**Université Lumière Lyon 2 Campus porte des alpes de Bron**

**Institut de la communication**

Une image contenant texte

Description générée automatiquement

**Master 1 Informatique**

**Cours de programmation avancée**

MOTEUR DE RECHERCHE A l’AIDE DE PYTHON

**Présentée par :**

Seynabou FAYE

Paul Naomi KALDJOB **Professeur : Julien VELCIN**

**Année académique : 2022/2023**

# Introduction

Dans le cadre de l’enseignement du cours de Programmation avancée du Master 1 informatique, nous avons réalisé un projet pour l’évaluer nos connaissance pour cet module. Ce module programmation avancée comporte un volet théorique (cours) et un autre pratique (travaux pratiques et projet). La fin de cet module est marquée par un projet qui nous permet de réaliser un programme python en passant par toutes les ´étapes de son cycle de vie : spécifications, analyse et conception, codage, vérification et maintenance. Nous avons donc créé une application qui nous permet de récupérer les mots les plus fréquents dans un document et de voir l’évolution de ces mots sur les années de notre étude.

Notre application a pour but d’aider les chercheurs en SHS à explorer des documents en adoptant une approche comparative tout en retrouvant facilement les informations les plus pertinentes.

# 1-Spécification

Des chercheurs en SHS (linguistique, social, etc) de l’université Lyon 2 veulent explorer des documents contenant énormément d’informations intéressantes, mais la masse de données rend compliquée leur exploration. Pour aider ces chercheurs, nous devons créer un outil permettant une analyse comparative de deux corpus (comparer les articles issus de Reddit et de ceux d’Arvix). Pour cela, nous allons développer un outil NLP avec python qui leur permettra à la fin du processus d’implémentation de pouvoir récupérer les mots les plus fréquents et de visualiser l’évolution de ces mots selon les trimestres de l’année pour chaque corpus via une interface kivy. Ainsi, ils pourront donc faire une comparaison des corpus. Lien github : https://github.com/naomi-kaldjob/projet-python/issues.

# 2-Analyse

Kivy est un cadre de développement d'applications gratuit, open-source et multiplateforme qui fonctionne sur diverses plateformes comme Android, iOS, Windows, Linux, etc. Il utilise le langage Python pour le développement. Kivy permet de développer des applications mobiles et des logiciels d'application multi-touch avec une interface utilisateur naturelle (NUI). Il permet aux utilisateurs de créer des widgets personnalisés très facilement en utilisant le langage intermédiaire connu sous le nom de kV.

Pour réaliser ce projet, nous avons utilisé plusieurs classes tels que :

* La classe Document qui est une classe qui contient toutes les informations concernant le document tels que (le titre du document, le nom de l’auteur, la date de publication, l’url source et le contenu textuel). Cette classe possède des méthodes tels que méthodes d’initialisation(\_\_init\_\_), méthode qui renvoie le texte à afficher (\_\_repr\_\_).
* La classe Author qui comporte toutes les informations sur chaque auteur tels que son nom, le nombre de documents publiés et une liste avec tous ses documents. Cette classe possède des méthodes tels que méthodes d’initialisation de variables (\_\_init\_\_), méthode qui compile tous les documents d’un auteur (add()).
* La classe corpus regroupe toutes les informations des autres classes. Cette classe possède des méthodes tels que méthodes d’initialisation de variables (\_\_init\_\_), des méthodes qui d’afficher les éléments du corpus triés selon la date et le titre (add() et show()),des méthodes qui retourne les passages des documents(search() et concorde()) , une méthode qui applique une chaine de traitement à une texte (nettoyer\_texte()) et une méthode permettant d’ afficher plusieurs statistiques textuelles(stats()).

# 3-Conception

## 3.1-Répartition des tâches

Les tâches ont été reparties équitablement pour mener à bien le projet. Chaque membre du groupe s’occupait de différents travaux pratiques du projet. Un membre s’occupait de la conception de l’application et un autre membre s’occupait de terminer les travaux dirigés.

