SAE 1.05 Emilien Busson
Mattéo Talazac

Traiter les données
Orthobot

Introduction :

L'avènement de l'ère numérique a engendré une explosion exponentielle de la quantité de données générées chaque jour. Dans ce trafic les réseaux sociaux y sont pour beaucoup, ils créent une quantité d'information très importante dut en partie au nombre de poste par jour qui est absolument délirant. Twitter enregistre 9120 postes par seconde dans le monde quant à lui Reddit contient 199 millions de postes. Mais dans toutes les informations que l'on peut trouver à travers le web nous remarquons une chose, la langue française se détériore en effet nous pouvons remarquer que l'orthographe des mots et les fautes de français en générales évoluent exponentiellement avec l'évolution des réseaux sociaux. Mais aussi internet et les réseaux sociaux sont connues pour la stérilité et la présence d'injure dans un bon nombre de débats.

Après cette prise de conscience nous avons entrepris de plonger dans le monde complexe du traitement des données dans notamment dans le but de quantifier cette dégradation dans les débats le langage français sur internet. Pour nous rendre compte si ces conséquences sont aussi mondial nous avons ouvert notre projet a d'autre langues.

Ce rapport expose en détail les différentes étapes de notre projet, de la mise en œuvre à la mise en œuvre pratique des techniques de traitement des données. Nous mettons en lumière les défis rencontrés, les méthodologies adoptées, ainsi que les résultats obtenus, tout en soulignant l'impact potentiel de notre travail sur la résolution de problèmes concrets.

Table des matières

Description du sujet :

Dans le cadre de cette SAE 1.05, nous avions la liberté du sujet le temps qu'il répondaient au problématique initial : Traiter des données. Le choix de notre sujet c'est effectué sur plusieurs critères.

L'originalité : Nous avons souhaité que notre projet est un trait d'originalité, c'est à dire qu'il ne soit pas des plus commun

Complexité : Le choix de notre sujet c'est également effectuer par apport à la complexité qu'il engendrer. En effet nous souhaitions un sujet un minimum complexe pour qu'il nous apporte un intérêt une motivation pour le réaliser.

Faisabilité : En relation avec la complexité il fallait que notre sujet soit faisable, le critère de faisabilité a donc été un facteur important dans la sélection du sujet.

C'est avec ces trois critères que nous avons entrepris les recherches d'un sujet de projet. Pour l'originalité, nous nous sommes dit que faire un travail autour des réseaux sociaux n'étaient pas des plus courant et que c'était plutôt original. De plus un travail sur les réseaux sociaux, peut apporter une forme de complexité. En effet, il faut trouver un moyen de récupérer les données et de savoir comment on va les traiter/analyser. C'est ici qu'intervient notre critère de faisabilité. La faisabilité va s'effectuer sur le fait de pouvoir récupérer les données. En effet tous les réseaux sociaux, ne permettent pas la récupération de données, soit de manière gratuite ou bien d'aucune manière quel qu'il soit. Par exemple Twitter propose une API, c'est à dire une interface de programmation, diminuant la complexité, voir permettant d'avoir un accès légal par le réseau social de récupérer ces données. Cependant l'API de twitter est payante et a un cout assez élever (à un prix d'entrée de minimum 100$/mois), impossible à prendre en charge pour un projet de SAE. Nous avons regardé si Reddit, proposait une API et si oui à quel prix. Nous avons constaté que Reddit proposait une API mais qui plus est gratuite. C'est dans ce contexte et avec ces informations que nous avons décidé de basée notre projet sur les données de Reddit. Il nous reste plus qu'à déterminer ce que nous allons faire avec les données de Reddit.

