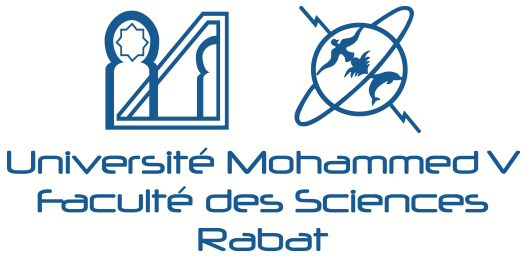
UNIVERSITE MOHAMMED V de Rabat
Faculté des Sciences



Master Ingénierie de données et de developpement logiciel

PROJET NLP

Intitulé :

Developpement d'une application WEB pour l'analyse des sentiments en anglais francais et arabe

Présenté par:
MEHDI LAGHEZALI

LABRHADDA SAAD

Année universitaire 2020-2021

Table des matières

Table des figures

1 Introduction

Le traitement du langage naturel (NLP) est la capacité d'un programme informatique à comprendre le langage humain tel qu'il est parlé et écrit, appelé langage naturel. C'est une composante de l'intelligence artificielle.
Le NLP existe depuis plus de 50 ans et a des racines dans le domaine de la linguistique. Il a une variété d'applications du monde réel dans un certain nombre de domaines, y compris la recherche médicale, les moteurs de recherche et la veille économique.
Le traitement du langage naturel comporte deux phases principales : le prétraitement des données et le développement de l'algorithme.
Le prétraitement des données implique la préparation et le « nettoyage » des données de texte pour que les machines puissent les analyser. Le prétraitement met les données sous une forme exploitable et met en évidence les caractéristiques du texte avec lesquelles un algorithme peut fonctionner.
Cependant, ces étapes permettront de réaliser plusieurs traitements et applications qui faciliteront la compréhension de l'être humain et comment il arrive à prendre ces décisions.
Parmi les applications du NLP on a le sentiment Analysis qui est l'une des applications les plus fréquentes de nos jours.
C'est une exploration contextuelle de texte qui identifie et extrait des informations subjectives dans le matériel source et aide une entreprise à comprendre le sentiment social de sa marque, de son produit ou de son service tout en surveillant les conversations en ligne.
Il était important de classer les conversations et les commentaires des clients afin d'assimiler l'arrière-pensée de chaque individu sur un sujet, un produit ou un service donné.
Les gens considèrent souvent le sentiment (positif ou négatif) comme la valeur la plus importante des opinions exprimées par les utilisateurs via les médias sociaux.
Cependant, en réalité, les émotions fournissent un ensemble plus riche d'informations qui concernent les choix des consommateurs et, dans de nombreux cas, déterminent même leurs décisions.

2 Bibliothèques utilisées

2.1 NLTK

Natural Language Toolkit (NLTK) est une bibliothèque logicielle en Python permettant un traitement automatique des langues, développée par Steven Bird et Edward Loper du département d'informatique de l'université de Pennsylvanie. En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation

2.2 HuggingFace

C'est une bibliothèque Python extrêmement populaire fournissant des modèles pré-entraînés extrêmement utiles pour une variété de tâches de traitement du langage naturel (NLP).
Auparavant , il ne prenait en charge que PyTorch , mais, depuis fin 2019, TensorFlow 2 est également pris en charge .
Bien que la bibliothèque puisse être utilisée pour de nombreuses tâches, de l'inférence du langage naturel à la question-réponse, la classification de texte reste l'un des cas d'utilisation les plus populaires et les plus pratiques.

2.3 Flask / Jinja

Flask est un framework de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de templates . Flask est basé sur deux modules werkzeug et jinja qui permettent de simplifier l'écriture des tests unitaires.

Jinja est un moteur de modele de flask , il est un fichier texte qui peut generer n'importe quel format texte (LATEX ,HTML,XML,ect.) , du coup il n'a pas besoin d'avoir une extension specifique .

2.4 RegEx

La bibliothèque d'expressions régulières fournit une classe qui représente les expressions régulières , qui sont une sorte de mini-langage utilisé pour effectuer une correspondance de modèle dans des chaînes.
Presque toutes les opérations avec des expressions régulières peuvent être caractérisées en opérant sur plusieurs d'objets.

