SCIENCE DES DONNÉES ET FOUILLE DU WEB

RSS-INTELLIGENCE

Enseignant : MARTEAU Pierre-François
Etudiants : ADAM Antoine & JOSSE Samuel

Faculté sciences & sciences de l'ingénieur
Université Bretagne Sud

Table des matières

Résumé

Le projet RSS-Intelligence consiste à récupérer des flux RSS provenant de différentes sources à la fois françaises et anglaises. Ces flux sont ensuite répartis dans six catégories différentes, à savoir :

- Art & Culture
- Finance & Économie
- Politique & Géopolitique
- Santé & Médecine
- Science & Technologie
- Sport

Ces flux sont insérés dans une base de données après avoir été nettoyés (c'est-à-dire après suppression des données indésirables ou inutiles).
Ces données peuvent être consultées grâce à l'outil ElasticSearch, qui permet de visualiser les données de la base dans un format plus approprié.
Ensuite, les données sont analysées pour en extraire un dictionnaire de mots, qui représente l'ensemble des mots présents dans chaque flux. Cette étape est réalisée deux fois, pour créer un dictionnaire par langue.
Enfin, ces dictionnaires sont fournis à un classifieur qui va s'en servir pour prédire la catégorie d'un flux.

Introduction

Ce projet correspond à un travail dans le cadre du cours Science des données et fouille du web. Il s'agit d'un projet collaboratif, de deux personnes, ayant été réalisé sur une durée d'un semestre.
L'objectif de ce projet est de permettre l'apprentissage du scraping de flux RSS, et de faciliter l'usage des différents outils d'apprentissage automatique. La réalisation du projet est guidée, mais les outils utilisés sont pour la plupart sélectionnés par les équipes.
Dans le cadre de ce projet, nous avons manipulé les outils suivants :

- Feedparser
- Langdetect
- BeautifulSoup
- Shell
- ElasticSearch
- SnowballStemmer
- Stopwords
- ScikitLearn
- Pandas
- Numpy

Description des principaux composants

Notre solution de gestion des flux RSS s'appuie sur plusieurs fichiers pour offrir à l'utilisateur le résultat souhaité :

ArticleScraper
Ce fichier permet de récupérer les flux RSS provenant de sources diverses afin de les stocker dans une base de données shelve.
Dans un premier temps, le programme récupère les documents au format XML ou RSS des sites cibles. Les sites choisis sont exclusivement en français ou en anglais. Chacun de ces sites est classifié dans l'une des catégories suivantes : Finance & Économie, Santé & Médecine, Art & Culture, Science & Technologie, Sport, Politique & Géopolitique.
L'information extraite de ces flux est un item, qui correspond à un article. Cet item est segmenté afin d'y récupérer l'information pertinente (titre, date de publication, résumé, etc…).
Lorsque les différentes informations sont isolées, plusieurs traitements sont appliqués. Tout d'abord, un id unique est attribué à chaque item, basé sur son titre, son URL et sa description. Puis, la langue de l'article est extraite à l'aide de la librairie Langdetect.
Enfin, on applique BeautifulSoup sur le contenu afin d'y supprimer des informations n'apportant rien au contenu (comme les balises html par exemple).
La sortie obtenue par ce fichier est stockée dans le dossier items, qui contient trois fichiers : article_db.bak, article_db.dat, article_db.dir.

IndexerSearcher
Ce fichier permet d'exploiter ElasticSearch. Grâce à un système de mots-clés, l'utilisateur saisit un mot, qui est ensuite recherché dans la base shelve créée précédemment.
Le résultat affiche l'ensemble des articles contenant le mot saisi. La sortie obtenue est une liste textuelle des articles correspondants, affichée dans la console.

DictionaryCreator
Ce fichier va interroger la base de données shelve afin de créer des dictionnaires de mots. Il va pour cela analyser l'ensemble des données de la base shelve afin de récupérer tous les mots de vocabulaire présents dans l'article. Ces mots sont issus du titre, du contenu et de la description.
Une fois les mots récupérés, deux filtres sont appliqués pour éliminer les éléments indésirables, comme la ponctuation ou les nombres d'un part, et dans un second temps pour supprimer les mots trop communs, n'apportant aucune plus-value pour différencier les catégories. Ce second traitement est fait à l'aide de la librairie Stopwords.
Ensuite, l'ensemble des mots extraits est passé dans un stemmer, de la librairie Snowball, qui va réduire tous les mots à leur racine, pour obtenir des dictionnaires moins lourds et plus simples pour la comparaison à d'autres textes.
La dernière étape de la constitution des dictionnaires consiste à vectoriser les mots, grâce au module countVectorizer de scikitLearn. Cela permet d'obtenir des métriques pour la manipulation et la comparaison avec d'autres mots lors de l'étape de classification.
Ces mots sont ajoutés à l'un des deux dictionnaires créés (français/anglais). Ces dictionnaires sont le résultat de sortie, présents sous la forme de fichiers joblib dans le dossier dico.