Nous voyons souvent passé des sujets dans les journaux télévisés et écrit, qui accusent de choses et d'autres les réseaux sociaux. Le sujet qui remonte souvent, porte sur la qualité du français employé par les jeunes et notamment à causes des réseaux sociaux. Nous avons donc décidé de vérifier par nous même si la communauté de reddit était elle aussi concerné par ces sujets d'actualités ou non. Notre projet portera donc sur la qualité orthographique des posts de Reddit. Nous allons regarder entre deux SubReddits lequel des deux fait le plus de fautes de français et surtout les types de fautes qui sont effectué. Pour rendre notre sujet plus complet nous avons pris l'initiative de traiter d'autres données. Parmi les sujets les plus courant à propos des réseaux sociaux dans les journaux, nous avons aussi remarqué que la courtoisie et la violence poster sur les réseaux sociaux en faisaient partie. Nous avons donc décidé également, de regarder si la courtoisie et le respect était aussi présent sur Reddit ou si c'était un Réseau social sans respect et sans violence au contraire de twitter qui est souvent pointer du doigt pour ce type de problème. Enfin nous avons souhaitez regarder l'activité sur reddit et sur les différents sub.

NB : Nous avons uniquement évoqué la langue française, mais l'entièreté de notre projet peut également fonctionner avec l'anglais

Description du Projet :

A-Description des données et traitement :

Dans ce projet nous ne disposons pas d'un fichier de données brut qui pourrait être stocker dans un CSV. Les données que nous allons traiter se feront grâce à des requêtes effectuées à l'API de Reddit. Dans notre projet, nous pouvons distinguer trois requêtes majeures. La première requête nous permet de récupérer l'identifiant des posts au sein d'un subreddit. La deuxième requête majeure nous permet de récupérer le contenu des commentaires d'un post spécifique. Cette requête fonctionne notamment grâce à la première requête. En effet cette requête ne peut être effectué sans l'identifiant du post en question. La dernière requête consiste à récupérer la date et l'heure à laquelle un post est publié.

Cependant pour qu'une autre partie du projet fonctionne notamment sur la partie de l'analyse orthographique, il faut faire d'autre requêtes. Ces requêtes sont effectuées sur un serveur local qui s'initialise au lancement du code. Ces requêtes permettent de faire l'analyse de chaque commentaire. Après analyse, la requête nous renvoie en réponse une liste de fautes repérées.

Pour effectuer les requêtes auprès de l'api reddit, nous utilisons la librairie officielle de reddit pour python : PRAW. Cette librairie ne nous limite pas aux requêtes que nous effectuons. En effet, grâce à celle-ci nous pourrions par exemple récupérer les noms des auteurs des commentaires des posts. Ce n'est qu'un exemple parmi tant d'autres.

Pour effectuer la correction orthographique nous utilisons la librairie language_tool_python. Cette librairie n'a spécialement d'autre fonctions que la correction orthographique. Cependant elle peut prendre certains paramètres en compte pour modifier son comportement sur la correction.

A partir de ces requêtes, cela nous créer des jeux de données qu'il est possible de traiter. Pour la partie analyse orthographique nous avons plusieurs traitements qui rentre en compte. En effet il y a dans un premier temps le traitement porter sur les données récupérer grâce à l'api de Reddit. Puis il y a les données que nous obtenons après l'analyse de celle récupéré grâce à l'API. Ce traitement consiste à ordonnée et classifier ces données, en deux catégories. La première est composée des types de fautes répertoriées. La deuxième est celle qui contient les données en elles même. Le nombre de fois ou le type de faute a été relevé et ça pour chaque subreddit étudiés. Ces deux catégories vont nous permettre de faire un diagramme en bâtons grâce à la librairie Matplotlib.

Ces requêtes nous permettent aussi de faire de l'analyse d'activités. Dans un premier temps nous récupérons les dates et heures de publications de chaque posts sur les subreddit sélectionnés. A partir de ces données cela nous permet de faire un calcul de moyenne. Cependant, le calcul de moyenne ne peut pas être fait avec les modules de base de python il faut importer la librairie datetime et les modules datetime et timedelta. Ces modules permettent dans un premier temps de formater l'heure sur un format plus lisible que le format initial des données qui sont au format Unix. C'est le module timedelta qui nous permet de faire le calcul de moyenne. C'est le résultat de ce calcul de moyenne qui est renvoyé à l'utilisateur.

