparés par des espaces), et un dictionnaire des synonymes allemands électroniques.
- un dictionnaire anglais-français électronique (c'est-à-dire un fichier où chaque ligne contient un mot anglais, suivi de la liste de ses traductions possibles en français) ainsi que des dictionnaires électroniques français-anglais, français-allemand, allemand-français, anglais-allemand, allemand-anglais.

a) Quelle approche choisissez-vous ? Décrivez les différentes étapes en faisant référence à des algorithmes vus en cours ou en décrivant brièvement leur fonctionnement (ce qu'ils prennent en entrée et renvoient en sortie). Précisez pour chaque étape le temps de réalisation prévu.

b) Comment faites-vous pour convaincre l'entreprise de la qualité du travail réalisé ?