TP3 Apprentissage statistique

El Haouari Naïla - Fontier Zoé

09 Décembre 2019

Nous avons choisi de travailler sur les scripts du "Seigneur des Anneaux", base récupérée sur le site Kaggle[1]. Cette base comprend les scripts des trois films en anglais.

1 Analyse des données

1.1 Processing

Dans un premier temps, on nettoie les données : on transforme toutes les lettres en minuscule (*lowercase*), puis on enlève les *stopwords* de la base, ainsi que d'autres mots – qui n'étaient pas inclus dans la fonction *stopwords*, mais dont on n'a pas besoin dans l'analyse, tels que *ll*, *ve*, qui sont issus des formes contractées de *will* et *have*. Cela nous permet de ne conserver que les mots pertinents pour l'analyse. On enlève ensuite les signes de ponctuation.

On applique ensuite la fonction **word2vec** à notre fichier .txt, c'est-à-dire qu'on associe à chacun des mots des scripts un vecteur de taille 200.

1.2 Similarité entre les mots

Nous procédons tout d'abord à une analyse textuelle, en reprenant le code du TP3. On observe la similarité avec certains mots : on choisit d'analyser les mots proches des différents noms de lieux et personnages.

Ci dessous sont présentés les tableaux de similarités. Les termes associés à Gandalf (table 1) sont des termes liés à son apparence ("grey", "white") ou à des évènements qui lui sont arrivés ("fall", "montain"). Nous remarquerons aussi que les termes associés à "gollum" et "smeagol" (tables 2 et 3) sont différents même s'ils réfèrent à une même personne. Le personnage initial de smeagol est moins associé au terme "precious" que son autre moi "gollum". Ce dernier est par ailleurs associé au terme "creature" comme il se surnomme parfois dans les films.

L'analyse textuelle nous montre que les différents personnages ont des mots associés bien différents et qui correspondent à leur trajectoire et leurs caractéristiques personnelles dans le film. Il semble y avoir une cohérence dans les personnages et leur environnement. Quand on regarde les termes qui caractérisent les royaumes qui s'affrontent (tables 4 et

[1] https://www.kaggle.com/paultimothymooney/lord-of-the-rings-data

Table 1: Word dimilarity to "gandalf"

gandalf	1
white	0,7779368
grey	0,7622961
tree	0,7081228
horses	0,6272097
fall	0,6193605
west	0,6072328
mountain	0,5846890
turn	0,5803779
age	0,5785128

Table 2: Word similarity to "gollum"

gollum	1
soon	0,8981906
called	0,8593391
precious	0,8372781
creature	0,8250619
brought	0,8248810
mountains	0,8159625
love	0,7618268
happen	0,7460491
came	0,7454552

Table 3: Word similarity to "smeagol"

smeagol	1
hurt	0,8856290
ask	0,8103682
places	0,8014548
wants	0,7834593
decent	0,78033228
course	0,7716626
precious	0,7506857
got	0,7462905
fish	0,7307805

Table 4: Word similarity to "gondor"

gondor	1
heir	0,8141964
rest	0,8066937
fell	0,7918128
king	0,7862670
faramir	0,7784294
hail	0,7636621
rule	0,7619454
isildur	0,7468368
swore	0,7442981

Table 5: Word similarity to "mordor"

mordor	1
send	0,7652489
orcs	0,7425383
welcome	0,7378827
air	0,7258227
courage	0,7240098
save	0,7128411
heard	0,7095453
thousand	0,7023486
many	0,7012640

