Université Mohammed V

École Nationale Supérieure d'Informatique et d'Analyse des Systèmes

Master Bioinformatique et Modélisation des Systèmes complexes liés à la Santé

Réalisé par :

Houda Akoumi et Souad Atigi

Partie 3: Analyse et extraction des tweets

Encadré par :

Pr. Leila Benhlima

Année universitaire 2021-2022

Table des matières

Table des figures

Chapitre 1

Scraping Twitter avec Python

Twitter est un réseau social en ligne populaire où les utilisateurs peuvent envoyer et lire de courts messages appelés « tweets». C'est aussi un instrument pour mesurer les événements sociaux, et chaque jour des millions de personnes tweetent pour exprimer leurs opinions sur tous les sujets imaginables. Cette source de données est précieuse tant pour la recherche que pour les entreprises.

Dans ce chapitre, nous expliquerons les outils utilisés pour récupérer des tweets en direct sur Twitter.

1.1 tweepy

Tweepy est l'une des bibliothèques python les plus populaires pour configurer l'accès avec Twitter. Tweepy prend en charge l'accès à Twitter via l'authentification de base et la nouvelle méthode, OAuth. Twitter a cessé d'accepter l'authentification de base, OAuth est donc désormais le seul moyen d'utiliser l'API Twitter. Cette API vous permet de rechercher et de récupérer, d'interagir avec ou de créer une variété de ressources différentes, notamment les tweets, les utilisateurs, les tendances et média.[1]

Tweepy est un excellent outil pour une automatisation simple, la création de robots Twitter ou un petit projet scolaire. Cependant, Tweepy a une limite de grattage de 3200 tweets et le temps le plus long que vous pouvez parcourir en une semaine. Il n'y a pas d'accès aux données historiques.[2]

FIGURE 1.1

1.2 snscrape

snscrape (sorti le 8 juillet 2020) est un scraper pour les services de réseaux sociaux (SNS). Il récupère des éléments tels que les profils d'utilisateurs, les hashtags ou les recherches et renvoie les éléments découverts, par exemple les publications pertinentes. Il ne sert pas seulement à gratter des tweets, mais également à travers divers autres sites de réseaux sociaux comme Facebook, Instagram, Reddit, VKontakte et Weibo (Sina Weibo).

snscrape permet d'extraire des tweets par l'intermédiaire de l'API de Twitter sans aucune restriction ni limite de requête. En outre, nous n'avons même pas besoin d'un compte de développeur Twitter pour récupérer des tweets.[3]

1.3 twint

Twint est un outil de scraping Twitter avancé écrit en Python qui permet de scraper les Tweets des profils Twitter sans utiliser l'API de Twitter.

Twint utilise les opérateurs de recherche de Twitter pour vous permettre de récupérer les Tweets d'utilisateurs spécifiques, de récupérer les Tweets relatifs à certains sujets, hashtags et tendances, ou de trier les informations sensibles des Tweets comme les e-mails et les numéros de téléphone. Je trouve cela très utile, et vous pouvez aussi devenir très créatif avec.

Twint effectue également des requêtes spéciales sur Twitter, vous permettant également de récupérer les abonnés d'un utilisateur Twitter, les Tweets qu'un utilisateur a aimés et ceux qu'il suit sans aucune authentification, API, Selenium ou émulation de navigateur.[4]

FIGURE 1.2

1.4 GetOldTweets3

L'API officielle de Twitter a la limitation gênante des contraintes de temps, vous ne pouvez pas obtenir de tweets plus anciens qu'une semaine. L'idée de GetOldTweets3 est de donner l'accès à des tweets plus anciens sans dépenser de l'argent. Mais, Twitter a supprimé le point de terminaison utilisé par GetOldTweets3, ce qui rend GOT inutile. Vous trouverez de nombreux projets sur GitHub qui utilisaient GetIOldTweets, mais depuis le 27 novembre 2019, aucune mise à jour du package n'a été apportée pour respecter les directives révisées de Twitter.[5]

Chapitre 2

Spark Streaming

Apache Spark Streaming est un système de traitement de flux évolutif et tolérant aux pannes qui prend en charge de manière native les charges de travail par lots et par flux. Spark Streaming est une extension de l'API Spark principale qui permet aux ingénieurs de données et aux data scientists de traiter des données en temps réel provenant de diverses sources, notamment (mais sans s'y limiter) Kafka, Flume et Amazon Kinesis.