ShelveOpen
Ce fichier permet de manipuler la base de données shelve dans laquelle se trouvent les flux RSS stockés.
Plusieurs manipulations sont possibles, comme la consultation d'un article à partir de son ID, la suppression d'articles ne correspondant pas aux langues désirées par l'utilisateur, et enfin plusieurs fonctions de comptage, pour connaitre le nombre d'articles par catégorie et au total.

Classifiers
Ce fichier créé deux datasets à partir des fichiers joblib (pour obtenir un dataset par langue). Plusieurs classifieurs sont utilisés pour chaque dataset, pour tester l'efficacité de chacun d'entre eux avec différentes métriques et ne conserver que le meilleur classifieur pour le test final.
Les différents classifieurs sont :

- Le KNN
- La régression logistique
- Le bayésien naïf
- Le SVM (Support Vecteur Machine)
- Le réseau neuronal
- Les forêts aléatoires

Pour l'entrainement du modèle, les données ont été réparties, 80% d'entre elles correspondant à l'entrainement et 20% au test. Un cross validation de 5 split a été appliqué sur les données d'entrainement, en utilisant le StratifiedKFold.
Après comparaison de la précision, le dataset français obtient de bien meilleurs résultats en exploitant le classifieur des forêts aléatoires, avec une précision de 73%.
Pour le dataset anglais, les résultats sont plus indécis, avec une précision de 73% pour le réseau neuronal et de 72% pour les forêts aléatoires. Cependant, le classifieur des forêts aléatoires a été choisi également pour ce dataset en raison du temps bien inférieur lors de l'entrainement.
D'autres métriques ont également été étudiées, à savoir la précision micro et macro, le recall micro et macro, et l'AUC micro et macro.

Flux d’information entre les différents composants

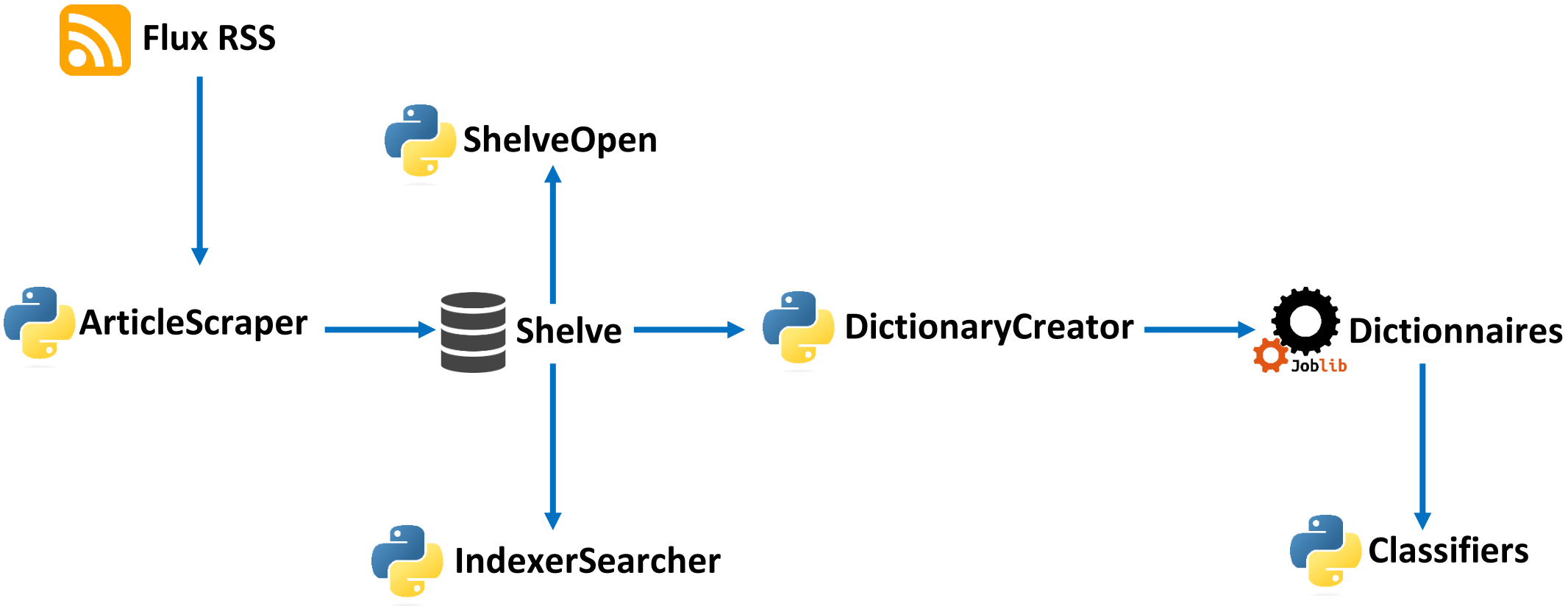
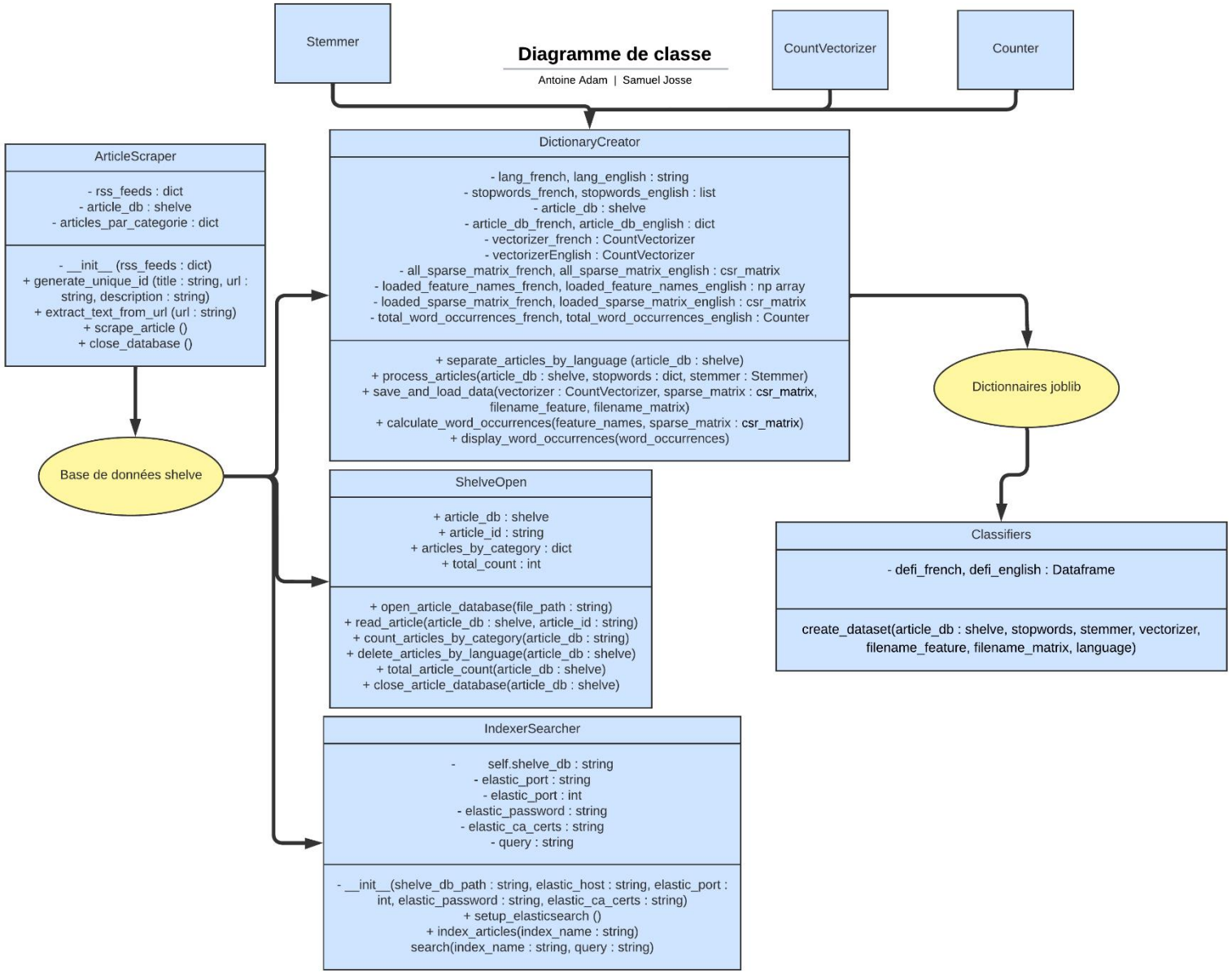