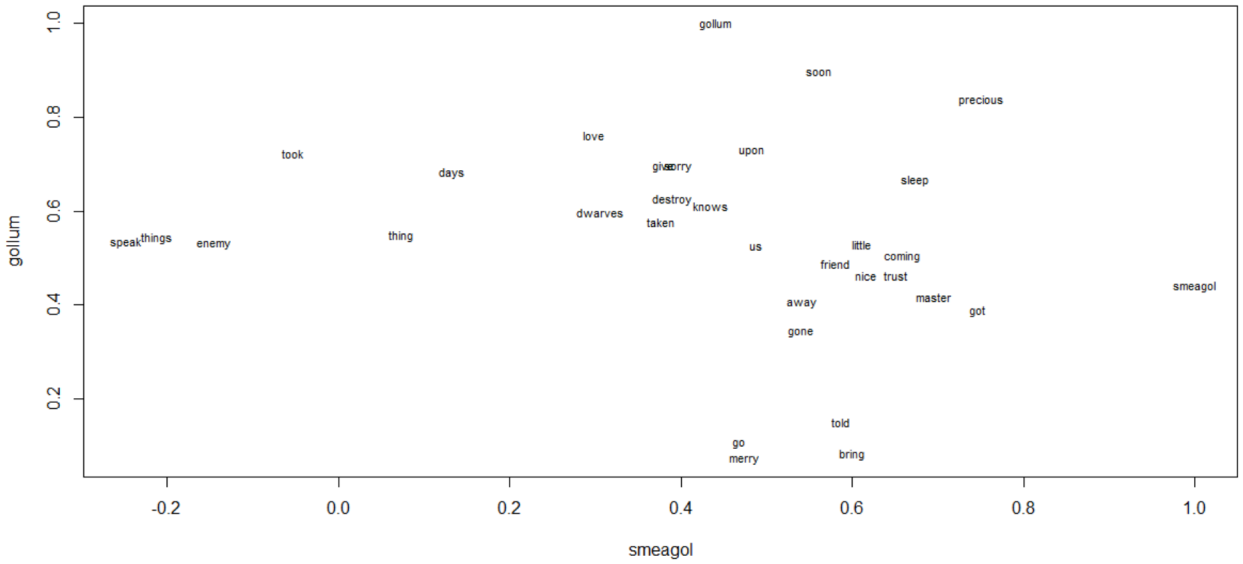

Figure 1: Similarité des mots avec Smeagol et Gollum

5), on remarque un lexique autour de légitimité, de la descendance, de la royauté associé au Gondor alors que le Mordor est associé aux orcs, créatures affreuses qui pullulent ("many", "thousand"). Ces deux mondes sont donc bien caractérisées par un univers lexical différent.

On peut également représenter visuellement les termes similaires à certains mots. On montre dans la figure 1 les termes similaires à *gollum* et à *smeagol*. Rappelons qu'il s'agit du même personnage, mais que Smeagol est son véritable nom lorsqu'il était encore sain d'esprit. On constate que *precious* est davantage lié à Gollum qu'à Smeagol, ainsi que *enemy*, *sorry*, et *taken* (en référence au fait qu'on lui a pris son anneau); tandis que des termes plus positifs, tels que *nice*, *friend*, *trust* sont associés à Smeagol – ainsi que *master*, qui est la façon dont Smeagol appelle Frodo.

1.3 Réduction de dimension

Dans un second temps, on recourt à la méthode TSNE pour réduire la dimension des vecteurs. On se focalise sur les 100 mots les plus fréquents, pour des raisons de lisibilité des résultats. La librairie **tsne** permet d'afficher, avec la fonction **plot**, un nuage de mots, regroupés selon leur similarité. On choisit de fixer le paramètre *perplexity* – qui correspond au nombre de voisins – sur 20, car cela permet de faire apparaître plus visiblement des clusters.

On représente les résultats sur la figure 2. S'il est assez difficile de lire tous les mots, on peut faire quelques observations. Dans la partie centre-bas, certains mots du monde des hobbits sont proches (*hobbits*, *shire* – le lieu de vie des hobbits – , *bilbo* et *baggins*). Un peu au-dessus, on trouve des termes similaires à la guerre et au combat (*ride*, *battle*, *war*). Tout à gauche, on observe qu'un cluster est formé par les termes *hope* et *still* – ce qui doit être dû au fait que les personnages espèrent qu'il reste encore de l'espoir. Plus à droite du nuage, on retrouve des termes davantage liés à Frodon, Sam, et leur voyage avec Gollum.

