Département MIDO
Module DM/ML
2015 / 2015-6

Travaux Pratiques n° 3 : Text Mining

Objectifs : s'initier à la fouille de texte et à l'Analyse Sémantique Latente.

Nous allons considérer dans ce TP l'analyse d'un corpus de pages wiki, extrait à partir de `http://edutechwiki.unige.ch`. Nous suivrons dans les grandes lignes la démarche expliquée dans `http://edutechwiki.unige.ch/fr/Tutoriel_tm_text_mining_package`.

1. Commencer par décompresser l'archive accompagnant le sujet dans votre espace de travail local (e.g. `~/DataMining/TP1/`)
2. Changer d'espace de travail : `setwd("~/DataMining/TP1/")`
3. Commencer par charger les librairies :

```
library(tm)
library(tm.plugin.webmining)
library(SnowballC)
```

4. Importer les données dans une variable "corpus", e.g. :

```
corpus <-Corpus(DirSource("./",encoding="UTF8"), readerControl = list(language="lat"
```

5. Nettoyer le corpus pour pouvoir le traiter :

   — Mettre en minuscule
   `wiki.cl1 <- tm_map(corpus, content_transformer(tolower))`
   — Tuer les balises
   `wiki.cl2 <- tm_map (wiki.cl1, content_transformer(extractHTMLStrip), encoding="UT`
   — Autres nettoyage

```
(kill_chars
         <- content_transformer (function(x, pattern) gsub(pattern, " ", x)))

tm_map (wiki.cl2, kill_chars, "\u2019")
tm_map (wiki.cl2, kill_chars,"’")
tm_map (wiki.cl2, kill_chars,"[«»’“‘\"]")
tm_map (wiki.cl2, kill_chars,"\\[modifier\\]")
```

   — Enlever les ponctuations qui restent :

```
wiki.cl3
 <- tm_map (wiki.cl2, removePunctuation, preserve_intra_word_dashes = TRUE)
```

— Enlever les mots fréquents :
```
wiki.essence <- tm_map (wiki.cl3, removeWords, stopwords("french"))
```
— Extraire les racines
```
wiki.racines <- tm_map (wiki.essence, stemDocument, language="french")
```
— Enlever les blancs s'il en reste
```
wiki.racines <- tm_map (wiki.racines, stripWhitespace)
```
— Test
```
wiki.racines[[2]]
class(wiki.racines)
```

6. Créer la matrice documents-termes

```
wiki.mots <- Corpus(VectorSource(wiki.racines))
matrice_termes_docs <- DocumentTermMatrix(wiki.mots)
```

7. Réduction de la matrice

```
inspect(removeSparseTerms(matrice_termes_docs, 0.4))
inspect(removeSparseTerms(matrice_termes_docs, 0.6))
```

8. Visualisation de la matrice termes-documents

```
# Créer une DTM avec des poids normalisées
mtd.norm <- as.matrix(removeSparseTerms(
  TermDocumentMatrix(wiki.mots, control=list(weighting=weightTf)),
  0.2))
corrplot (mtd.norm, is.corr=FALSE)
```

9. Création de la matrice TFidF

```
mtd.TfIdf2 <- as.matrix(removeSparseTerms(
  TermDocumentMatrix(wiki.mots, control=list(weighting=weightTfIdf)),
  0.2))
# Plot simple
corrplot (mtd.TfIdf2, is.corr=FALSE)
```

10. Appliquer une Décomposition en Valeurs Singulières à la matrice `mtd.TfIdf2`. Expliquez ? (commande `svd`)
11. Appliquer une ACP à cette matrice de données. Interprétez !
12. Appliquer la méthode k-moyennes sur la matrice `mtd.TfIdf2` avant et après application de la Décomposition en Valeurs Singulières. Commentez !
13. Appliquer la méthode k-moyennes sur la matrice `mtd.TfIdf2` avant et après application de l'ACP. Commentez !
14. Comparer les résultats des deux approches.
15. Appliquer la méthode EM sur la matrice `mtd.TfIdf2` avant et après application de la Décomposition en Valeurs Singulières. Commentez !
16. Appliquer la méthode EM sur la matrice `mtd.TfIdf2` avant et après application de l'ACP. Commentez !
17. Comparer les résultats des deux approches.
18. EM vs K-means : Discuter les résultats obtenus par ces deux approches dans toutes les configurations considérées ci-dessus.