Analyse de textes, graphes « petit monde », PageRank, etc.

Préambule

Le choix du langage de programmation est laissé à l'appréciation des étudiants. Une compte-rendu synthétique du TP devra être rendu pour le 9 février 2007.

Si ce n'est déjà fait, récupérer l'archive textes.tgz à l'url `http://www.lif.univ-mrs.fr/~liva/doku.php?id=teaching:20062007:textwebmining`. Il s'agit d'articles de presse récupérés de differents journaux qui concernent les personnages politiques éventuellement candidats à la présidentielle.

1 Construction d'un graphe de proximité

1. Faire une programme qui, étant donné un ensemble de textes, récupère l'ensemble du vocabulaire présent dans ces textes. Utiliser ce programme sur différentes collections de textes parmi les textes proposés et faire un graphe de l'évolution du vocabulaire en fonction de la taille de la collection, puis de la taille totale (nombre de mots) du corpus.
2. En faisant une recherche sur Internet, expliquer en quelques mots ce qu'est la loi de Heaps. Vérifier que l'évolution de la taille du vocabulaire suit bien une loi de Heaps. Estimer (à la main) les paramètres de cette loi pour les textes étudiés.
3. Faire un programme qui transforme les textes en vecteurs : étant donné une collection de textes dont le vocabulaire total est de taille d, la transformation d'un texte de la collection en vecteur consiste simplement à calculer un vecteur de taille d où la i-ème composante correspond au nombre de fois où le i-ème terme du vocabulaire apparaît dans le texte. Faire une autre version où le vecteur est binaire et une autre où les fréquences sont normalisées.
4. Faire un programme qui construit un graphe de proximité entre candidats de la manière suivante : pour chaque candidat, on calcule un vecteur caractéristique à partir des différents textes qui lui correspondent (on peut par exemple considérer qu'un collection de m textes est simplement un grand texte et calculer les fréquences comme précédemment, le vocabulaire étant calculé par ailleurs) et on cherche les k plus proches voisins de ce vecteur au sens d'une norme ou similarité donnée ; ces k voisins constitueront dans le graphe les k sommets adjacents au candidat étudié. On peut également décider de définir les voisins en fonction d'un seuil. Utiliser la distance euclidienne et le cosinus (similarité).
5. Etudier les différents graphes obtenus selon les journaux.
6. Est-ce que les graphes obtenus en utilisant un seuil sont des graphes petit monde ?
7. Calculer les valeurs PageRank des candidats...

2 Analyse sémantique latente et Analyse en composantes principales

Une méthode pour mettre en valeur des relations sémantiques entre termes est l'analyse sémantique latente (vue en cours).

1. Programmer la méthode d'analyse sémantique latente.
2. L'appliquer aux textes des candidats à la présidentielle en ne gardant que 2 ou 3 composantes. Représenter graphiquement les candidats une fois cette analyse faite (On pourra traiter les textes journal par journal).
3. Programmer la méthode d'analyse en composantes principales. Garder 2 axes et représenter les candidats dans l'espace obtenu (On pourra traiter les textes journal par journal).
4. Proposer une méthode pour voir le positionnement des journaux les uns par rapport aux autres.

3 Classification de candidat

(On pourra utiliser Weka.)

- Est-il possible de distinguer automatiquement le candidat dont il est question dans un texte (les noms de candidats ont été enlevés, bien sûr) avec des méthodes de classification supervisée ?
- L'analyse sémantique latente où l'analyse en composantes principales permettent-elles d'améliorer les résultats ?