Université de Lille

PJE

ANALYSE DE SENTIMENTS SUR TWITTER

Présenté par Corentin Duvivier et Florentin Bugnon

Plan

Description du projet et de la problématique

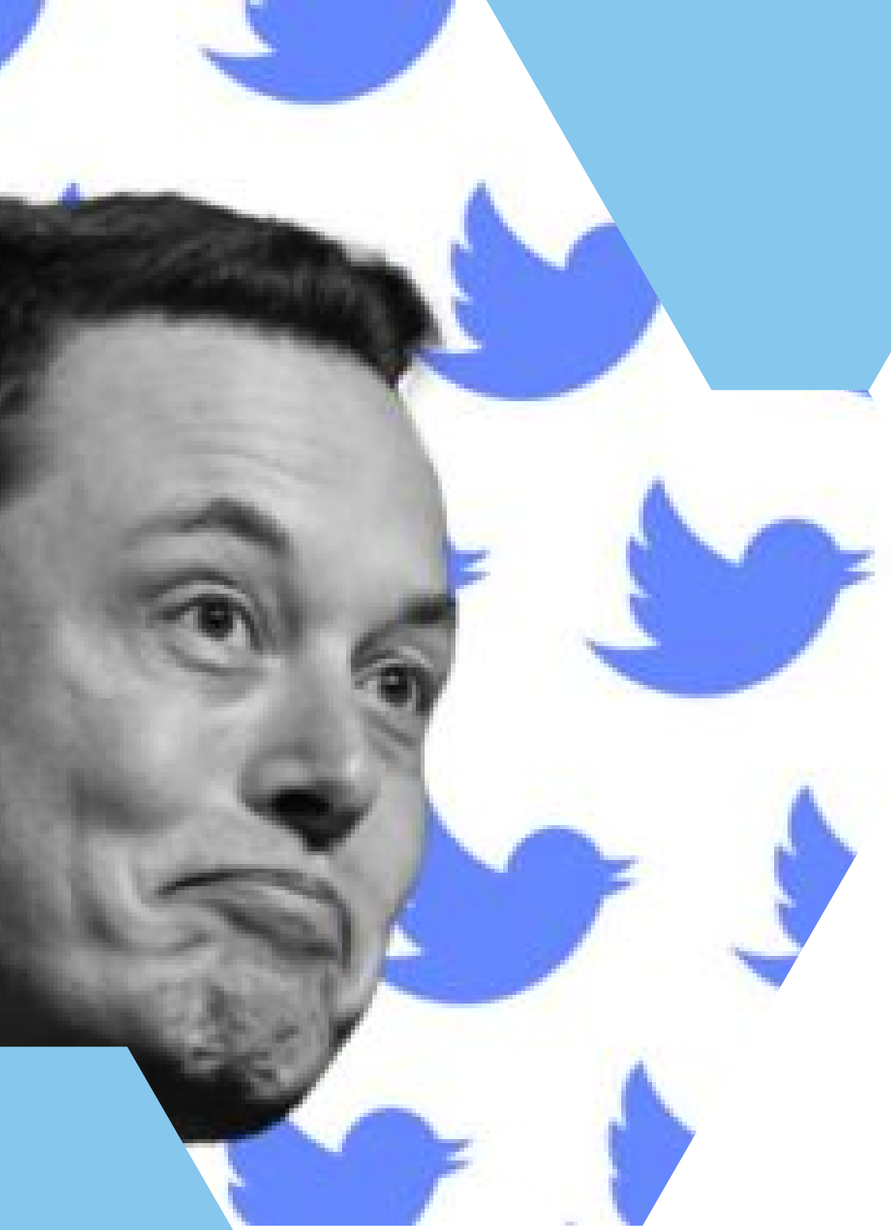
API Twitter

Nettoyage des tweets

Algorithmes de classification

Interface utilisateur

Analyse et résultats



DESCRIPTION DU PROJET

? Quelle est l'opinion générale exprimée par les utilisateurs de Twitter, sur un sujet précis?

Twitter

API TWITTER

Twitter fournit une API à n'importe qui souhaitant récupérer/manipuler des tweets, des utilisateurs et tout ce que contient l'application.

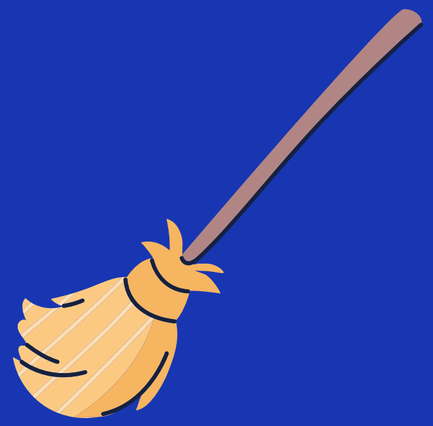
PORTAIL DE DEVELOPPEURS

Pour pouvoir utiliser l'API Twitter, il est nécessaire de remplir un formulaire pour avoir un compte "développeur" et obtenir ses tokens d'accès.