## 3.2- Les problèmes rencontrés

La création de l’interface a été difficile. Nous avions rencontré des problèmes à maitriser le langage Kivy parce que nous n’avions pas été formé sur le langage kivy. Nous avons eu des difficultés à créer des champs d’entrées, des boutons et des graphiques. Nous avons dû chercher sur plusieurs sites de programmation. Nous avons eu des difficultés à créer des boutons radio sous kivy et à afficher les meilleures documents avec les meilleures scores par source.

## 3.3- Exemple commenté d’utilisation du programme du début à la fin

Le programme se présente comme ceci :

Une image contenant texte

Description générée automatiquement

* Un champ pour rentrer une thématique
* Un bouton pour défiler entre les pages

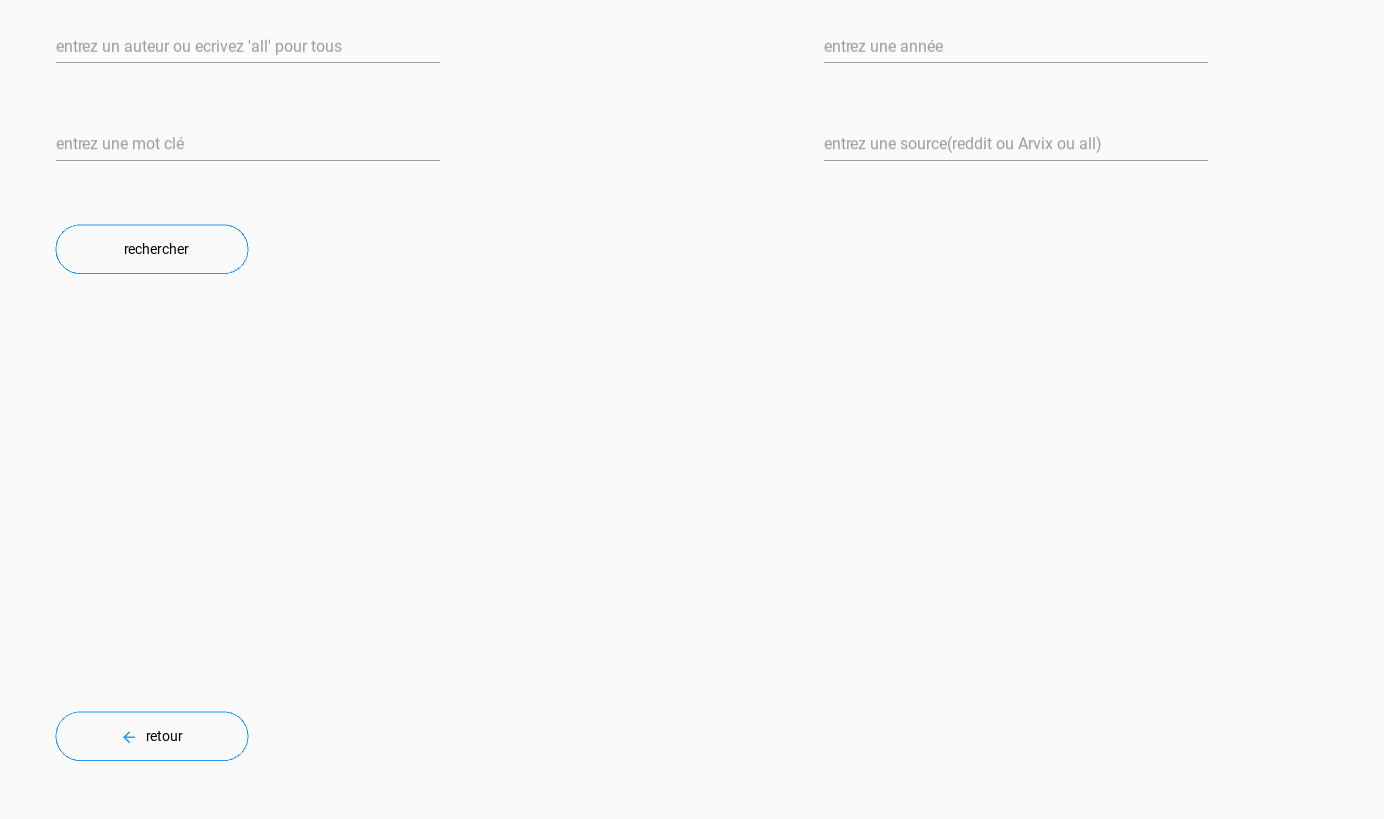
Si j’entre une thématique comme ‘clustering ‘, il en ressort certains résultats :