Ces données traitées peuvent être transférées vers des systèmes de fichiers, des bases de données et des tableaux de bord en direct.

Son abstraction clé est un Discretized Stream ou, en bref, un DStream, qui représente un flux de données divisé en petits lots. Les DStreams sont construits sur les RDD, l'abstraction de données de base de Spark. Cela permet à Spark Streaming de s'intégrer de manière transparente à tous les autres composants Spark tels que MLlib et Spark SQL.

Spark Streaming est différent des autres systèmes qui ont soit un moteur de traitement conçu uniquement pour le streaming, ou ont des API de traitement par lots et de diffusion similaires, mais compilent en interne sur différents moteurs. Le moteur d'exécution unique de Spark et son modèle de programmation unifié pour le traitement par lots et le streaming offrent des avantages uniques par rapport aux autres systèmes de streaming traditionnels.

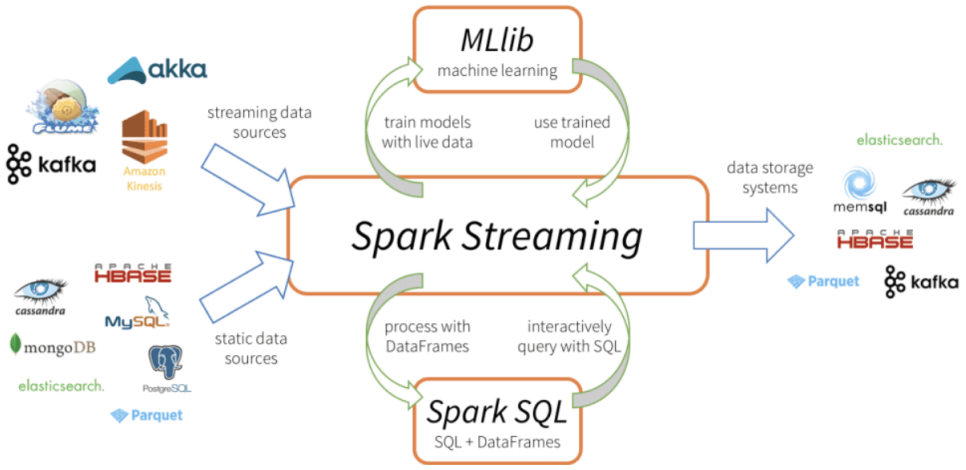

FIGURE 2.1 – Architecture de Spark Streaming

Cette unification de capacités de traitement de données est la principale raison de l'adoption rapide de Spark Streaming. Il permet aux développeurs d'utiliser très facilement un cadre unique pour satisfaire tous leurs besoins de traitement.[6, 7]

Chapitre 3

Analyse de sentiments

L'analyse des sentiments, également appelée opinion mining, est l'un des domaines de recherche les plus actifs dans le traitement du langage naturel depuis le début des années 2000. L'objectif de l'analyse des sentiments est de définir des outils automatiques capables d'extraire des informations subjectives de textes en langage naturel, telles que des opinions et des sentiments, afin de créer des connaissances structurées et exploitables par un système d'aide à la prise de décision.

Une opinion est un quintuplet :

$$\left(e_i, a_{ij}, s_{ijkl}, h_k, t_l\right),$$

où e_i est le nom d'une entité, a_{ij} est un aspect de e_i, s_{ijkl} est le sentiment sur l'aspect a_{ij} de l'entité e_i, h_k désigne le porteur de l'opinion, et t_l est le moment où l'opinion est exprimée par h_k.

Le sentiment s_{ijkl} est positif, négatif ou neutre, ou exprimé avec différents niveaux de force/intensité, comme le système de 1 à 5 étoiles utilisé par la plupart des sites d'évaluation (par exemple, Amazon).