B-Algorithme

Vous trouverez ci-dessous les algorithmes de chacune des analyses et traitement des données :

Analyse Orthographique :

Demander à l'utilisateur d'entrée le nom du premier subbreddit
Demander à l'utilisateur d'entrée le nom du deuxième subbreddit
Demander à l'utilisateur d'entrée le nombre de posts à étudier par subbreddit

Demander à l'utilisateur d'entrée le nombre de commentaires à étudier par posts sélectionné

Pour chaque subreddit :
Accéder au subreddit indiquer par l'utilisateur
Définir une liste qui pourra stocker les identifiants des posts
Temps que la limite de posts à étudier par l'utilisateur n'est pas atteinte :
Récupérer l'identifiant du post et le stocker dans la liste prévue à cet effet

Pour chaque identifiant de posts stocker dans la liste :
Définir un dictionnaire qui stockera chaque type de fautes et le nombre de fois qu'elle a été recensé
Temps que la limite de commentaires à étudier n'est pas atteinte :
Pour chaque commentaire :
Pour chaque phrase du commentaire :
Vérification si le commentaire est un lien
Si le commentaire est un lien :
Le commentaire n'est pas pris en compte et on passe au commentaire suivant
Sinon :
Analyse de la phrase et recensement des fautes
Si une faute est repérée et que son type est déjà présent dans le dictionnaire :
On ajoute 1 au type de faute dans le dictionnaire
Si la faute n'est pas repérée :
On ajoute le type de faute avec une valeur de 1

On compare la taille des deux dictionnaires :
Si la taille du dictionnaire du premier subreddit > dico deuxieme subreddit :
On définit un nouveau dictionnaire categories qui prends dans un premier temps les valeurs du dictionnaire du premier subreddit
Pour clef du dictionnaire du deuxième subreddit :
Si la clef n'est pas présente dans le dictionnaire categories :
On ajoute la clef avec sa valeur associée dans le dico categories
Sinon :
Aucune action effectuer

Définition d'une liste valeurs_set1 qui va regrouper uniquement les valeurs du dico du 1er subreddit
Définition d'une liste valeurs_set2 qui va regrouper uniquement les valeurs du dico du 1er subreddit
Pour chaque clef du dictionnaire catégories :
Si la clef n'est pas présente dans le dictionnaire initial du deuxième subreddit :
On ajoute à valeurs_set2 la valeur 0
On ajoute à valeurs_set1 la valeur associée au dictionnaire
Si la clef n'est pas présente dans le dictionnaire initial du premier subreddit :
On ajoute à valeurs_set1 la valeur 0
On ajoute à valeurs_set2 la valeur associée au dictionnaire
Sinon :
On ajoute à valeurs_set1 et valeurs_set2 leur valeur associée

Si la taille du dictionnaire du deuxième subreddit > dico premier subreddit :
On définit un nouveau dictionnaire catégories qui prends dans un premier temps les valeurs du dictionnaire du deuxieme subreddit
Pour chaque clef du dictionnaire du premier subreddit :
Si la clef n'est pas présente dans le dictionnaire catégories :
On ajoute la clef avec sa valeur associée dans le dico catégories
Sinon :
Aucune action effectuer

Définition d'une liste valeurs_set1 qui va regrouper uniquement les valeurs du dico du 1er subreddit
Définition d'une liste valeurs_set2 qui va regrouper uniquement les valeurs du dico du 1er subreddit
Pour chaque clef du dictionnaire catégories :

Si la clef n'est pas présente dans le dictionnaire initial du deuxième subreddit :
On ajoute à valeurs_set2 la valeur 0
On ajoute à valeurs_set1 la valeur associée au dictionnaire
Si la clef n'est pas présente dans le dictionnaire initial du premier subreddit :
On ajoute à valeurs_set1 la valeur 0
On ajoute à valeurs_set2 la valeur associée au dictionnaire
Sinon :
On ajoute à valeurs_set1 et valeurs_set2 leurs valeurs associées

Génération d'un graphique en plot qui prend en valeurs d'abscisses les types de fautes recenser et en ordonnées le nombre de fois où elles ont été recensées

Analyse de la qualité de discussion :

Analyse de la qualité de discussion :
Demander si l'utilisateur souhaite travailler sur un post reddit spécifique ou un subreddit

