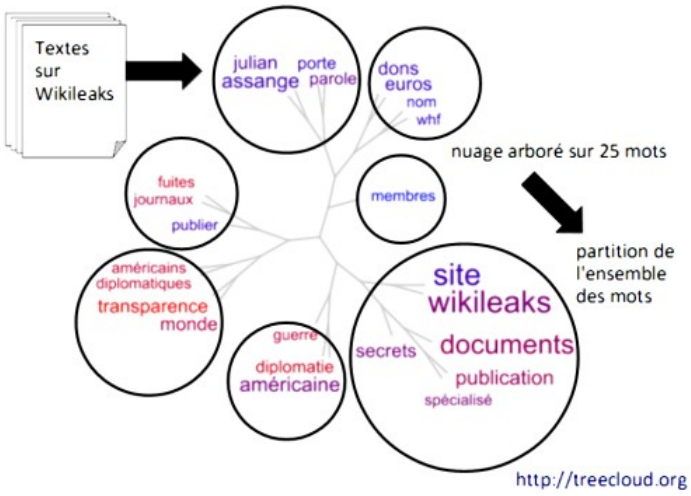
# Projet d'ingéniérie linguistique – *Classes de mots*

## Introduction




Le principe du projet *classes de mots* est de trouver une méthode pour regrouper les mots d'un texte en classes qui reflètent leur proximité dans le texte. Cette proximité indique aussi une proximité sémantique.

Pour cela, il faudra créer un classifieur non supervisé qui parviendra à identifier les regroupements en classes de 25 mots présents au sein d'un texte construit de manière à faire apparaître ces regroupements.

## Étapes du projet

Le projet est découpé en trois étapes, décrites plus en détails ci-dessous :

- tâche 1 : écrire un texte sur Wikileaks faisant apparaître un ensemble prédéfini de 25 mots
- tâche 2 : programmer une méthode de calcul des distances entre ces 25 mots à partir d'un ensemble de textes les contenant
- tâche 3 : programmer une méthode de classification non supervisée, qui permettra de partitionner cet ensemble de 25 mots en classes sémantiques.

Ces étapes devant s'effectuer en équipes de 3 (2 ou 4 à négocier par mail à philippe.gambette@gmail.com si le nombre d'étudiants n'est pas divisible par 3), ces équipes seront constituées, le 18 mars au plus tard, sur le document partagé situé à l'adresse suivante : http://tinyurl.com/ProjetInfoling2012.

Dès que vous aurez choisi une méthode pour les étapes 2 et 3, vous l'indiquerez dans le document partagé afin que les autres groupes puissent faire preuve d'originalité en choisissant une autre méthode que celles déjà mentionnées.

**Étape 1 – Écriture du texte**

Il s'agit d'écrire un texte en français d'au moins 300 mots faisant apparaître les mots de l'arbre en haut à droite de cette feuille (julian, assange, porte, parole, dons, euros, nom, whf, membres, wikileaks, site, documents, publication, spécialisé, secrets, guerre, diplomatie, américaine, américains, diplomatiques, transparence, monde, fuites, journaux, publier), éventuellement en leur ajoutant des majuscules (tout particulièrement Julian, Assange, WHF, et Wikileaks), tout en respectant au mieux les classes suivantes :
[[julian, assange, porte, parole], [dons, euros, nom, whf], [membres], [wikileaks, site, documents, publication, spécialisé, secrets], [guerre, diplomatie, américaine], [américains, diplomatiques, transparence, monde], [fuites, journaux, publier]]

« Respecter les classes » signifie que vous devrez tenter de faire apparaître les mots d'une même classe à proximité les uns des autres dans votre texte. Pour cela, vous pouvez vous inspirer de l'arbre en haut à droite de cette page, et en particulier de ses couleurs : les mots en rouge apparaissent plutôt au début du texte utilisé pour cet arbre, et les mots en bleu plutôt à la fin.

**Étape 2 – Distance entre mots**

A la fin de l'étape 1, un sous-ensemble de textes vous sera fourni, le « corpus de test », qui vous permettra d'évaluer votre méthode de distance et de partitionnement. Ce corpus est appelé « le texte », ci-dessous.

Plusieurs formules sont possibles pour calculer la distance entre deux mots *a* et *b* à partir de leur proximité dans un texte. Une méthode classique est de recouvrir le texte par des fenêtres de mots glissantes, de 10 mots de large, par exemple, puis de compter le nombre de fenêtres qui contiennent *a* et *b*, qui contiennent *a* sans *b*, qui contiennent *b* sans *a*, et enfin qui ne contiennent ni *a* ni *b* (voir cours 2). Plusieurs formules de cooccurrence permettent alors de déterminer une distance entre *a* et *b* à partir de ces quatre valeurs (Stefan Evert, *The Statistics of Word Cooccurrences, Word Pairs and Collocations*, http://elib.uni-stuttgart.de/opus/volltexte/2005/2371/, p. 66 et p. 76 à 83).