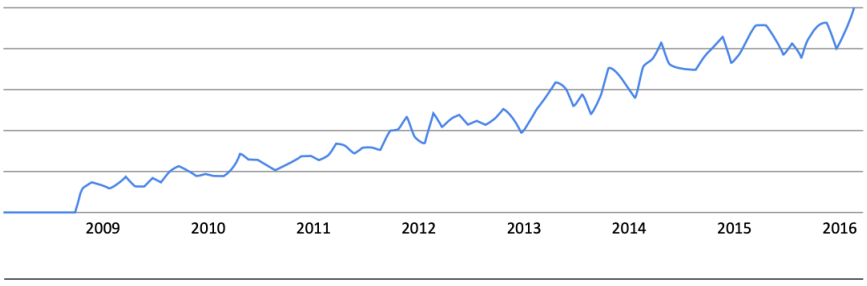

FIGURE 3.1 – Google Trends data liées au mot clé : analyse des sentiments

Les principales caractéristiques qui constituent l'analyse des sentiments sont :

1. Catégorisation des sentiments : phrases objectives et subjectives

Le premier objectif lorsque l'on traite de l'analyse des sentiments consiste généralement à distinguer les phrases subjectives des phrases objectives. Si une phrase donnée est classée comme objective, aucune autre tâche fondamentale n'est requise, alors que si la phrase est classée comme subjective, sa polarité (positive, négative ou neutre) doit être estimée. La classification de la subjectivité est la tâche qui permet de distinguer les phrases qui expriment des informations objectives (ou factuelles) et les phrases qui expriment des points de vue et des opinions subjectifs (phrases subjectives).

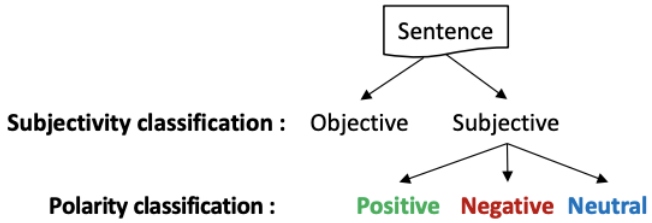

FIGURE 3.2 – Workflow pour l'analyse de sentiments

2. Les niveaux d'analyse En général, l'analyse des sentiments dans les réseaux sociaux peut être étudiée principalement à trois niveaux :

— **Niveau du message :** l'objectif est de classer la polarité de l'ensemble d'un message d'opinion.
— **Niveau de la phrase :** l'objectif est de déterminer la polarité de chaque phrase contenue dans un message texte.
— **Niveau entité et aspect :** effectue une analyse plus fine que le niveau du message et de la phrase. Il repose sur l'idée qu'une opinion se compose d'un sentiment et d'une cible (d'opinion).

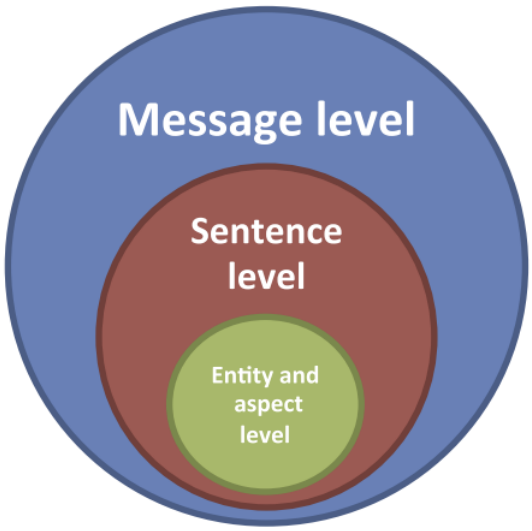

FIGURE 3.3 – Les différents niveaux d'analyse

3. Le rôle de la sémantique La sémantique du langage utilisé dans les réseaux sociaux est fondamentale pour analyser avec précision les expressions des utilisateurs. Le contexte d'une expression textuelle est donc un élément crucial qui doit être pris en compte pour traiter correctement le sentiment sous-jacent. Une phrase "prise telle quelle" peut apparaître comme négative ou positive, mais si elle est correctement analysée d'un point de vue sémantique, elle peut être complètement différente. Par exemple, les phrases "J'ai regardé le plus terrifiant des films d'horreur. C'était comme un vrai cauchemar ! PAAAANIIIICCC"" peut être initialement interprétée comme négative, mais en tenant compte du contexte dans lequel ce type d'opinions est exprimé (c'est-à-dire une communauté d'amateurs de films d'horreur) et de certains indices