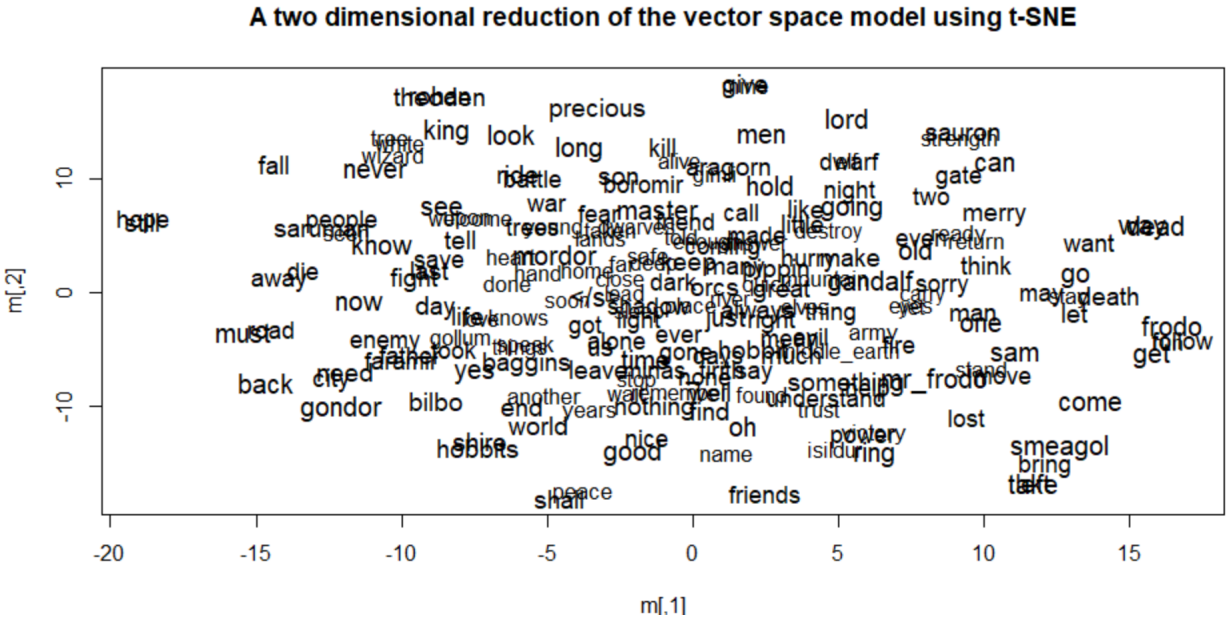

Figure 2: Réduction de dimension – Similarité entre les 100 mots les plus fréquents

Bien que ce graphe permette de visualiser en deux dimensions les similarités entre les mots, on remarque bien qu'il est difficile à interpréter, et que beaucoup de mots se chevauchent. On applique alors d'autres méthodes d'analyse textuelle.

1.4 Clustering

Dans un deuxième temps, on propose d'effectuer un clustering des mots, pour mieux visualiser la similarité entre les mots. On se focalise sur les 200 mots les plus fréquents, et on opte pour 6 clusters. On effectue alors un clustering par kmeans, avec le code inclus dans la figure 3. On obtient les clusters suivants :

1. Mots associé au combat et à la mort : *dead, death, long, day, ride, fear, army, die, ready, battle, lands, etc.*

2. Mots associés aux hobbits et à la thématique du voyage : *frodo, way, hobbits, bilbo, shire, pippin, road, baggins, follow, mountain, home, etc.*

3. Mots associés à Gollum : *go, precious, us* (car Gollum a deux personnalités et dit "us" au lieu de "I"), *master, smeagol, gollum, trust, taken, etc.*

4. Lié aux rois des hommes et à Gandalf : *gandalf, white, wizard, return* (pour Gandalf) ;*king, lord, rohan, city, man, theoden* (pour les hommes).

