Projet L3
Ingénierie des langues
Fouille de données

GOEHRY Martial
16711476

2 octobre 2022

Table des matières

1 Introduction

Ce projet à pour but de permettre de détecter les mails comme étant spam ou ham. La définition d'un spam dans le dictionnaire *Larousse* est :

> "Courrier électronique non sollicité envoyé en grand nombre à des boîtes aux lettres électroniques ou à des forums, dans un but publicitaire ou commercial."

Il est possible d'ajouter à cette catégorie tous les mails indésirables comme les tentatives d'hameçonnage permettant de soutirer des informations personnelles à une cible.

L'objectif est de travailler uniquement sur les données textuelles issues du corps du mail. Nous avons donc en point de départ les éléments suivants :

- langue : anglais
- corpus : monolingue écrit
- type : e-mail

Le schéma ci-dessous donne une vue synthétique des étapes du projet :

2 Récolte des données

2.1 Recherche de dataset

2.2 Téléchargement des données

3 Pré-traitement

3.1 Extraction des corps des mails

3.2 Nettoyage

Par regex

Par module

3.3 Mise en base

Stockage des données : ElasticSearch

Stockage des données statistiques du traitement : SQLite

3.4 Recherche de caractéristiques

Références

3.5 Analyse préliminaire

A Développement visualisation distribution de Zipf

Présentation La loi de distribution de Zipf est une loi empirique (basée sur l'observation) qui veut que le mot le plus fréquent est, à peu de chose près, 2 fois plus fréquent que le 2^{eme}, 3 fois plus fréquent que le 3^{eme} etc.

La formulation finale de la 1^{ere} loi de Zipf est la suivante :

$$|mot| = constante \times rang(mot)^{k \approx 1}$$

avec $|mot|$ la fréquence d'apparition d'un mot, *constante* une valeur propre à chaque texte, *rang(mot)* la place du mot dans le tri décroissant par fréquence d'apparition et k un coefficient proche de 1.

Développement Afin de pouvoir utiliser les résultats de cette distribution dans mon analyse, j'ai développé un ensemble de fonctions sur un corpus "*reconnu*". Mon choix s'est porté sur le corpus *Brown* présent de la libraire *nltk*. Ce corpus contient environ 500 documents contenant 1 millions de mot en anglais.

Le processus d'analyse se fait sur 2 versions de ce corpus.
— la première version contient tous les mots sans modifications
— le seconde version contient tous les mots sans les *stopwords*
Les *stopwords* sont des mots qui n'ont pas ou peu de signification dans un texte. Ces mots sont retirés dans la 2^{e} version pour voir l'effet d'une réduction sur la distribution de Zipf.

Les paragraphes ci-dessous détaillent les étapes du développement :

Étape 1 - calcul de fréquence La première étape est de compter les occurrences de tous les mots des 2 corpus.

```
1  def frequence_mot(bag, freq=None):
2      """
3      Calcule la frequence de chaque mot dans un sac de mot
4      :param bag: <list> - liste de tous les mots d'un texte
5      :param freq: <dict> - dictionnaire avec {<str> mot: <int> frequence}
6      :return: <dict> - dictionnaire avec la frequence par mot {mot:
       frequence}
7      """
8      if freq is None:
9          freq = {}
10     for mot in bag:
11         freq[mot] = freq.get(mot, 0) + 1
12     return freq
```

Étape 2 - classement La deuxième étape a pour objectif de classer les mots en fonction de leur fréquence

```
1  def classement_zipf(dico):
2      """
3      Trie un dictionnaire de mots : occurence et leur assigne un rang en
       fonction du nombre d'occurence
```

```
    :param dico: <dict> dictionnaire de mot: occurences
    :return: <list> {"rang": <int>, "mot": <str>, "frequence": <int>}
    """
    ranked = []
    for rang, couple in enumerate(sorted(dico.items(), key=lambda item:
item[1], reverse=True), start=1):
        ranked.append({"rang": rang,
                       "mot": couple[0],
                       "frequence": couple[1]})

    return ranked

```

Etape 3 - choix de la constante Le premier paramètre a déterminer est la *constante*. Pour ce faire j'effectue le calcul suivant pour tous les mots :

$$constante = |mot| \times rang(mot)$$

On obtient une liste de toutes les constantes théoriques pour chaque mot selon son rang. De cette liste, nous allons extraire la moyenne et la médiane.

```

```

Etape 4 - recherche du coefficiant

```

```

Résultats

B Tableau des choix technologiques

Élément	Retenu	Raisons	Observations
Datasets			
Mail de la compagnie Enron	Non	Mails non classés	Non retenu pour la phase de développement car pas de moyen fiable de contrôler la sortie automatiquement
Mail du projet SpamAssassin	Oui	Mails déjà pré-triés	Mails principalement en Anglais déjà pré-trié en catégorie Spam et Ham
Brown dataset (nltk)	Oui	Corpus d'un million de mots en Anglais publié en 1961	Dataset utilisé pour le développement de la visualisation de la distribution de Zipf
Stopwords (nltk)	Oui	Corpus de mots commun non significatif dans un texte	Utilisation dans le développement de la visualisation de la distribution de Zipf
Langage et Modules			
Python	Oui	Langage polyvalent pour le traitement des données	
Module email	Oui	Module natif pour le traitement des mails	Grande flexibilité pour la lecture des mails
Bases de données			
ElasticSearch	Oui	Technologie utilisée dans mon entreprise. Présence d'une interface de visualisation des données Kibana.	Application dockerisée.
SQLite	Oui	Base de données légère pour stocker uniquement les données statistiques des étapes	Rapide à mettre en place et déjà intégrée

C Bibliographie

D Sitotec