lexicaux typiques du langage des réseaux sociaux, nous devrions en déduire un jugement (réellement) positif. Les lexiques, les corpus et les ontologies doivent être construits et utilisés correctement pour nous permettre d'avoir une compréhension approfondie de la sémantique du langage naturel dans les réseaux sociaux en ligne.

4. Opinions explicites et implicites

— **L'opinion explicite :** une opinion explicite est une déclaration subjective qui donne une opinion régulière ou comparative.

— **Opinion implicite :** une opinion implicite est une déclaration objective qui implique une opinion régulière ou comparative et qui exprime généralement un fait souhaitable ou indésirable. [8]

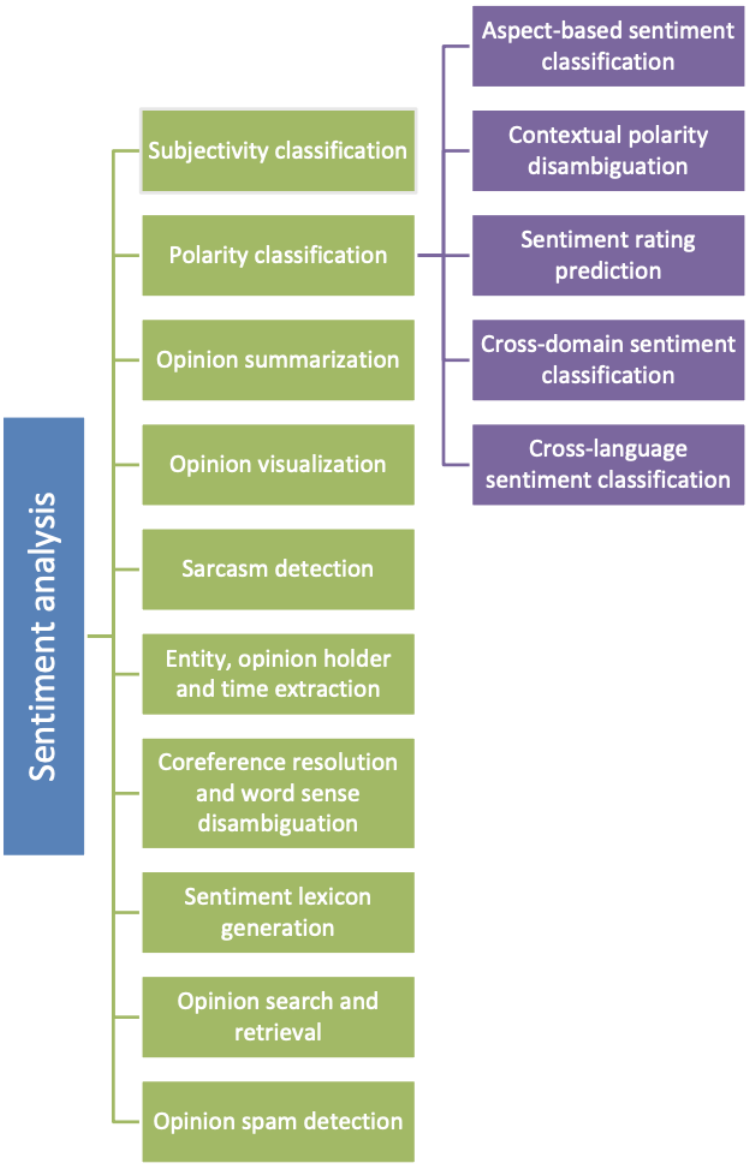

FIGURE 3.4 – Tâches d'analyse des sentiments

Chapitre 4

Explication et résultat de code

Ce travail vise à identifier les conversations autour du vaccin COVID-19 dans le monde, les sentiments des gens et les sujets de conversation les plus populaires en utilisant les tweets parlant des vaccins COVID-19.