RECUPERATION DE TWEETS

On récupère un certains nombres de tweets en fonctions de mots clés et de la langue.

NETTOYAGE DES TWEETS



01 **Nettoyage des @username**

"@[a-zA-Z0-9_]+" -> ""

02 **Nettoyage des URL**

r'((http|https)\:\/\/)?[a-zA-Z0-9\.\/\?\:@\-_=#]+\.([a-zA-Z]){2,6}([a-zA-Z0-9\.\&\/\?\:@\-_=#])*' -> "url"

03 **Nettoyage des ReTweets**

"RT @[a-zA-Z0-9_]+:" -> ""

04 **Nettoyage des #hashtag**

"#[a-zA-Z0-9_]+" -> ""

Algorithmes de classification

Naïf

A partir de 2 fichiers contenants respectivement des mots jugés négatifs et positifs, on détermine le sentiment général du tweet en fonction du nombre de mots présents dans chaque fichier.

Bayes

A partir d'une base de tweets existants, on va effectuer une étude probabiliste pour déterminer l'annotation du tweet sur lequel on travaille.

Bayes V2

A partir d'une base de tweets existants, on va effectuer une étude probabiliste, en prenant en compte la fréquence des mots, et la présence de bi-grammes pour déterminer l'annotation du tweet sur lequel on travaille.

KNN

A partir d'une base de tweets déjà annotés, on va annoter le tweet en fonction des annotations des tweets lui ressemblant le plus, présents dans la base de données.

Classification naïve

Paramètres : les tweets à classifier, une liste de mots négatifs, une liste de mots positifs

Pour chaque Tweet :

- **Pour chaque mot du Tweet :**
  - **On compte le nombre de mots positifs/négatifs présents dans le Tweet**
  - **S'il y a plus de mots positifs que négatifs :**
    - **On annote le tweet comme positif**
  - **S'il y a plus de mots négatifs que positifs :**
    - **On annote le tweet comme négatif**
  - **Sinon, on annote le tweet comme neutre**

Classification KNN

<u>Paramètres</u> : les tweets à classifier, une base de données contenant des tweets déjà annotés, une distance noté d (int)

Pour chaque tweet :

- **On cherche un nombre d de tweets les plus ressemblants dans la base de données (Voisinage)**
- **Si la proportion de tweets positifs est la plus élevée:**
  - **On annote le tweet comme positif**
- **Si la proportion de tweets négatifs est la plus élevée:**
  - **On annote le tweet comme négatif**
- **Sinon :**
  - **On annote le tweet comme neutre**

Classification Bayesienne V1

Paramètres : les tweets à classifier, une base de données contenant des tweets déjà annotés

Pour chaque tweet :

- **On calcule la probabilité d'appartenance à une classe (en remplaçant neutral par positif et négatif) selon la formule suivante:**

$$P(C|T) = \prod_{M \in T} P(m|C) \bullet P(C)$$

où C est la classe, T un tweet, et M un mot du tweet

- **Si la probabilité que le Tweet soit neutre est la plus élevée:**
  - **On annote le tweet comme neutre**
- **Si la probabilité que le Tweet soit positif est la plus élevée:**
  - **On annote le tweet comme positif**
- **Sinon :**
  - **On annote le tweet comme négatif**

Classification Bayesienne V2

<u>Paramètres</u> : les tweets à classifier, une base de données contenant des tweets déjà annotés

Pour chaque tweet :

