Programmation Orientée Objet
Programmation JAVA

FISE 3A ICy / FISA 3A Informatique

TP 1 partie 2 : Les chaines de caractères

Mohamed Amine BOUDIA
UPHF, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France
Email : mohamedamine.boudia@uphf.fr

- Une des tâches les plus importantes de l'Intelligence Artificielle, est la fouille de données textuelles, dit Text Mining

- Nous appliquons des algorithmes d'apprentissage pour des fins bien définies

- Mais, les algorithmes d'apprentissage sont conçus pour être appliqué sur des données structurés et vectorisés et généralement sur des valeurs numériques comme suite :

	Année	Puissance	Prix	Classe
Instance 1	2022	130	15000	1
Instance 2	2003	45	17000	0
Instance 3	2017	80	9000	1
Instance 4	1999	60	8500	0

- Afin d'appliquer les algorithmes d'apprentissage sur les données textuelles, on doit convertir les textes à ce format (instance, attribut) avec des valeurs numériques, en gardant le sens, ce qu'on appelle : **Vectorisation**.

	Mot 1	Mot 2	Mot 3	Classe
Texte 1	0	2	4	1
Texte 2	1	3	2	0
Texte 3	7	6	7	1
Texte 4	4	2	1	0

- Afin d'appliquer les algorithmes d'apprentissage sur les données textuelles, on doit convertir les textes à ce format (instance, attribut) avec des valeur numérique, en gardant le sens, ce qu'on appelle : Vectorisation.

	Mot 1	Mot 2	Mot 3	Classe
Texte 1	0	2	4	1
Texte 2	1	3	2	0
Texte 3	7	6	7	1
Texte 4	4	2	1	0

- Pour cela, plusieurs étapes sont requises, nous vous demandons d'écrire les programmes java dédiés pour la vectorisation.

- On suppose que une collection des textes à traiter appelé CORPUS est déjà existante dans votre programme, quelle est la structure de données que vous utilisez pour les traiter?

  Réponse : String corpus = new String[1000] ;

- Etape 1 : Nettoyage

  Consiste à supprimer les caractères spéciaux, les mots vides ou / et les nombres selon le problème a résoudre

  Ecrire un programme qui demande à l'utilisateur de choisir un ou plusieurs modes de nettoyage et puis fait le nettoyage sur les textes. Si l'utilisateur choisi le mode de nettoyage par mots vides, il doit entrer les mots vides et finir la liste par un « 0 ».

- Etape 2 : Passer du texte brut à Texte/Attribut, choix d'attribut

  Il y a plusieurs façon pour passer du texte à Texte/Attribut, cela est un le choix des attributs, ce choix doit garder la sémantique et avoir une relation avec les textes, on l'appelle : tokinisation .

  Nous allons programmer dans ce TD deux méthodes phares : tokinisation par mot et tokinisation par N-Gram

  Ecrire un programme qui demande à l'utilisateur de choisir la méthode de tokinisation puis fait la tokinisation les textes de corpus

- Etape 3 : Codification

  Après avoir choisi les attributs qui représentent les textes sans perdre de la sémantique, nous devons choisir maintenant la façon de calculer les valeurs d'attributs (valeur texte/attribut).

  - La valeur peut être booléen (l'attribut existe ou pas dans le texte)
  - La valeur peut être la récurrence (nombre d'apparition d'attribut dans le texte) appelé tf.
  - ou par rapport à son importance au corpus appelé TF-IDF:

    $$tf\text{-}idf(t,i) = tf(t,i) \times \log(N / \text{nombre de texte ayant l'attribut } i)$$

    Où N est le nombre de texte.

  Ecrire un programme qui demande à l'utilisateur de choisir la méthode de codification puis fait la codification les textes corpus.