FIGURE 4.1 – Architecture de notre travail

4.1 Importation des tweets

Avant de commencer, nous devons extraire des données de Twitter avec snscrape. Nous avons utilisé TwitterSearchScraper pour trouver des tweets avec le hashtag (covid vaccine), pendant une durée de 1 mois (de 31-10-2021 à 30-11-2021).

	Datetime	Hashtags	Tweet Id	Text	Username	retweetCount
0	2021-11-29 23:59:59+00:00	[Fauci]	1465470629365403657	What #Fauci has done by financing the creation...	blackhawkinc	5
1	2021-11-29 23:59:59+00:00	None	1465470628954140673	Covid vaccine mandate in Queensland: Annastaci...	dailymail241	0
2	2021-11-29 23:59:59+00:00	None	1465470627603787778	Natural Immunity Works Better Than COVID Vacci...	wrestlerkw7	2
3	2021-11-29 23:59:58+00:00	None	1465470625837993992	Covid vaccine mandate in Queensland: Annastaci...	usmail24	0
4	2021-11-29 23:59:53+00:00	None	1465470602333106185	@bitcharoo96 Yesssssss… I've got a little cockt...	RominaReilly	0

FIGURE 4.2 – Exploration des données avec head()

4.2 Prétraitement des tweets

Cette étape consiste à supprimer les ponctuations, toutes les mentions (@ nom d'utilisateur), les stopwords (mots comme "the", "and", etc.). Ces paramètres ont très peu d'importance car ils ne donnent aucune indication sur les sentiments et leur suppression nous permet d'économiser de la mémoire, de la puissance de calcul et parfois d'augmenter la précision du modèle.

4.2.1 tweet-preprocessor

Preprocessor est une bibliothèque de prétraitement des données de tweet écrite en Python. Elle facilite le nettoyage, l'analyse ou la tokenisation des tweets[9]. La fonction "clean" de preprocessor permet d'éliminer les hashtags, les mentions, les liens, les emojis, ..etc

On a comme résultat :

```
Out[79]: 0        What has done by financing the creation of the...
1        Covid vaccine mandate in Queensland: Annastaci...
2        Natural Immunity Works Better Than COVID Vacci...
3        Covid vaccine mandate in Queensland: Annastaci...
4        Yesssssss Ive got a little cocktail of covid va...
                              ...
19996    Covid vaccine is not safeMy friend had gotten ...
19997    That's true. But you may want to ask for clari...
19998    This is alleged reduction in C19 deaths not al...
19999    COVID-19 DEATHSTRUMP: JAN , -JAN , -- deadBIDE...
20000    Ironically MMA is infested with chuds now also...
Name: Text, Length: 20001, dtype: object
```

FIGURE 4.3 – Résultat de tweet-preprocessor

4.2.2 Les méthodes de string

On va réaliser trois tâches avec string :

— remplacer les majuscules par les minuscules en utilisant la fonction string.lower().
— éliminer les nombres dans le texte, et les remplacer par un espace avec la fonction string.replace()
— remplacer la ponctuation par un espace, avec la fonction string.maketrans()[10]

```
Out[87]: 0        what ha done by financing the creation of the ...
1        covid vaccine mandate in queensland  annastaci...
2        natural immunity work better than covid vaccin...
3        covid vaccine mandate in queensland  annastaci...
4        yes ive got a little cocktail of covid vaccine...
                              ...
19996    covid vaccine is not safemy friend had gotten ...
19997    thats true  but you may want to ask for clarif...
19998    this is alleged reduction in c19 death not all...
19999    covid  19 deathstrump  jan   jan    deadbiden ...
20000    ironically mma is infested with chuds now also...
Name: Text, Length: 20001, dtype: object
```

FIGURE 4.4 – Résulat des méthodes string

4.2.3 NLTK

NLTK est une plate-forme leader pour la création de programmes Python pour travailler avec des données de langage humain. Il fournit des interfaces faciles à utiliser à plus de 50 corpus et ressources lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, la tokenisation, la radicalisation, le balisage, l'analyse et le raisonnement sémantique, des wrappers pour les bibliothèques NLP de puissance industrielle, et un forum de discussion actif[11].

