



UNIVERSITE IBN TOUFAIL

FACULTE DES SCIENCES

Master Génie Logiciel pour Le Cloud

Présenté par : Abdellah BIROUK & Ayoub MANÇOUR BILLAH

Deep Learning/ Machine Learning

Emotion Recognition (using text)

Année Universitaire 2020-2021

Table des matières

I. INTRODUCTION	3
II. RÉALISATION.....	5
1. Python	5
2. JupyterLab	6
3. Anaconda.....	6
4. Google Colab.....	7
5. Pandas	7
6. NumPy	8
7. Matplotlib	8
8. Neattext	8
III. DATASET.....	9
IV. Test et fonctionnement :	10
1. Phase de nettoyage et traitement du texte :	10
2. Phase de la reconnaissance et du calcul :.....	11
3. Présentation du graphe de notre étude :	13
V. CONCLUSION	14

Liste des figures

Figure 1. Logo Python	5
Figure 2. Logo JupyterLab	6
Figure 3. Logo Anaconda.....	6
Figure 4. Logo Google Colab	7
Figure 5. Logo Pandas.....	7
Figure 6. Logo NumPy	8
Figure 7. Logo Matplotlib	8
Figure 8. Exemple DataSet.....	9
Figure 9. Bibliothèques	10
Figure 10. Nettoyage du texte.....	10
Figure 11. Suppression des stop_words	11
Figure 12. émotions.txt	12
Figure 13. Calcul et Affichage	13
Figure 14. Graphe.....	13

I. INTRODUCTION

Dans ce projet nous allons vous présenter une reconnaissance émotionnelle en utilisant un texte (Emotion Recognition using text), ce projet est un système de reconnaissance des émotions multimodal qui est construit pour extraire des informations sur les émotions de la saisie de texte. Le système de reconnaissance des émotions classe les émotions selon différents types de base : le bonheur, la tristesse, la colère, la peur, la surprise, le dégoût... Si la valeur d'intensité d'émotion de l'émotion actuellement reconnue est inférieure à un seuil prédéfini, la sortie d'émotion est déterminée comme étant neutre.

Le système de reconnaissance des émotions proposé peut détecter les émotions à partir du texte. Pour évaluer l'approche acoustique, un drame diffusé, y compris le contenu textuel. Lors de la sélection des fonctionnalités, un ensemble initial de fonctionnalités acoustiques contenant 33 fonctionnalités est d'abord analysé et la reconnaissance multimodale d'émotion à partir du texte est extraite.

II. RÉALISATION

Durant la réalisation de ce projet nous avons été menés à l'utilisation du langage de programmation PYTHON ainsi que plusieurs bibliothèques de ce langage.

- **Programmes et langages utilisés :**

1. Python

Python est un langage de programmation interprété, multi paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions.



Figure 1. Logo Python

2. JupyterLab

JupyterLab est un environnement de développement interactif basé sur le Web pour les blocs-notes, le code et les données Jupyter.

JupyterLab est flexible : configurez et organisez l'interface utilisateur pour prendre en charge un large éventail de flux de travail dans les domaines de la science des données, de l'informatique scientifique et de l'apprentissage automatique. JupyterLab est extensible et modulaire : écrivez des plugins qui ajoutent de nouveaux composants et s'intègrent aux composants existants.



Figure 2. Logo JupyterLab

3. Anaconda

Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique, qui vise à simplifier la gestion des paquets et de déploiement.



Figure 3. Logo Anaconda

4. Google Colab

Google Colab est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.



Figure 4. Logo Google Colab

- **Bibliothèques Utilisées :**

5. Pandas

Pandas est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles. Pandas est un logiciel libre sous licence BSD.



Figure 5. Logo Pandas

6. NumPy

NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.



Figure 6. Logo NumPy

7. Matplotlib

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy.