2.5 TensorFlow

TensorFlow est une plate-forme Open Source de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie.

3 Application web

3.1 Environnement virtuel

L'environnement virtuel est très intéressant lors d'un développement , c'est un répertoire qui contient une installation de Python , ou on peut installer des bibliothèques dans une version différente sans les installer sur le système .

3.2 Les Dossiers

Bien qu'il ne nécessite pas d'architecture spécifique, il existe quelques bonnes pratiques à suivre:

- sentiment-analysis.py : il contient le code qui effectuera le traitement du texte ainsi que l'analyse du sentiment et fera des prédictions.
- app.py : est le code principal qui exécutera notre application Flask. Il contiendra les différentes routes pour notre application, répondra aux requêtes HTTP et décidera quoi afficher dans les modèles. Dans notre cas, il appellera également notre fichier "sentiment-analysis.py".
- Dossier Templates : Un modèle est un fichier HTML qui peut recevoir des objets Python et est lié à l'application Flask. Par conséquent, nos pages html seront stockées dans ce dossier.
- Dossier static : les feuilles de style, scripts, images et autres... Nous y placerons nos fichiers Javascript et CSS.

Ce projet nécessitera:

- des fichiers statiques: des fichiers CSS, JS et bootstrap ainsi que des images.
- un fichier template: index.html.
- Notre fichier principal: app.py

.

3.3 *Script Python*

Dans ce script nous avons défini 3 fonctions principales pour chaque langue (Anglais , Français,Arabe),ces 3 fonctions suivront les memes étapes du Pre-Processing à savoir la suppression des ponctuations ainsi que la tokenisation des mots par le biais de la méthode word-tokenize de la bibliothèque nltk, l'utilité de Cette méthode et d'avoir des données séparées ce qui nous facilitera l'analyse de chaque mot indépendamment.

Pour La fonction ar-sentiment c'est encore plus précise que les autres fonctions, puisque l'arabe contient des caractères speciales et une ponctuation (point , point d'interrogation ...) différente des autres langues.

la variable arabic-punctuations contient donc toutes les ponctuations de la langue arabe,ensuite il faut enlever les signes diacritiques arabes qui peuvent poser des problèmes lors du traitement.

On a effectué la méthode des Stop Words qui éliminera des mots inutiles s'il existe dans la liste des mots vides fournie par NLTK pour chaque langue afin d'avoir des informations utiles prête a être analyser sans ambiguïté.

Néanmoins ,chacune de ces fonctions aura son propre model d'analyse:

Distilbert :est un modèle de transformer, plus petit et plus rapide que BERT, qui a été pré-formé sur le même corpus de manière auto-supervisée, en utilisant le modèle de base BERT comme enseignant. Ce model est entrainé pour analyser des paragraphes en Anglais avec une probabilité similaire à Bert mais plus rapide .

tblard/tf-allocine: ce Pre-trained model est un modèle d'analyse des sentiments en français, basé sur CamemBERT , et affiné sur un ensemble de données à grande échelle extrait des avis des utilisateurs d' Allociné.fr
akhooli/xlm-r-large-arabic-sent: qui est une classification multilingue des sentiments des critiques en arabe en ajustant XLM-Roberta-Large.

Ces modèles seront affectés au texte à analyser à travers la méthode pipeline de la bibliothèque Transformers

Pour le fichier app.py qui est le fichier principale , il permet de récuperer le texte saisi dans la page web ainsi que la langue choisie, ensuite il suffit de faire l'appel a la fonction qui fait l'analyse du texte. Une fois le resultat recupéré, il faut calculer la taille de l'arc N pour que la barre de progression circulaire soit correcte. Finalement, il suffit d'envoyer la variable N, le texte et le score à la page WEB.

```
def ang_sentiment(text):
    lower_case= text.lower()
    cleaned_text = lower_case.translate(str.maketrans('','',string.punctuation))
    tokenized_words = word_tokenize(cleaned_text,"english")

    final_words=[]
    for word in tokenized_words:
        if word not in stopwords.words("english"):
            final_words.append(word)

    result=" ".join(final_words)
    nlp = pipeline('sentiment-analysis')
    return (nlp(result))
```