Nous avons effectué deux tâches de NLTK : la lemmatisation et élimination des stopwords.

— **La lemmatisation :** consiste à réduire un mot dans sa forme « racine » en utilisant WordNetLemmatizer de NLTK.

FIGURE 4.5 – NLTK

— **Elimination des stopwords :** éliminer les mots les fréquents dans la langue (nous avons spécifié le français et l'anglais) en utilisant nltk.corpus.stopwords de NLTK.

```
Out[85]: 0        what ha done by financing the creation of the ...
1        covid vaccine mandate in queensland : annastac...
2        natural immunity work better than covid vaccin...
3        covid vaccine mandate in queensland : annastac...
4        yes ive got a little cocktail of covid vaccine...
                              ...
19996    covid vaccine is not safemy friend had gotten ...
19997    that's true . but you may want to ask for clar...
19998    this is alleged reduction in c19 death not all...
19999    covid - 19 deathstrump : jan , - jan , - - dea...
20000    ironically mma is infested with chuds now also...
Name: Text, Length: 20001, dtype: object
```

FIGURE 4.6 – Résultat de traitement NLTK

4.3 Analyse de sentiments

TextBlob est une bibliothèque python pour le traitement du langage naturel (NLP). TextBlob utilise activement Natural Language ToolKit (NLTK) pour réaliser ses tâches. TextBlob renvoie deux propriétés : polarité et subjectivité.

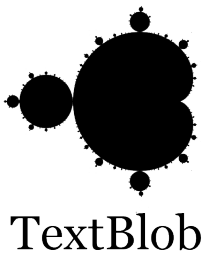

FIGURE 4.7 – TextBlob

La polarité est flottante qui se situe dans la plage de [-1,1] où 1 signifie une déclaration positive et -1 signifie une déclaration négative. Les phrases subjectives font

généralement référence à une opinion personnelle, à une émotion ou à un jugement. La subjectivité quantifie la quantité d'opinions personnelles et d'informations factuelles contenues dans le texte. Elle est aussi un flotteur qui se situe dans la plage de [0,1].

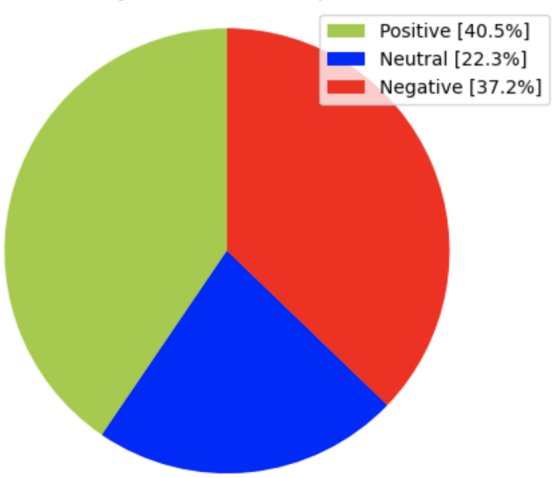

FIGURE 4.8 – Résultat de l'analyse de sentiment pour la vaccination

Nous ajoutons au dataframe les colonnes polarity, subjectivity et sentiment.