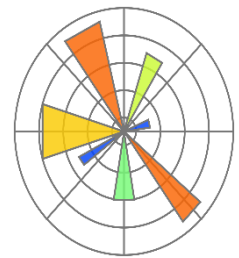


Figure 7. Logo Matplotlib

8. Neattext

Neattext est un simple package NLP python pour nettoyer les données textuelles et pour traiter le texte lors de l'exécution de projets NLP et ML. Il a été conçu pour résoudre le problème suivant :

- ✓ Nettoyage des données textuelles non structurées
- ✓ Réduire le bruit [caractères spéciaux, mots vides]
- ✓ Réduire la répétition de l'utilisation du même code pour le prétraitement de texte.

III. DATASET

Nous avons eu recours à une multitude de données. Un fichier contenant plus de 500 mots qui expriment des sentiments et des réactions à notre représentation que l'on se fait de quelque chose. Ils expriment ce que l'on perçoit d'une situation et des pensées qui y sont associées.

```
'victimized': 'triché',  
'accused': 'triché',  
'acquitted': 'distingué',  
'adorable': 'aimé',  
'adored': 'aimé',  
'affected': 'attiré',  
'afflicted': 'triste',  
'aghast': 'craintif',  
'agog': 'attiré',  
'agonized': 'triste',  
'alarmed': 'craintif',  
'amused': 'heureux',  
'angry': 'en colère',  
'anguished': 'triste',  
'animated': 'heureux',  
'annoyed': 'en colère',  
'anxious': 'attiré',  
'appalled': 'en colère',  
'appeased': 'distingué',  
'appreciated': 'estimé',  
'apprehensive': 'craintif',
```

Figure 8. Exemple DataSet

IV. Test et fonctionnement :

Dans ce chapitre, on va traiter les différentes phases de notre méthode de reconnaissance textuelles des émotions.

1. Phase de nettoyage et traitement du texte :

Dans une première partie, on importe les bibliothèques dont on a besoin :

```
[1]: import string
      from collections import Counter

      import matplotlib.pyplot as plt
```

Figure 9. Bibliothèques

Ensuite, on commence par ouvrir notre fichier texte (.txt) par l'encodage UTF-8 et le passer dans une variable. Notre traitement commencera par passer la variable en minuscule, supprimer la ponctuation et diviser le texte en mot par mot dans une liste :

```
[2]: # On fait passer notre fichier txt à la variable text avec l'encodage utf-8:
      text = open("read.txt", encoding="utf-8").read()

      # On convertit le texte en minuscule car en Python on respecte la casse : text # Text
      lower_case = text.lower()

      # On supprime la ponctuation pour nettoyer plus le texte:
      cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))

      # On fait diviser le texte en mots par la fonction split():
      tokenized_words = cleaned_text.split()
```

Figure 10. Nettoyage du texte

Après, on passe à la dernière étape du nettoyage du texte : les 'stop words' qui sont des mots qui n'ont aucune valeur dans notre étude émotionnelle. Par exemple les pronoms personnels ou les accords 'am, how, he, she, and...', on a créé une liste contenant la plupart des noms répétitifs. On compare alors chaque élément de notre liste 'tokenized_words' avec les 'stop_words' par une boucle 'for' en supprimant les mots qui en appartiennent :

```
[3]: # On supprimer les stop_words de notre liste de mots (texte):
    final_words = []
    for word in tokenized_words:
        if word not in stop_words:
            final_words.append(word)
```

