Master 1 Informatique

Rapport de Projet de Traitement automatique du texte en IA

Sujet :

Définir le(s) genre(s) d'un livre en fonction de son résumé.

Kilian CLARK
Sébastien MARRO

Professeure : Cabrio Elena
Année universitaire 2020 - 2021

Table des matières

Chapitre 1

Introduction et présentation du sujet

1.1 Introduction

1.2 Présentation du sujet

1.2.1 Choix du sujet

En ce qui concerne notre choix de sujet, celui-ci a été plutôt laborieux. En effet, nous voulions tout d'abord choisir un sujet concernant les musiques et paroles de chansons. Le principe aurait été de réaliser une sorte de contrôle parental sur les musiques en fonction des paroles utilisées. Nous aurions réalisé un programme capable de rendre un pourcentage de grossièreté en fonction des mots ou expressions utilisés. Une fois ce pourcentage obtenu nous aurions pu avoir un pourcentage moyen par artiste. Tout ceci nous aurait permis de pouvoir sélectionner une playlist de chansons en enlevant les chansons ou artistes ayant un pourcentage supérieur à 60% de grossièreté pour donner un exemple. Nous n'avons pas pu réaliser ce projet à cause du manque de ressources disponibles dû aux « copyrights » des chansons. Nous avons donc dû changer de sujet de projet pour un autre qui nous intéressait également.

Nous allons maintenant vous introduire ce sujet que nous avons choisi afin de réaliser notre projet.

1.2.2 Le sujet choisi

Notre sujet de projet porte sur les livres et leur résumé. En outre, l'objectif de ce projet est de pouvoir définir le ou les genre(s) d'un livre en fonction de son résumé. Ceci nous permettrait de pouvoir classer des livres plus facilement ou bien simplement chercher des livres suivant leur(s) catégorie(s) que ce soit pour des sites de vente de livres ou bien des bibliothèques.

1.3 Jeu de données

En ce qui concerne le jeu de données, nous avons utilisé un fichier texte contenant approximativement 17000 livres. Dans ce jeu de données, nous pouvions trouver pour un livre :
- Un identifiant WikiID
- Un identifiant FreebaseID
- Le titre du livre
- L'auteur du livre
- La date de parution
- Les genres de ce livre
- Le résumé du livre

Nous allons maintenant nous pencher sur la conception, répartition des tâches et réalisation de notre projet.

Chapitre 2

Conception, répartition et implémentation

2.1 Conception et répartition du travail

2.1.1 Détails de conception

En ce qui concerne la conception de notre projet, nous avons tout d'abord fait le point sur comment traiter les données que nous avons utilisées ainsi que sur la répartition des tâches que nous aborderons dans quelques instants.

Pour le traitement des données, nous avons décidé qu'il fallait tout d'abord convertir notre fichier texte en fichier CSV. Ensuite, une fois que nous avons récupéré le fichier CSV avec les données à l'intérieur, il fallait regrouper les catégories similaires en un seul et même groupe. Après avoir regroupé les catégories similaires, on devait faire le prétraitement de nos données. Pour finir, il fallait classifier les résumés en fonctions des mots utilisés à l'intérieurs

2.1.2 Répartition du travail

Pour ce qui est de la répartition du travail, nous avons décidé de travailler en salon vocal puisque nous devions travailler à distance. Ainsi, nous partagions notre écran pour développer et nous discutions de ce que nous pouvions faire. Nous avons aussi utilisé un repository github pour le projet afin d'avoir accès au code chacun de notre côté. Cela nous a permis de pouvoir travailler même lorsque nous n'étions pas disponibles en même temps.

Nous allons maintenant aborder comment nous avons implémenté ce que nous avions prévu pour la conception.

2.2 Implémentation

2.2.1 Traitement des données

Parseur

Pour le traitement des données, nous avons :
Tout d'abord, nous avons créé un script python « parseur.py » nous permettant de convertir le fichier texte de base en fichier csv nommé « data.csv ».

Wrapper / Grouper les genres similaires

Ensuite, nous avons réalisé un autre script « genreWrapper.py » qui permet de regrouper les catégories similaires en une seule principale et écrit dans le fichier « data-grouped.csv ».

