Projet de Text Mining

Nous sommes des passionnés de musique et nous voulions faire tourner le projet autour de ce thème. Nous avons donc décidé d'étudier des paroles de chanson. Initialement, le but de notre projet était de prédire l'artiste qui a interprété la chanson à partir des paroles. Bien évidemment, il est impossible d'avoir une liste exhaustive d'artistes, nous avons donc choisi de prédire de quel artiste les paroles d'une chanson se rapprochaient le plus parmi 20 des artistes les plus populaires en RnB.
N'ayant pas de corpus à disposition, nous avons dû en constituer un. Pour ce faire, nous avons utilisé l'api de genius afin de récupérer les paroles des chansons des artistes qui nous intéressaient. Le nombre de chansons sélectionnées par artiste est au maximum 200 mais certains ou certaines n'ont pas chanté assez de chansons (l'artiste avec le moins de chansons est Aaliyah : elle en a chanté 116). Il y a donc globalement à peu près le même nombre de musiques pour chaque artiste et elles sont toutes en anglais.

Nous avons utilisé une méthode de classification supervisée pour répondre à la problématique énoncée ci-dessus. Afin d'utiliser les algorithmes déjà implémentés, nous avons dû transformer les données textuelles en données numériques. Pour cela, nous avons suivi les étapes de preprocess suivantes :

1. Nettoyage du corpus

Nous avons dans un premier temps supprimé la ponctuation (qui ne contient aucune information)
Nous avons ensuite séparé les mots les uns des autres (tokenisation)
Puis nous avons essayé de réduire au maximum le bruit en supprimant les mots vides
Enfin, nous avons homogénéisé notre corpus en regroupant les différentes formes de mots (lemmatisation ou stemming)

2. Passage des données textuelles aux données numériques (Tf-Idf puis prise en compte des n-gram)

Pour que les modèles de machine learning classique puissent fonctionner, ils ont besoin de données numériques. Nous avons donc converti notre corpus en données numériques grâce au tf-idf lissé.
Afin de conserver les liens entre les mots, nous avons décidé de conserver les unigram et les bigram.

3. Apprentissage d'un modèle bayésien naïf

Les données ont été divisées en 2 : une partie (66% des données) a servi à l'apprentissage du modèle et une autre (33%) à l'estimation de sa performance.
Nous avons effectué une validation croisée pour optimiser les paramètres de notre modèle à partir des données d'apprentissage.

4. Probabilité d'appartenance pour chaque artiste sur des nouvelles chansons

Une fois notre modèle entraîné, nous avons pu tester les performances sur les données de test. Pour 20 classes à prédire, notre modèle a une exactitude de 0.385.
Nous avons ensuite utilisé notre modèle sur des chansons interprétées par des artistes ne faisant pas partie du corpus. Nous avons ainsi pu observer que "Hello" de Adèle se rapproche plus de Christina Aguilera que de Rihanna par exemple.

5. Réduction du nombre de classe

Dans l'optique d'obtenir des résultats plus significatifs, nous avons réduit le nombre de classes à prédire à 10. Les résultats obtenus sont alors de 0.457.

Ce projet a été une source inépuisable de joie et d'amusement qui nous a grandi. Notre modèle n'est pas très performant (que ce soit pour 20 ou 10 classes) ce qui n'est pas surprenant : il y a beaucoup de classes

à prédire et les textes sont des données difficiles à utiliser en machine learning. De plus, la taille de notre corpus n'est sans doute pas assez importante (peu d'artistes ont chanté plus de 200 chansons dont les paroles sont répertoriées). Rien n'a été particulièrement difficile mis à part certains problèmes "techniques" de programmation (qui sont courants dans ce genre de projet). Les pistes que nous pourrions approfondir seraient peut-être de remplacer les entités nommées dans les textes par leur attribut (exemple : remplacer "New York" par "LIEU" ou "Stephen Hawking" par "PERSONNE"), ou bien de mieux sélectionner les artistes à prédire, selon leur niveau de langue ou les champs lexicaux utilisés, de façon à ce qu'ils soient le plus discriminant possibles entre eux. On peut aussi imaginer utiliser des modèles de deep learning déjà pré-entrainés (comme BERT par exemple) et adapter ces modèles à notre corpus en ré-entrainant les couches de prédiction de façon à ce qu'elles s'adaptent à nos données.