Une image contenant texte

Description générée automatiquement

Ces résultats sont des nuages de mots. Les nuages de mots permettent de percevoir les mots les plus fréquents dans un corpus. Nous observons 2 nuages de mots, l’un pour les articles d’arvix et l’autre celui de reddit.

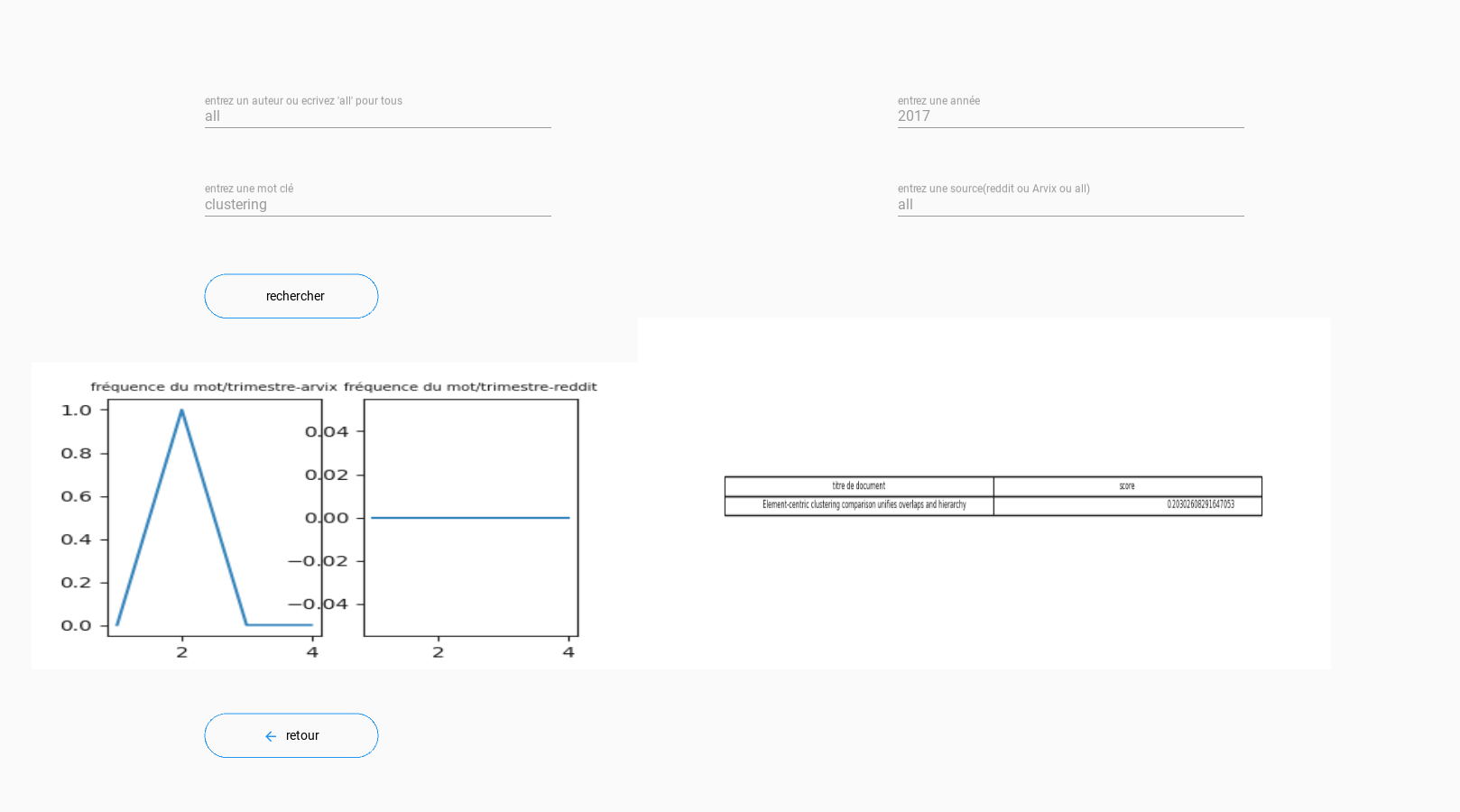
Si on clique sur le bouton suivant, on tombe sur cette page :