Diagramme de conception



Limites et améliorations à prévoir

Le projet fournit des résultats avec les données récupérées à travers les flux RSS des sources choisies, et parvient à classer avec une certaine efficacité les flux dans les catégories correspondantes. Malgré cela, plusieurs problèmes subsistent :

- Les fichiers joblib générés par le DictionaryCreator ne parviennent pas à être ouverts directement par le Classifiers. Ils sont donc recréés de manière identique directement dans le fichier Classifiers, ce qui prend un temps conséquent, notamment pour le benchmark.

Plusieurs améliorations sont également possibles pour faire évoluer le projet :

- Aucune interface graphique n'est disponible pour visualiser de manière plus ergonomique les données, cela ajouterait une plus-value conséquente pour la lecture des flux sélectionnés par l'utilisateur
- Le chatbot n'a pas été développé, mais permettrait une approche plus intéressante pour la sélection des flux cohérents avec les demandes utilisateur.
- Plusieurs langues pourraient être ajoutées sans que cela ne soit complexe à mettre en place, et permettrait de viser un public plus large.

Conclusion

Les consignes principales du projet ont été respectées, les prédictions du classifier offrent un résultat satisfaisant, qui pourrait bien entendu être amélioré (utilisation de gridSearchCV pour trouver les meilleurs paramètres du modèle par exemple).
Le travail au sein de l'équipe s'est déroulé sans accroc, la répartition s'est faite naturellement au fur et à mesure de l'avancée du projet.
Le projet en lui-même était intéressant, nous permettant d'acquérir davantage de compétences sur divers sujets. Le choix du binôme est un vrai plus, cela permet de travailler plus sereinement en connaissant les capacités de chacun.