5. Vocabulaire d'action et de précipitation : *back, get, them, take, going, see, leave, kill, hurry, help, move, etc.*.

```
##########      CLUSTERING      ##########
set.seed(10)
centers = 6
clustering = kmeans(model[1:200,],centers=centers,iter.max = 40)
for (i in 1:6){
  print(paste("cluster", i))
  print(names(clustering$cluster[clustering$cluster==i]))
}
```

Figure 3: Code pour le clustering

6. Vocabulaire lié à la guerre, mais en particulier entre le Gondor et le Mordor : *men, ring, gondor, mordor, sauron, isildur, last, many, fight, days, father, enemy, evil, middle-earth, etc.*

En dehors des termes listés ci-dessus, qui permettent de définir des catégories, il convient de mentionner que chaque cluster comprend également un certain nombre de termes assez génériques, et dont on ne sait pas pourquoi l'algorithme les a assignés à ce cluster en particulier. On peut supposer que certains termes des scripts du Seigneur des Anneaux apparaissent fréquemment dans divers contextes et sont pour cela difficilement catégorisables.

2 Apprentissage supervisé : retrouver les clusters prédits

Dans cette seconde partie, on cherche à appliquer des algorithmes d'apprentissage supervisé, comme ceux vus dans les deux premiers TP, à notre jeu de données. Puisqu'on ne dispose pas vraiment de labels à prédire – comme on n'a que des mots dans nos données –, on va chercher à prédire les classes prédites dans le clustering effectué dans la partie 1.4, et comparer les différents algorithmes.
On va donc se focaliser sur les 200 mots les plus fréquents. On transforme les clusters et le modèle en dataframe pour pouvoir appliquer les algorithmes. On a donc 200 features associés à chaque mots (puisqu'on avait associé à chaque mot un vecteur de taille 200). On divise ensuite notre jeu de données en un jeu d'entraînement (150 observations, soit 75% des données) et un jeu de test (50 observations).

2.1 Algorithmes d'apprentissage utilisés

On test différents algorithmes d'apprentissage supervisé.

Dans un premier temps, on effectue un algorithme des k-plus proches voisins. On détermine le nombre k de plus proches voisins par cross-validation, en divisant notre échantillon d'entraînement en 5 sous-échantillons, et en prédisant le numéro de cluster pour chacun des mots par knn, pour k variant de 1 à 10. On effectue 100 itérations, et on choisit le nombre k qui minimise le taux d'erreur moyen. Il s'agit de k=8 – et on testera également pour k=9, car les taux d'erreurs sont proches.
On essayera également de prédire le cluster par arbre de classification, et avec l'algorithme *randomforest*.

2.2 Comparaison des prédictions

Pour l'approche par KNN, on obtient une *accuracy* de 84% pour l'algorithme des plus proches voisins, avec k=8 et k=9. L'approche par arbre de décision est bien moins bonne, puisqu'on obtient une précision de 58%. Enfin, c'est avec l'algorithme de *Randomforest* qu'on obtient la meilleur *accuracy* puisqu'elle est de 90% – notons que l'on a utilisé les paramètres par défaut, et que cette précision peut probablement être optimisée en modifiant les paramètres de l'algorithme.

On cherche ensuite à afficher les mots qui ont été mal prédits. Pour les KNN, il s'agit de : *"days", "end", "gandalf", "get", "just", "many", "mordor", "told"*, et pour l'algorithme de *Randomforest*, uniquement *"days", "end", "many", "mordor", "told"*. On retrouve également ces mots, ainsi que d'autres, dans la prédiction par arbre de décision. On peut imaginer que ces mots pourraient avoir leur place dans divers clusters – car certains appartiennent à un vocabulaire assez général.