On observe 4 champs d’entrées :

* Un champ de mot-clé
* Champ d’auteur
* Champ de l’année
* Champ de la source

En remplissant les champs comme ci-dessous, on obtient ces résultats



Un graphique qui suit l’´évolution temporelle d’un mot donnée suivant l’année, le nom de l’auteur pour chaque source(Reddit, Arvix ou les deux) et une table qui affiche le titre du document ou l’importance du mot est forte.

## 4- La validation

### 4.1 Tests unitaires

Chaque méthode de chaque classe devrait réaliser une action précise. Chaque méthode a été testée individuellement dans le fichier tp. Prenons l’exemple de la méthode search de la classe corpus. Cette méthode retourne des passages de documents contenant le mot en entré.

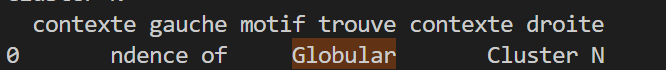
Choisissons un mot comme globular. Affichons tous les documents et recherchons le mot entré dans ces tous les documents. Ce mot n’est présent qu’une fois dans ces documents.

Une image contenant texte

Description générée automatiquement

Nous allons vérifier si les mots qui précèdent le mot entré sont les même lors de l’utilisation des fonctions search et concorde.





Les contextes à gauche et à droite sont similaires qu’aux phrases à gauche et à droite de tous les documents.

## 4.1.2 test globaux

Le programme se présente comme ceci :

Une image contenant texte

Description générée automatiquement

* Un champ pour rentrer une thématique
* Un bouton pour défiler entre les pages

Si j’entre une thématique comme ‘clustering ‘, il en ressort certains résultats :

Une image contenant texte

Description générée automatiquement

Ces résultats sont des nuages de mots. Les nuages de mots permettent de percevoir les mots les plus fréquents dans un corpus. Nous observons 2 nuages de mots, l’un pour les articles d’arvix et l’autre celui de reddit.

# 5-Maintenance

Ce logiciel est très efficace et on pourrait l’ajouter certaines fonctionnalités tels que le fait de filtrer sur les auteurs, les titres, les années et afficher les commentaires de chaque document. Pour cela, ajouter une liste déroulante pour les auteurs, les titres et les années sur l’interface ainsi que d’ajouter dans la table créée une colonne commentaires. L’interface sélectionne les documents avec les meilleurs commentaires. Cette dernière fonctionnalité est difficile à implémenter parce qu’il faudrait analyser les sentiments des commentateurs pour déterminer s’ils sont positifs ou pas.

# Conclusion

Ce projet programmation python est une mission d’étude. Pour réaliser ceci, nous avons recouru à différentes technologies et de développement nécessaires pour répondre aux besoins.

A travers ce projet, nous avons appris également à travailler en binôme avec des idées divergentes

Ce projet est pour nous une occasion, d’approfondir d’une part nos connaissances théoriques acquises durant ce module, et d’autre part la pratique de nouvelles technologies.

Nous avons rencontré, tout au long du projet, des problèmes, dont certains ont pu être réglés et d’autres non.