Weka + fouille de textes

1 Objectif

Au cours de cette séance de TP, nous allons voir comment utiliser Weka pour faire du traitement de données textuelles à partir de documents bruts (i.e. qui ne sont pas encore sous la forme de vecteurs).

Cette séance est particulièrement importante en ce sens qu'elle sera à la base du projet qui comptera pour l'évaluation de cette unité d'enseignement.

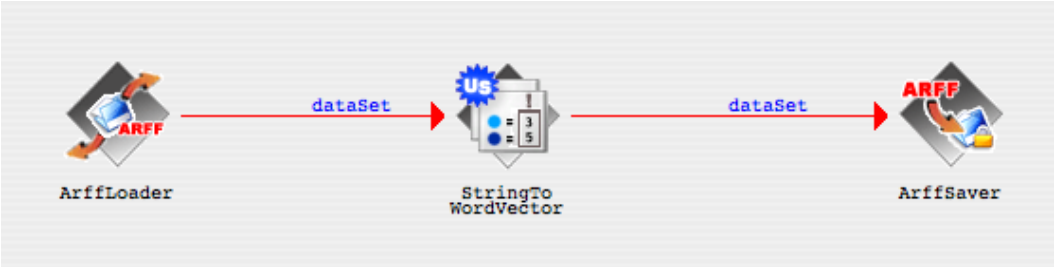
2 Transformation TF-IDF

2.1 Utilisation du KnowledgeFlowManager

Nous décrivons comment utiliser un des outils de Weka, le KnowledgeFlowManager (qui se trouve dans le menu *Application* de Weka), pour créer des représentations TF-IDF d'une collection de textes.

Nous allons travailler avec la base de textes anglais mini_newsgroups.tar.gz, composée de 2000 articles de newsgroups, qui se divisent en 20 thèmes différents (100 articles par thème). Afin de ne pas surcharger Weka, qui ne supporte pas très bien le traitement de collections de textes volumineuses, nous ne considérerons que les thèmes *sci.electronics*, *sci.med*, *sci.space*, soit 300 articles. Chacun des thèmes correspond à un répertoire et tous ces répertoires sont des sous-répertoires qui seront regroupés dans un dossier *mini_newsgroups*.

Pour créer les vecteurs TF-IDF correspondant à ces textes, il suffit de créer le diagramme suivant dans le KnowledgeFlowManager :



La boîte de gauche permet de spécifier le répertoire contenant les sous-répertoires associés aux thèmes étudiés (qui, dans le cas de la classification supervisée permettront de définir l'étiquette des articles). Lorsque le traitement des données sera lancé, les vecteurs TF-IDF de l'ensemble des fichiers se trouvant dans les sous-répertoires de *mini_newsgroups* seront calculés.

Celle du milieu définit les paramètres TD-IDF à utiliser, en permettant notamment de préciser le stemmer (algorithme de lemmatisation) choisi et si le recours à une stop-list doit avoir être lieu.

Enfin, la dernière boîte porte l'information sur le ficher .arff (format Weka) qui contiendra l'ensemble de tous les vecteurs TF-IDF. S'il l'a été précisé dans la boîte du milieu, il est possible de faire en sorte que l'attribut de thème apparaisse comme le premier attribut de chacun des vecteurs TF-IDF. Cela peut être utile pour faire de la classification.

Une fois ce schéma mis en place, il suffit de charger les données dans la première boîte (clic droit sur la boîte) pour que la chaîne de traitement des articles se trouvant dans les répertoires sus-nommés soit executée et pour que les vecteurs TF-IDF soient produits. Il est alors très facilement possible d'utiliser les outils de Weka, ou bien tout autre outil, pour procéder à une analyse des textes (par exemple, k-moyennes, classification supervisée – petit bémol concernant ce point, cf. ci-dessous –, analyse en composantes principales, visualisation des textes).

2.2 Utiliser les filtres à la main

Si l'objectif de l'étude est de faire de la classification supervisée, en utilisant par exemple le classifieur Naive Bayes ou bien la régression logistique, il n'est pas possible d'utiliser le KnowledgeFlowManager pour la création des données TF-IDF. En effet, si le type de tâche visée est la classification, alors il est impératif d'avoir à disposition un moyen de valider la performance du classifieur obtenu. Pour cela la méthode la plus communément implémentée consiste à utiliser des données de test indépendantes des données d'apprentissage ; or, le KnowledgeFlowManager ne permet pas de produire, à partir d'une collection de documenst, deux collections de vecteurs TF-IDF indépendantes.

Afin de générer des ensembles de vecteurs TF-IDF d'apprentissage et de test indépendants, il est nécessaire d'utiliser un appel "manuel" au filtre TF-IDF. Ainsi, pour produire un fichier d'apprentissage et un fichier test, il est nécessaire de disposer de deux répertoires *mini_newsgroups_train* et *mini_newsgroups_test*, chacun d'eux contenant des sous-répertoires associés au thèmes d'intérêt. Pour produire les fichiers .arff d'apprentissage et de test, il suffit de faire les trois commandes suivantes (qui peuvent être accompagnées de plusieurs options que nous ne décrivons pas ici mais que nous pourrons voir en TP) :

```
java weka.core.converters.TextDirectoryLoader -dir mini_newsgroups_train
    -F 1 > train.arff
java weka.core.converters.TextDirectoryLoader -dir mini_newsgroups_test
    -F 1 > test.arff
java weka.filters.unsupervised.attribute.StringToWordVector -b -i train.arff
    -o train.out.arff -r test.arff -s test.out.arff
```

Le fichier *train.out.arff* et le fichier *test.out.arff* contiennent alors respectivement des données d'apprentissage et de test qui sont bien indépendantes.

3 Sujet et déroulement du projet

Le sujet du projet est l'utilisation des techniques de fouille de données textuelles vues en cours (+ éventuellement d'autres techniques comme l'analyse canonique des corrélations) pour l'analyse du positionnement des grands périodiques nationaux par rapport à la campagne présidentielle et plus précisément par rapport aux candidats. Les séances de TP seront à présent entièrement consacrées au projet, pour lesquels il est laissé toute libertés concernant les méthodes utilisées et les aspects étudiés. Une très bonne base de textes, chacun étant associé à un (ou plusieurs) candidat (ou ex-candidat), peut être récupérée de la page de Jean Véronis à l'adresse suivante : .