**Rapport Projet Python**

1. Description du projet

Le projet réalisé dans le cadre du cours de Python consiste en la création de deux outils distincts : l’un permettant l’aspiration de textes sur un site Web, l’autre permettant la classification de ces textes en fonction de leur thème. Pour la première partie du projet nous avons choisi de récolter une vingtaine d’articles de la rubrique « Culture » du journal en ligne Le Monde (en ligne : lemonde.fr). Dans la seconde partie du projet, nous avons décidé de les classer en fonction des cinq catégories : Art (sculpture, peinture, photographie etc.), Scène (théâtre, opéra etc.), Musique, Cinéma et Littérature.

1. Les packages utilisés

Pour la partie de crawling, nous avons utilisé la bibliothèque BeautifulSoup. Ce module peut être téléchargé via Pip. C’est ce module qui permet de récupérer le contenu html d’une page et de manipuler les différentes balises pour récupérer les informations nécessaires principales et utiles pour notre tâche. Nous avons également utilisé le module requests afin de faire une requête à partir de l’url de la page que l’on souhaite récupérer. Enfin nous avons utilisé le moule os pour manipuler les dossiers (création des dossiers qui contiendront le corpus de la première partie et les textes classés pour la seconde partie du projet). Ces deux modules sont déjà disponibles avec la version de base de python 3.

Pour la seconde partie, nous avions tout d’abord comme objectif d’utiliser le module Scikit-learn. Nous avons abandonné l’idée car il s’agit d’un module de Machine Learning et il nous aurait donc fallu beaucoup trop de données d’apprentissage annotées pour réaliser l’apprentissage du module ce qui demande beaucoup trop de temps pour mettre en œuvre cette partie par rapport au temps imparti. Nous avons donc décidé de rester sur un modèle utilisant un lexique pour faire une recherche de champ lexical dans les textes tokenisés avec le module Spacy.

1. le découpage en modules, les chaines de traitements mises en œuvres (faites des schémas)
2. les choix algorithmiques et les principales structures de données (fichiers, tableaux, etc.)

Pour la première partie, deux fonctions sont utilisées l’une utilisant requests pour récupérer une page, mettre le contenu de cette page html dans une variable et parser le contenu html à l’aide de BeautifulSoup en faisant une soup. L’autre fonction récupère les liens présents dans les balises d’une page html et enregistre dans un fichier texte le contenu de chaque page dans un dossier « Corpus ».

Pour la deuxième partie, une fonction permet d’ouvrir un fichier, d’en extraire le titre et le reste du fichier. Une autre fonction permet de tokeniser le texte. Nous avons décidé d’utiliser un dictionnaire des champs lexicaux de chaque catégorie. Chaque mot de chaque texte test tokenisé est comparé au dictionnaire. Quand l’un des mots est similaire, il est ajouté à un dictionnaire avec le thème en question avec le nombre de fois où il est présent. Ce mot est pondéré s’il fait partie du titre. Puis une fonction prend en entrée le champ lexical d’un texte et en donne le thème et une liste de pourcentages d’appartenance à chaque thème enregistré dans un fichier et classé dans un dossier correspondant à son thème.

1. une évaluation sommaire des sorties

Pour la seconde partie, nous avons évalué manuellement les résultats des sorties : pour 20 fichiers 2 sont mal catégorisés : text11 appartient à la catégorie arts mais est catégorisé comme musique et text17 catégorisé comme musique appartient à littérature. Ainsi l’évaluation est la suivante : il y a 90% de bonnes réponses en moyenne.

1. les bogues constatés

Nous n’avons pas constaté de bogues lors de l’exécution du projet. Toutes les fonctionnalités marchent comme nous le souhaitions.

1. Les améliorations à apporter et extensions prévues.

Pour améliorer ce projet, il faudrait plus de catégories avec plus de mots dans chacune d’elles. De plus, nous pourrions faire comme nous avions initialement prévu : en utilisant le module Scikit-learn. Avec Scikit-learn, il y a beaucoup de données d’apprentissages annotées et donc les données de test seront beaucoup plus précises. L’évaluation serait donc meilleure.