**Développement d'un Moteur de Recherche d'Information Maison**

Table des matières

[Présentation du projet 1](#_Toc155725071)

[Spécifications 1](#_Toc155725072)

[Analyse 2](#_Toc155725073)

[Conception 2](#_Toc155725074)

[Problèmes rencontrés 2](#_Toc155725075)

[Exemple d’utilisation 2](#_Toc155725076)

[Validation 3](#_Toc155725077)

[Maintenance 3](#_Toc155725078)

# Présentation du projet

Ce projet vise à créer une solution personnalisée de recherche d'information, en mettant l'accent sur la mise en œuvre de nos propres algorithmes plutôt que de recourir à des bibliothèques existantes telles que scikit-learn ou nltk. L'objectif principal est d'approfondir notre compréhension des mécanismes fondamentaux qui sous-tendent ces outils "prêts à l'emploi".

Vous pouvez retrouver ce projet à l’adresse suivante : <https://github.com/AlysonTT/Python>

# Spécifications

L'interface graphique offre la possibilité d'explorer diverses actions, telles que suivre l'évolution temporelle d'un mot, analyser la fréquence d'un terme, effectuer une recherche par mots-clés, auteurs ou date spécifique, ainsi que visualiser l'intégralité du corpus. De plus, la comparaison de deux documents est également incluse dans les fonctionnalités.

L'évolution temporelle d'un mot est mise en évidence à travers une frise temporelle, offrant une représentation visuelle claire des variations de la fréquence de ce mot au fil du temps.

Les utilisateurs peuvent accéder à la fréquence de tous les mots présents dans le corpus, ainsi qu'à la mesure de l'importance d'un terme, calculée selon la méthode TFxIDF, grâce à un bouton dédié sur notre interface graphique.

La fonctionnalité de recherche avancée permet aux utilisateurs de spécifier des critères tels que des mots-clés, des auteurs, et/ou une date spécifique, simplifiant ainsi l'exploration ciblée du corpus.

Une option pratique est disponible pour afficher l'ensemble du corpus, offrant une vue d'ensemble facilitant l'analyse globale des documents présents.

Une fonctionnalité intégrée permet de comparer deux documents et d'indiquer la présence éventuelle de mots communs entre eux.

Ces caractéristiques combinées font de notre interface graphique un outil complet et convivial, permettant aux utilisateurs d'explorer, d'analyser et de comparer efficacement le contenu de leur corpus de textes.

# Analyse

Notre environnement de travail s'est articulé autour de Visual Studio Code, GitHub, et la bibliothèque Tkinter pour l'interface graphique. Visual Studio Code a été notre IDE principal, offrant une expérience de développement fluide. GitHub a facilité la collaboration au sein de notre binôme en permettant le suivi des versions et la gestion du code source. Pour l'interface utilisateur, nous avons choisi Tkinter, une bibliothèque Python, pour sa simplicité d'utilisation et son intégration harmonieuse avec notre projet.

En ce qui concerne les données identifiées dans les spécifications, nous avons utilisé les API de Reddit et arXiv. Pour interagir avec l'API de Reddit, nous avons employé la bibliothèque PRAW, tandis que pour l'API arXiv, nous avons utilisé la bibliothèque urllib.

Pour Reddit, nous avons récupéré tous les articles disponibles, cependant, une limite maximale d'articles a été définie alors que pour arXiv, une requête a été formulée pour extraire des articles contenant les termes "clustering" et "Dirichlet". Ces données extraites serviront de base pour notre analyse ultérieure.

Voici le diagramme de classes final de ce projet :

**Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement**

# Conception

Au début du projet, Alyson débutait les travaux dirigés car elle avait des cours avant Maude. Ensuite, Maude prenait le relais pendant son créneau horaire de cours. Si le travail dirigé n'était pas terminé, nous le poursuivions ensemble lorsque nous n'avions pas de cours en commun.

Concernant la partie graphique, Alyson a pris en charge la création de l'ensemble de l'interface, incluant l'affichage des résultats ainsi que la section dédiée à la comparaison de deux documents. Maude, de son côté, s'est focalisée sur la mise en place d'options permettant à l'utilisateur de sélectionner des critères tels que les sources, les auteurs, et une date spécifique. Elle a également travaillé sur la frise temporelle en intégrant les éléments graphiques nécessaires pour faciliter ces différentes actions.

## Problèmes rencontrés

**????**

## Exemple d’utilisation

**????**

# Validation

Chaque fonctionnalité implémentée, telle que la recherche spécifique ou la frise temporelle, a été testée à l'aide de fichiers dédiés. Cela nous a permis de développer un code fonctionnel sur une branche distincte, que nous avons ensuite fusionnée avec la branche principale. Cette approche nous a assuré d'avoir constamment un projet opérationnel à chaque étape du développement.

# Maintenance

**Evolution possible ??**