Carlos Andres CORTES MIRANDA
DATA732
Janvier 2024
Rapport

Sémantique Clustering of articles

Table des matières

1. Introduction

L'objectif principal de ce projet, c'était d'utiliser différents outils de classification pour pouvoir réaliser une étude simple sur le jeu des données des articles. Dans l'approche que j'ai choisi ce la classification sémantique des différents articles, avec l'objectif de regarder le comportement des différentes pages web et voir si on peut trouver des tendances dans la rédaction des différents types des articles.

L'idée initial était de trouver les différents groupes sémantiques qu'il existe dans le jeu de données des articles et selon leur date de publication voir quels sont topics les plus parles selon les différentes périodes des dates.

2. Choix des technologies

Pour réaliser cette étude, j'ai décidé de travailler sur des fichiers .notebook, du à leur flexibilité des au moment des lancer des taches des traitement lourdes. Comme la quantité des articles et les méthodes que j'ai utilisé ont besoin des grandes capacités des traitement, les fichier notebook permet de modifier information déjà traite plusieurs fois sans lancer tout le code. Cela permet d'essayer différentes méthodes et corriger les fautes plus facilement, surtout pour quelqu'un comme moi qui en est encore à ses premiers pas dans les applications ml.

Toutes les méthodes de clustering, réduction dimensionnel et Machine Learning utilise lors de ces études sont de la librairie python « Scikit learn ». Le choix de cette librairie est dû aux larges quantités des exemples, documentation, information utile et questions et réponses que ont peux trouver sur ces fonctions.

Pour réaliser toutes les réductions dimensionnelles, j'ai utilisé le méthode « T-distributed Stochastic Neighbor Embedding », la fonction de Sci-kit learn. J'ai décide d'utiliser T-SNE parce qu'il s'agit d'une technique non linéaire de réduction de la dimensionnalité. Dans notre cas les articles ne suivent aucune loi mathématique. Donc une approche qui est moins sensible à l'ordre des points de données se rend très utile dans notre cas.

Pour la génération des graphs j'ai opté pour la librairie « Plotly » dans python et la plateforme « Tableau ». « Plotly » parce qu'elle est bien intégrée avec l'utilisation des dataframes pandas. Tableau parce qu'on a appris à l'utiliser cette année dans le module et j'ai trouve convenaient de l'utiliser pour générer les graphs avec les résultats finales.

3. Méthodes choisis pour la classification

Pour le choix des méthodes de classification sémantique, j'ai applique deux types : K-means et Latent Dirichlet Allocation(LDA). K-means est une des méthodes les plus courants pour faires des classifications mais pas le plus approprié pour des classifications sémantiques du texte. K-means reste plus simple et plus rapide que LDA, mais selon plusieurs études LDA est efficace pour trouver les sujets caches dans la donnée.

J'ai utilisé deux méthodes de vectorisation de la librairie « SciKit Learn » : CountVectorization et TfidfVectorizer. CountVectorization est une méthode plus simple qui génère une matrice de comptage des occurrences de mots. Cependant TfidfVectorizer attribue des scores aux mots en fonction de leur importance relative dans le corpus. TF-IDF (Term Frequency-Inverse Document Frequency) prend en

compte à la fois la fréquence d'un terme dans un document particulier (Term Frequency) et son importance globale à travers tous les documents (Inverse Document Frequency).

4. Problèmes rencontres

Le principal problème rencontre est lie à la capacite de calcul dès mon pc et la manque d'accès à un centre de traitement puissante. Avec une cpu i5 (7eme gen) certains notebooks avec les 36000 articles ont pris plus des 4+ heures a tournes pour avoir des résultats. La tour qu'on a accès à Polytech est un possible solution mais l'incapacité d'ajouter des librairies limite l'utilisation. Pour cela dans certains cas les résultats affiches sur les notebooks sont avec 935 articles. Les études, la paramétrisation des vectorizateurs et classificateurs est différent selon la quantité des articles.

Autre problème que j'ai rencontré est le manque de préparation dans le sujet. L'application des modelés comme Kmeans ou LDA est possible jusqu'à certain dégrée. J'ai eu beaucoup des problèmes à la paramétrisation des modèles surtout en travaillent avec la totalité des articles, parce l'utilisation des toutes les « features » génère des résultats anormaux, donc il fallait ajouter les conditions sur la vectorisation des mots. Avec le 36000 articles la quantité des « features » dépasse les millions, donc il fallait enlever les mots qui apparaissent le moins. Cela c'est un possible solution mais pas la meilleure, dû à quelle génère la perte d'information.

5. Pistes d'amélioration

Pendant le développement de ce projet, j'avais l'idée d'utiliser un modelé de « Zero-shot classification ». J'ai récupère le modèle de « Hugging Face » : MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli · Hugging Face. Ce modelé permet de donner en input un texte et un groupe des labels. En return, il t'envoie les pourcentages correspondants aux classifications du texte selon les labels. En appliquant faisant une classification de toutes les articles publiés dans un pages web, on aura trouvé quels sont les tendances de chaque page web. Le seul problème c'est qu'effectuer la classification d'un seul texte prend au tour de 5-10 seconds. Réaliser la classification avec 1000 articles aura pris un temp non négligeable. Trouver autres moyennes de réaliser cet étude c'est comme réaliser un étude de sentiment plus complet.