Si l'utilisateur choisi le post :
Demander le lien du post
Demander la langue à étudier
Demander le nombre de commentaires à étudier
Récupérer l'identifiant du post en question et le stocker dans une variable
Récupérer la liste de commentaire lier a l'identifiant du post
Définir la liste polarité
Définir un compteur initialiser a 1
Temps que le compteur ! = la limite fixée par l'utilisateur ou liste de commentaire pas atteint :
Pour chaque commentaire, on calcul sa polarite et on l'ajoute à la liste de polarite
On ajoute 1 au compteur
Calcul de la moyenne des éléments de la liste
Retourne la moyenne obtenue et l'affiche dans l'interface

Si l'utilisateur choisi subreddit :
Créer une liste vide posts
Temps que la limite de pots fixé par l'utilisateur ne soit pas atteinte :
Récupérer l'identifiant de chaque posts et les ajouter à la liste posts
Définir une liste polarite vide
Pour chaque identifiant des posts de la liste posts :
Temps que le compteur ! = la limite fixée par l'utilisateur ou liste de commentaire pas atteint :
Pour chaque commentaire, on calcul sa polarite et on l'ajoute à la liste de polarite
On ajoute 1 au compteur
Calcul de la moyenne des éléments de la liste
Retourne la moyenne obtenue et l'affiche dans l'interface

Analyse de la qualité de l'activité :

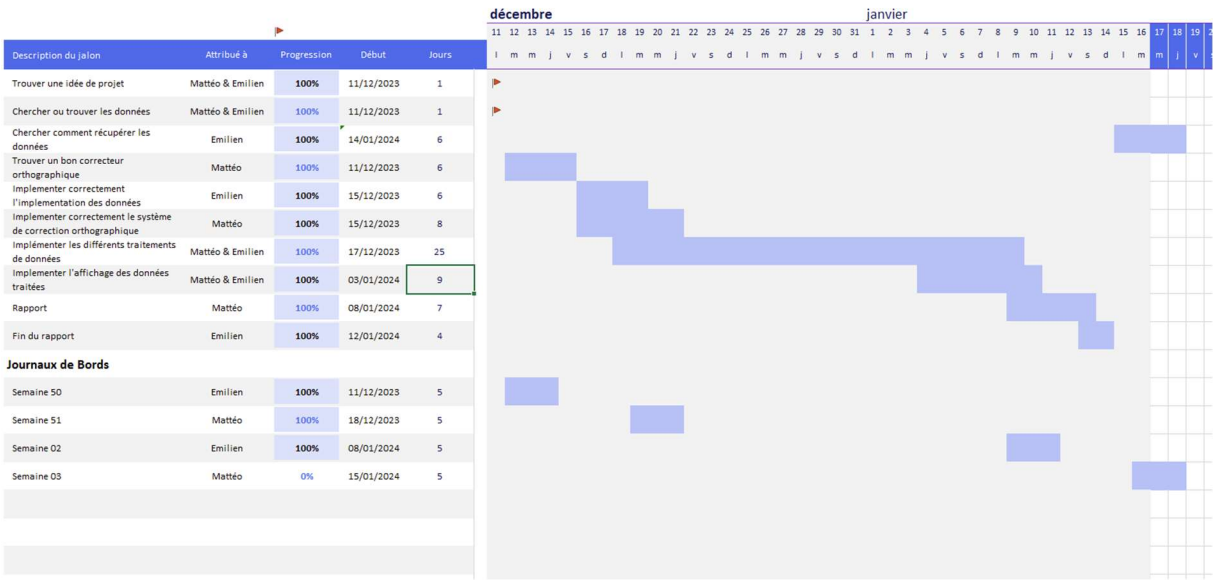
Entrer le nom du sub reddit
Le nombre de post à étudier
Accéder au subreddit indiquer par l'utilisateur
Définir une liste objets qui pourra stocker les dates de chaque post
Temps que la limite de posts à étudier par l'utilisateur n'est pas atteinte :
Ajouter à la liste objets la date et l'heure de chaque post

Créer une liste time_diff qui stockera la différence entre deux dates
Pour chaque élément de la liste objet :
Ajouter à la liste time_diff la différence entre l'objet précédent et l'actuel