Figure 11. Suppression des stop_words

2. Phase de la reconnaissance et du calcul :

En premier lieu, on a créé dans un fichier 'émotions.txt' une liste contenant la plupart des mots qui peuvent exprimer un sentiment quelconque (environ 500 mots).

```
1 | 'victimized': 'triché',
2 | 'accused': 'triché',
3 | 'acquitted': 'distingué',
4 | 'adorable': 'aimé',
5 | 'adored': 'aimé',
6 | 'affected': 'attiré',
7 | 'afflicted': 'triste',
8 | 'aghast': 'craintif',
9 | 'agog': 'attiré',
10 | 'agonized': 'triste',
11 | 'alarmed': 'craintif',
12 | 'amused': 'heureux',
13 | 'angry': 'en colère',
14 | 'anguished': 'triste',
15 | 'animated': 'heureux',
16 | 'annoyed': 'en colère',
17 | 'anxious': 'attiré',
18 | 'appalled': 'en colère',
19 | 'appeased': 'distingué',
20 | 'appreciated': 'estimé',
21 | 'apprehensive': 'craintif',
22 | 'approved of': 'aimé',
23 | 'ardent': 'lascif',
24 | 'aroused': 'lascif',
25 | 'attached': 'attaché',
26 | 'attracted': 'attiré',
27 | 'autonomous': 'indépendant',
28 | 'awed': 'craintif',
29 | 'awkward': 'embarrassé',
30 | 'beaten down': 'impuissant',
31 | 'beatific': 'heureux',
32 | 'belonging': 'attaché',
33 | 'bereaved': 'triste',
34 | 'betrayed': 'triché',
35 | 'bitter': 'en colère',
```

Figure 12. émotions.txt

Après, on vérifie si chaque élément de la liste [final_words] appartient à la liste des émotions :

On ouvre la liste des émotions -> On passe par chaque ligne -> Si le mot appartient, on l'extrait avec la méthode split().

Si le mot est présent dans la liste des émotions, on l'ajoute à [emotion_list].

Et à la fin on calcule le nombre de chaque sentiment.

```
[4]: emotion_list = []
with open('emotions.txt', 'r', encoding="utf-8") as file:
    for line in file:
        clear_line = line.replace("\n", '').replace(", ", '').replace("'", '').strip()
        word, emotion = clear_line.split(':')

        if word in final_words:
            emotion_list.append(emotion)

print(emotion_list)
w = Counter(emotion_list)
print(w)
```

Figure 13. Calcul et Affichage

3. Présentation du graphe de notre étude :

Dans la dernière étape, on représente notre étude sous forme de graphe contenant les différentes émotions avec le nombre d'occurrences à l'aide de la bibliothèque matplotlib :

Présenter Les sentiments (émotions dans un graphe):

```
with plt.style.context('ggplot'):
    fig, ax1 = plt.subplots(figsize=(3, 3), dpi=100)
    ax1.bar(w.keys(), w.values())
    fig.autofmt_xdate()
    plt.title('Graphe des sentiments')
    plt.ylabel('Nombre d\'occurrences')

    plt.savefig('graph.png')
plt.show()
```

```
[' heureux', ' heureux', ' attaché', ' heureux', ' attiré', ' seul', ' libre', ' détesté', ' heureux', ' intitulé', ' heureu
x', ' aimé', ' détesté', ' intitulé']
Counter({' heureux': 5, ' détesté': 2, ' intitulé': 2, ' attaché': 1, ' attiré': 1, ' seul': 1, ' libre': 1, ' aimé': 1})
```

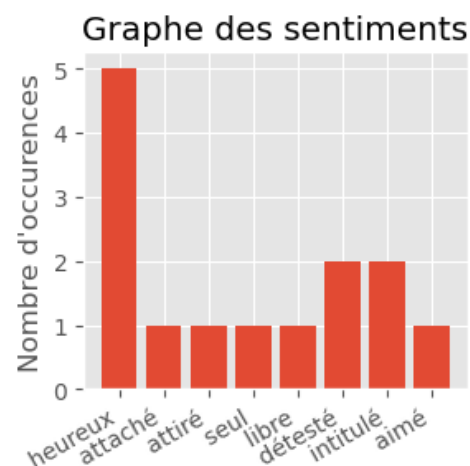


Figure 14. Graphe

V. CONCLUSION

Avec ce projet nous avons étudié une méthode de reconnaissance textuelle des émotions. Cette application qui repose sur l'apprentissage profond (Deep Learning) qui peut extraire des émotions des informations textuelles sans avoir besoin d'un logiciel de reconnaissance textuelle sophistiqué. Cette reconnaissance textuelle permet d'obtenir une vue d'ensemble sur l'opinion du public au sujet de certains thèmes, cela aide les entreprises à extraire des informations à partir des données du web social.