	Datetime	Hashtags	Tweet Id	Text	Username	retweetCount	polarity	subjectivity	sentiment
0	2021-11-29 23:59:59+00:00	[Fauci]	1465470629365403657	what ha done by financing the creation of the ...	blackhawkinc	5	0.100000	0.250000	positive
1	2021-11-29 23:59:59+00:00	None	1465470628954140673	covid vaccine mandate in queensland annastaci...	dailymail241	0	0.000000	0.000000	negative
2	2021-11-29 23:59:59+00:00	None	1465470627603787778	natural immunity work better than covid vaccin...	wrestlerkw7	2	0.245455	0.451515	positive
3	2021-11-29 23:59:58+00:00	None	1465470625837993992	covid vaccine mandate in queensland annastaci...	usmail24	0	0.000000	0.000000	negative
4	2021-11-29 23:59:53+00:00	None	1465470602333106185	yes ive got a little cocktail of covid vaccine...	RominaReilly	0	-0.187500	0.500000	positive
5	2021-11-29 23:59:51+00:00	[COVID, CORONAVIRUS]	1465470592606425089	and our child why dont revive covids vaccine ...	CelioRamires	0	0.100000	0.200000	negative
6	2021-11-29 23:59:46+00:00	None	1465470573589385220	exclusive forced to get vaccine to remain lun...	hlwegley	0	-0.366667	0.333333	negative
7	2021-11-29 23:59:44+00:00	None	1465470566580854784	in conversation with the surgeon general the ...	AMWADoctors	3	0.275000	0.700000	neutral
8	2021-11-29 23:59:41+00:00	None	1465470551707688960	obviously im hoping for better vaccine the qu...	brettbutlerisok	0	0.375000	0.472500	positive
9	2021-11-29 23:59:38+00:00	None	1465470541142405120	much more than a dozen anecdotal case do your...	adequatechris	0	0.250000	0.625000	positive

FIGURE 4.9 – Le nouveau dataframe

Pour visuliser les tweets, nous avons utilisé wordcloud qui représente les mots les plus fréquents de notre data :

FIGURE 4.10 – Représentation du data avec wordcloud

4.4 File SparkStreaming

4.4.1 Importation de fichiers

Le streaming structuré est construit sur le moteur SparkSQL d'Apache Spark qui s'occupera de l'exécution du flux au fur et à mesure que les données continuent de recevoir. Tout comme les autres moteurs de Spark, il est évolutif et tolérant aux pannes. Le streaming structuré améliore les API Spark DataFrame avec des fonctionnalités de streaming.

La première tâche à faire est diviser notre dataframe en plusieurs fichiers(4) et les sauvegarder, afin de faire le file sparkStreaming.

Nous allons d'abord importer les bibliothèques Pyspark requises à partir de Python et démarrer une SparkSession :

```
# Diviser notre dataframe en 4, avec preseque 5000 tweets dans chacun.
df1 = df.iloc[:5000]
df2 = df.iloc[5000:10000]
df3 = df.iloc[10000:15000]
df4= df.iloc[15000:]
from pyspark.sql import SparkSession
#Créer Session PySpark
spark = SparkSession.builder \
    .master("local[4]") \
    .appName("Big Data") \
    .getOrCreate()
#Create PySpark DataFrame from Pandas
df1=spark.createDataFrame(df1)
df2=spark.createDataFrame(df2)
df3=spark.createDataFrame(df3)
df4=spark.createDataFrame(df4)
```

FIGURE 4.11 – SparkSession

Nous précisons le schéma des données : ('Text string, polarity double, subjectivity double, sentiment string')

Nous devons spécifier un "format ()" pour le streaming vers une destination et "outputMode ()" pour la détermination des données à écrire dans un récepteur de streaming. Les formats les plus utilisés sont la console, le kafka, le parquet et la mémoire. Nous utiliserons l'option console comme format afin que nous puissions suivre nos résultats de streaming depuis le terminal. Nous avons trois options pour la méthode outputMode(). Ceux-ci sont :

- **append :** seules les nouvelles lignes seront écrites dans le récepteur.
- **complete :** toutes les lignes seront écrites dans le récepteur, chaque fois qu'il y aura des mises à jour.
- **update :** seules les lignes mises à jour seront écrites dans le récepteur, chaque fois qu'il y aura des mises à jour.

Nous Chargeons les 4 fichiers en streaming un par un, en utilisant le " readStream" : spark.readStream (Batch id : de 0 à 3)

```
{
  "id" : "a02f6a36-419a-4bc7-998d-632e74559e8b",
  "runId" : "e2794161-2ad5-4942-ad7e-bf8e5545b19d",
  "name" : "display_query_1",
  "timestamp" : "2022-01-12T08:57:09.000Z",
  "batchId" : 0,
  "numInputRows" : 0,
  "inputRowsPerSecond" : 0.0,
  "processedRowsPerSecond" : 0.0,
  "durationMs" : {
    "latestOffset" : 171,
    "triggerExecution" : 172
```


FIGURE 4.12 – Importation de fichier 1 : (BatchId=0)

4.4.2 Requêtes SQL

Nous pouvons effectuer des requêtes SQL directement (proche à python), ou bien créer une vue temporaire avec notre dataframe (proche à SQL).