Effectuer la moyenne entre tous les éléments de la liste
Regarder si le résultat et retourner avec des jours
S'il y a un jour :
On retourne le résultat tel quel
Si le résultat est renvoyé en jour négatif
Retourner uniquement l’heure, les minutes et les secondes
Sinon :
Renvoyer uniquement l’heure, les minutes et les secondes

Gestion du projet :

Afin de mener à bien ce projet nous avons mis en place des outils de gestion de projet. Nous avons notamment mis en place un diagramme de Gantt nous permettant de mettre en place une certaine organisation et une répartition des taches efficaces. Ce diagramme n'a pas forcément été mis à jour sur l'avancement des taches mais nous nous sommes appuyés sur celui-ci tout au long du projet pour savoir où nous en étions et quelles taches nous devions faire.



Description du jalon	Attribué à	Progression	Début	Jours
Trouver une idée de projet	Mattéo & Emilien	100%	11/12/2023	1
Chercher ou trouver les données	Mattéo & Emilien	100%	11/12/2023	1
Chercher comment récupérer les données	Emilien	100%	14/01/2024	6
Trouver un bon correcteur orthographique	Mattéo	100%	11/12/2023	6
Implementer correctement l'implementation des données	Emilien	100%	15/12/2023	6
Implementer correctement le système de correction orthographique	Mattéo	100%	15/12/2023	8
Implémenter les différents traitements de données	Mattéo & Emilien	100%	17/12/2023	25
Implementer l'affichage des données traitées	Mattéo & Emilien	100%	03/01/2024	9
Rapport	Mattéo	100%	08/01/2024	7
Fin du rapport	Emilien	100%	12/01/2024	4
Journaux de Bords				
Semaine 50	Emilien	100%	11/12/2023	5
Semaine 51	Mattéo	100%	18/12/2023	5
Semaine 02	Emilien	100%	08/01/2024	5
Semaine 03	Mattéo	0%	15/01/2024	5

Conclusion :

Pour Conclure sur ce projet nous allons évoquer plusieurs points. Cette SAE, nous a permis d'améliorer une nouvelle fois nos capacités à travailler en groupe. En effet ce faisant en binôme, il a fallu se répartir correctement le travail afin qu'il n'y soit pas de disparité et que tout le monde puisse être efficace. Ce projet nous a permis également d'acquérir de nouvelles compétences et d'en améliorer d'autres. En effet, nous avons commencé la SAE avec uniquement les bases de la programmation vu dans la ressource R1.07. Ces bases ont pu parfaitement être appliquées et travaillées. De plus nous avons pu en acquérir de nouvelles notamment avec le travail des API qui a été très enrichissant. Nous avons pu aussi découvrir le fonctionnement des interfaces graphiques notamment avec la librairie Tkinter. Il est vrai qu'elle n'est pas très poussée sur notre projet, mais cela nous a permis de nous initier à cette librairie et au fur et à mesure des projets futur.

Bilan :

Ce projet n'est pas parfait, et il est possible de l'améliorer sur certains plans. Dans un premier temps, il est surement possible d'améliorer notre code. Il est fonctionnel mais est très certainement perfectible. Cependant, ce projet ne s'est pas fait sans obstacle. En effet tout d'abord nous souhaitions travailler sur les données de twitter cependant comme nous l'avons évoqué plus haut il est impossible de manière gratuite et légal d'utiliser les données Twitter. Nous avons perdu beaucoup de temps sur ce problème avant de nous resigner à changer de réseaux sociaux et à passer sur Reddit.

Nous avons eu d'autres obstacles sur le projet. Notamment sur la gestion des fautes d'orthographe. En effet nous avions besoin d'un outil puissant et complet. Cependant il a été difficile de trouver une librairie adaptée à nos attentes et il a fallu faire des recherches pendant longtemps retardant certaines parties du projet.

Enfin il est à noter que la librairie de Reddit n'est pas toujours à jour sur certaines données récupérable ce qui peut rendre impossible certaines idées. Par exemple nous souhaitions faire un graphique sur la fréquence d'activité par rapport aux années mais cela est impossible avec des données incomplètes