Livrable :

- les distances entre paires de mots devront être stockées dans un fichier .CSV, qui est un fichier texte permettant de stocker un document tableur, où la première ligne contient la liste des mots séparés par des « ; » puis chaque ligne suivante contient un mot, suivi d'un « ; », suivi de la liste de ses distances avec les autres mots (nombres flottants notés « à l'américaine » : 1.5, et pas 1,5) séparées par des « ; ».
- le script Python, nommé `creeMatrice.py`, devra prendre un seul argument, l'adresse du fichier texte de type *[adresse].txt*, et écrire le fichier .CSV décrit ci-dessus à l'adresse *[adresse].txt.csv*.

Exemple du début d'un fichier CSV décrit ci-dessus :

;julian;assange;porte;parole;dons;euros;nom;whf;membres;wikileaks;site;documents;publication;spécialisé;secrets;guerre;diplomatie;américaine;américains;diplomatiques;transparence;monde;fuites;journaux;publier

julian;0.0;0.1;0.3;0.35;0.8;0.9;0.4;0.9;0.57;0.45;0.63;0.89;0.99;1.0;0.65;0.84;0.54;0.8;0.83;0.93;0.94;0.73;0.84;0.82;0.5

assange;0.1;0.0;0.3;0.32;0.85;0.95;0.34;1.0;0.59;0.42;0.6;1.0;1.0;1.0;0.61;0.82;0.5;0.85;0.8;0.83;0.92;0.79;0.7;0.89;0.5

...

**Étape 3 – Partition des mots**

Toute méthode de classification non supervisée (voir cours 3) peut-être utilisée ici, soit directement à partir de la matrice de distance, soit en l'utilisant comme un ensemble de vecteurs décrivant chacun des 25 mots.
Le résultat attendu peut être comparé au résultat prévu (fourni dans l'énoncé du projet) grâce aux distances classiques entre partitions, par exemple Rand et Rand corrigé (http://en.wikipedia.org/wiki/Rand_index#Adjusted_Rand_index).

Livrable :

- la partition devra être stockée dans un fichier texte, où chaque mot apparaîtra sur une ligne, les mots d'une même classe apparaissant de manière consécutive, et chaque classe étant séparée par un saut de ligne.
- le script Python, nommé `creeClasses.py`, devra prendre un seul argument, l'adresse du fichier CSV de type *[adresse].txt.csv*, et écrire le fichier texte décrit ci-dessus à l'adresse *[adresse].txt.partition.txt*.

## Échéances

- 18 mars : constitution des équipes sur http://tinyurl.com/ProjetInfoling2012.
- 25 mars : envoi du livrable de l'étape 1 (texte d'au moins 300 mots qui contient au moins une fois chacun des 25 mots-clés).
- 30 mars : réception du corpus de test (pour tester la formule de distance et la méthode de partition, et les optimiser).
- 29 avril : envoi des livrables des étapes 2 et 3 (fichier CSV de la matrice de distance entre les 25 mots, fichier TXT contenant la partition calculée, scripts Python `creeMatrice.py` et `creeClasses.py`, rapport court de 2 pages maximum).
- mai : évaluation des projets sur le corpus d'évaluation, délivrance d'un certificat de participation au projet.

## Évaluation

Chacune des trois étapes du projet sera évaluée :

- de manière qualitative, en prenant en compte le respect des contraintes, la qualité, la fonctionnalité, et la réutilisabilité du code, la créativité des travaux réalisés.
- de manière quantitative, en calculant un score de distance entre le fichier envoyé et ce qui était attendu.

## Contenu du rapport

Le rapport sur le projet, d'au plus deux pages, contiendra trois parties :

- méthodes : courte explication du choix des méthodes et de leur paramétrage, avec éventuellement des renvois vers les sources d'information utilisées (articles scientifiques, pages web).
- résultats obtenus et perspectives : commentaires sur les résultats obtenus, et les perspectives à court et moyen terme si vous deviez améliorer le projet.
- interactions : description structurée des activités de chaque membre de l'équipe. Même s'il n'est pas complètement exclu que les membres d'une même équipe obtiennent des notes différentes, la capacité à valoriser les interactions et l'esprit d'équipe, ainsi que les outils de travail collaboratif, sera particulièrement appréciée dans cette partie. Une prévision équilibrée des activités de chaque membre de l'équipe, dès le début du projet (en évitant d'attribuer la responsabilité de toute une étape à un seul membre de l'équipe), permettra d'assurer la qualité de cette partie, et du résultat final.