2.2.2 Preprocessing et classification

Preprocessing

Pour la suite, nous avons fait un script pour le prétraitement des données qui s'appelle « preprocessing.py ». Ce script permet de modifier le texte du résumé pour qu'il soit utilisable par le réseau de neurone. On met tout le texte en minuscule, on supprime la ponctuations et symboles. Ensuite on « tokenise », retire les « stopwords » et pour finir, on fait du « stemming » sur les mots.
Dans le prétraitement, nous séparons aussi nos données de « data-grouped.csv » en deux afin d'entrainer et tester notre IA : - dataSet-Train.csv pour l'entraînement (80% d'entrainement et 20% de test de prédiction) - data-test.csv pour les tests avec tous les livres qui n'avaient, de base, pas de genres

Classification

Après le prétraitement, on classifie les données prétraitées « dataSet-Train.csv » à l'aide de notre script « classifier.py ». Dans ce script, nous pouvons retrouver notre système de classification des résumés en fonction des mots contenus dans ces résumés. Enfin, nous avons un script main qui permet de prendre des livres de notre fichier « data-test.csv » et d'essayer de retourner le genre de ce livre.

Après avoir abordé la conception et l'implémentation de notre projet, nous allons maintenant aborder les résultats que nous avons obtenus ainsi que les problèmes rencontrés.

Chapitre 3

Résultats obtenus et problèmes rencontrés

3.1 Résultats obtenus

Nous allons maintenant voir les résultats que nous avons obtenus. Après avoir entrainé notre IA sur les 80% dédiés à l'entrainement pur, nous faisons des prédictions sur les 20% restants des données de « dataSet-Train.csv »

Lors de ces prédictions, notre IA parvient bien à déterminer le genre du livre en fonction du résumé avec une précision d'environ 95%. Ces résultats sont assez convaincants cependant lorsque nous essayons de trouver le genre du livre avec les données de « data-test », nos prédictions ne fonctionnent pas comme on le souhaiterait. En effet, les prédictions obtenues ne sont quasiment que du genre « Fictions » même quand le livre n'appartient pas à ce genre-là. Nous avons donc cherché à comprendre d'où pouvait provenir notre erreur et nous avons découvert que les données utilisées contiennent beaucoup trop de livres du genre « Fictions » par rapport aux autres genres.

Cela nous ramène donc aux problèmes rencontrés lors de notre projet.

3.2 Problèmes rencontrés

3.2.1 Problèmes rencontrées

Le premier problème que nous avons pu rencontrer est donc notre jeu de données. En effet il y a un trop de livres du genre « Fictions » par rapport aux autres genres. Ce problème est sûrement dû au groupement de tous les genres en similaires en genre principal. Il est donc possible que si nous regroupions en plus de groupes différents nous aurions une plus grande diversité de prédictions pour nos données de tests.

Le second problème que nous avons rencontré est que notre dataset avait seulement deux colonnes. Ces colonnes étaient : - words (étant le résultat du préprocessing. Ce résultat est une liste de mots tokenisés puis stemmés) - genre (correspondant aux genres du résumés. Ces genres sont contenus dans une liste) Le problème venait de la colonne genre qui était considérée comme une seule valeur alors qu'elle en contenait plusieurs.

3.2.2 Solutions proposées

Pour résoudre ce problème, nous avons décidé de mieux séparer les genres en créant une colonne pour un genre spécifique. Chaque colonne contient ainsi soit un 1 si le livre appartient à ce genre soit un 0 si ce n'est pas le cas. En résolvant ce problème, nous sommes passés d'une précision de prédiction de 40% à une précision d'environ 95% ce qui correspond à notre précision actuelle.

Chapitre 4

Améliorations possibles

4.1 Améliorations

Pour les améliorations possibles, nous pourrions prendre un jeu de données plus diversifier au niveau des genres et avec un meilleur ratio de genres. C'est-à-dire prendre le même nombre de livres de genres différents (par exemple 100 livres de fiction, 100 livres d'action, 100 autobiographies, etc. . .).

4.2 Autres applications possibles

En ce qui concerne les autres applications possibles, on pourrait par exemple le faire sur des synopsis de films ou bien alors de séries ou encore sur les jeux vidéo.

Chapitre 5

Conclusion

Pour conclure, nous avons dû changer notre sujet par manque de données disponibles et le sujet que nous avons finalement choisit nous correspond et sommes content de l'avoir réaliser.
En ce qui concerne la conception et l'implémentation de notre projet, le travail à domicile a rendu cela un peu plus complexe mais nous sommes parvenus à faire ce que nous souhaitions faire.
Nous avons d'abord parser nos données puis fait le prétraitement et enfin la classification sur les données faites pour l'entraînement de notre réseau. Puis, nous avons rencontrés quelques problèmes notamment au niveau de notre dataset mal réparti au niveau des pourcentages de genres de livres présents. Enfin malgré ces petits problèmes, nous avons pu avoir des résultats concluants sur nos prédictions sur nos données d'entraînements.