Exemple :

1. Nous cherchons à calculer le nombre d'occurrences des valuers de polarité. Nous exécutons le code suivant :

```
# Calculer nombre d'occurence de polarité
groupe_polarity=df.groupBy('polarity').count()
query=groupe_polarity.writeStream.format("console").outputMode('complete').start()
```

FIGURE 4.13 – Code pour la colonne 'ploarity'

Nous avons comme résultat :

Result updated 6s ago

	polarity	count
1	-0.15384615384615385	1
2	0.12727272727272726	1
3	-0.2916666666666667	3
4	0.07999999999999999	1
5	0.23304473304473305	1
6	-0.060714285714285714	1

Truncated results, showing first 1000 rows.

FIGURE 4.14 – Nombre d'occurrence de polarité

2. Nous cherchons à compter le nombre des sentiments ayant comme valeur de polarité 0.5. Nous exécutons le code suivant :

```
# Afficher les catégories des sentiments ayant comme valeur de polarité 0.5
requete=spark.sql("""
          SELECT sentiment, count(sentiment)
          FROM table WHERE polarity == 0.5
          GROUP BY sentiment""")
```

FIGURE 4.15 – Code pour 'sentiment'

Nous avons comme résultat :

	sentiment	count(sentiment)
1	positive	69
2	neutral	13
3	negative	30

FIGURE 4.16 – Résultat pour 'sentiment'

Et puisque nous avons travaillé dans l'environnement databricks, alors nous pouvons afficher le résultat de nos requêtes de plusieurs façons :

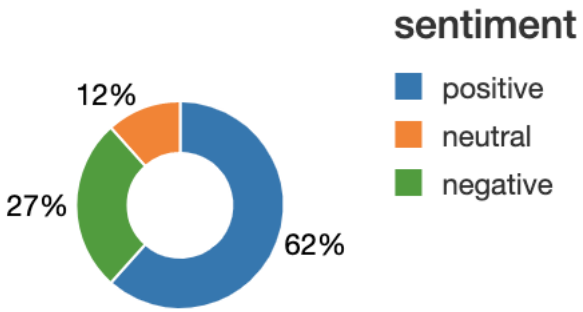

FIGURE 4.17 – Résultat pour 'sentiment' (pie)

Bibliographie

[1] *tweepy documentation*. URL : https://www.pythoncentral.io/introduction-to-tweepy-twitter-for-python/#:~:text=Tweepy%5C%20provides%5C%20access%5C%20to%5C%20the,can%5C%20get%5C%20the%5C%20appropriate%5C%20information..

[2] *tweepy limit*. URL : https://www.geeksforgeeks.org/extraction-of-tweets-using-tweepy/.

[3] *snscrape github*. URL : https://github.com/JustAnotherArchivist/snscrape.

[4] *twint documentation*. URL : https://github.com/twintproject/twint.

[5] *GetOldTweets*. URL : https://pypi.org/project/GetOldTweets3/.

[6] *Spark Stream Documentation*. URL : https://spark.apache.org/docs/latest/streaming-programming-guide.html.

[7] *What is Spark Streaming ?* URL : https://databricks.com/glossary/what-is-spark-streaming#:~:text=Spark%5C%20Streaming%5C%20is%5C%20an%5C%20extension,%5C%2C%5C%20databases%5C%2C%5C%20and%5C%20live%5C%20dashboards..

[8] Federico Pozzi et al. *Sentiment analysis in social networks*. Morgan Kaufmann, 2016.

[9] *tweet-preprocessor documentation*. URL : https://pypi.org/project/tweet-preprocessor/.

[10] *string methods*. URL : https://docs.python.org/3/library/string.html.

[11] *nltk documentation*. URL : https://www.nltk.org/.