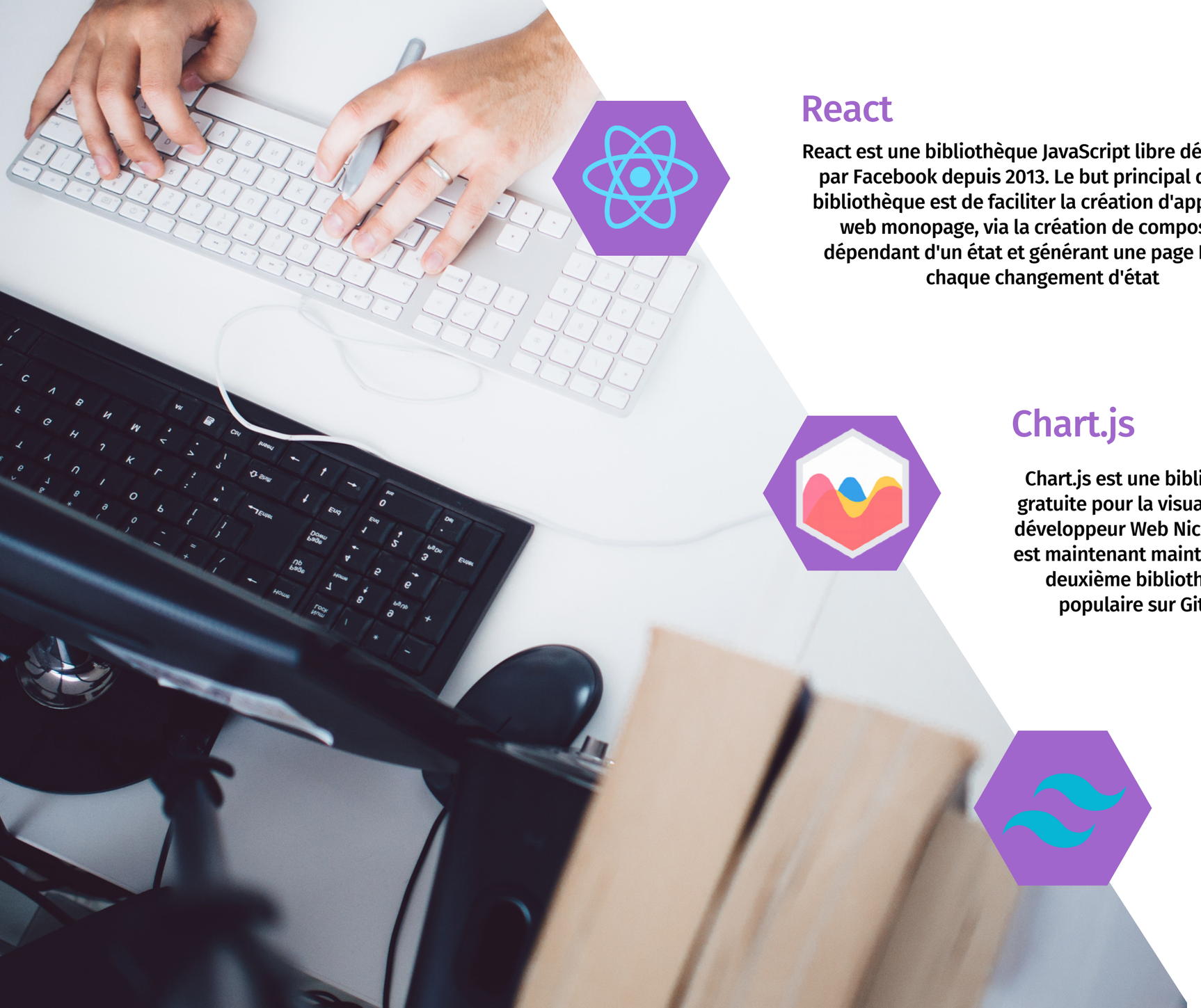
- **On calcule la probabilité d'appartenance à une classe (en remplaçant neutral par positif et négatif) selon la formule suivante:**

$$P(C|T) = \prod_{M \in T} P(m|C)^{n_m} \bullet P(C)$$ où C est la classe, T un tweet, m un mot du tweet, n le nombre de fois où m est présent

- **Si la probabilité que le Tweet soit neutre est la plus élevée:**
  - **On annote le tweet comme neutre**
- **Si la probabilité que le Tweet soit positif est la plus élevée:**
  - **On annote le tweet comme positif**
- **Sinon :**
  - **On annote le tweet comme négatif**

Améliorations de la classification Bayesienne V2

- *Prise en compte de la fréquence d'un mot dans un tweet*
- *Ignorance des mots d'une taille inférieures ou égaux à 3*
- *Prise en compte de bi-grammes, c'est-à-dire toute suite de 2 mots.*



React

React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état

Chart.js

Chart.js est une bibliothèque JavaScript open source gratuite pour la visualisation de données. Créée par le développeur Web Nick Downie en 2013, la bibliothèque est maintenant maintenue par la communauté et est la deuxième bibliothèque de graphiques JS la plus populaire sur GitHub par le nombre d'étoiles

Tailwind CSS

Tailwind CSS est un framework CSS open source. La principale caractéristique de cette bibliothèque est que, contrairement à d'autres frameworks CSS comme Bootstrap, elle ne fournit pas une série de classes prédéfinies pour des éléments tels que des boutons ou des tableaux

Analyses expérimentales

Taux d'erreur

Analyse présence + unigramme

0.1888

Analyse présence + bigramme

0.0955

Analyse présence + unigramme + bigramme

0.0311

Analyse fréquence + unigramme

0.02111

Analyse fréquence + bigramme

0.02333

Analyse fréquence + unigramme + bigramme

0.02111

Université de Lille

Merci !

N'hésitez pas si vous avez des questions.

Corentin Duvivier
Florentin Bugnon