Figure 1: Script du traitement du texte anglais

```
def fr_sentiment(text):
    lower_case= text.lower()
    cleaned_text = lower_case.translate(str.maketrans('','',string.punctuation))
    tokenized_words = word_tokenize(cleaned_text,"french")

    tokenizer = AutoTokenizer.from_pretrained("tblard/tf-allocine")
    model = TFAutoModelForSequenceClassification.from_pretrained("tblard/tf-allocine")


    final_words=[]
    for word in tokenized_words:
        if word not in stopwords.words("french"):
            final_words.append(word)

    result=" ".join(final_words)
    nlp = pipeline('sentiment-analysis',model=model,tokenizer=tokenizer)
    return (nlp(result))
```

Figure 2: Script du traitement du texte français

```python
def ar_sentiment(text):
    arabic_punctuations = '''`÷×:<>_()*&^%][_-"…”!|+¦~{}',.؟":/._'''
    english_punctuations = string.punctuation
    punctuations_list = arabic_punctuations + english_punctuations
    arabic_diacritics = re.compile("""
                                ّ    | # Tashdid
                                َ    | # Fatha
                                ً    | # Tanwin Fath
                                ُ    | # Damma
                                ٌ    | # Tanwin Damm
                                ِ    | # Kasra
                                ٍ    | # Tanwin Kasr
                                ْ    | # Sukun
                                ـ     # Tatwil/Kashida
                            """, re.VERBOSE)
    translator = str.maketrans('', '', punctuations_list)
    text=text.translate(translator)

    text = re.sub(arabic_diacritics, '', text)

    text=re.sub(r'(.)\1+', r'\1', text)

    tokenized_words = word_tokenize(text)

    sw=stopwords.words('arabic')
    tokens=[i for i in tokenized_words if not i in sw]

    result=" ".join(tokens)

    tokenizer = AutoTokenizer.from_pretrained("akhooli/xlm-r-large-arabic-sent")

    model = AutoModelForSequenceClassification.from_pretrained("akhooli/xlm-r-large-arabic-sent")

    nlp = pipeline("sentiment-analysis",model=model,tokenizer=tokenizer)

    return (nlp(result))
```

Figure 3: Script du traitement du texte arabe

```python
from flask import Flask, render_template, request, redirect, url_for
from transformers import pipeline
import math
import sentiment_analysis

app = Flask(__name__)
app.config["IMAGE_UPLOADS"] = r"C:\Users\hp\project-nlp\App\static"
@app.route('/index',methods=['GET','POST'])

def index():
    if request.method == 'POST':

        text=request.form["text"]
        lang=request.form["lang"]

        if lang=="1":
            result= sentiment_analysis.ang_sentiment(text)
            if result[0]['score']<0.55 and result[0]['score']>0.45:
                result[0]['label']="NEUTRAL"
        elif lang=="2":
            result= sentiment_analysis.fr_sentiment(text)
            if result[0]['score']<0.55 and result[0]['score']>0.45:
                result[0]['label']="NEUTRAL"
        else:
            result= sentiment_analysis.ar_sentiment(text)
            if result[0]['label']=="LABEL_0":
                result[0]['label']="NEUTRAL"

            elif result[0]['label']=="LABEL_1":
                result[0]['label']="NEGATIVE"

            else:
                result[0]['label']="POSITIVE"

        n = ((100-math.floor(result[0]['score']*100))/100)*math.pi*(115*2)

        return render_template("index.html",number= n,text=text ,note=result[0]['label'],result=result[0]['score'])

    return render_template("index.html",number=720,result=0)

if __name__ == "__main__":
    app.run()
```

Figure 4: Code principal

4 Simulation

4.1 Description de la page d'acceuil:

La page d'acceuil de notre application web se modelise à partir des bases du language html ,css, et javascript ,facile à manipuler par n'importe quel utilisateur .Cette premiere page permet à l'utilisateur d'inserer n'importe quel texte afin de le traiter, analyser le sentiment et avoir le score final.
Il est nécessaire de choisir la langue adéquate avant de lancer le programme pour arriver aux résultats souhaités.

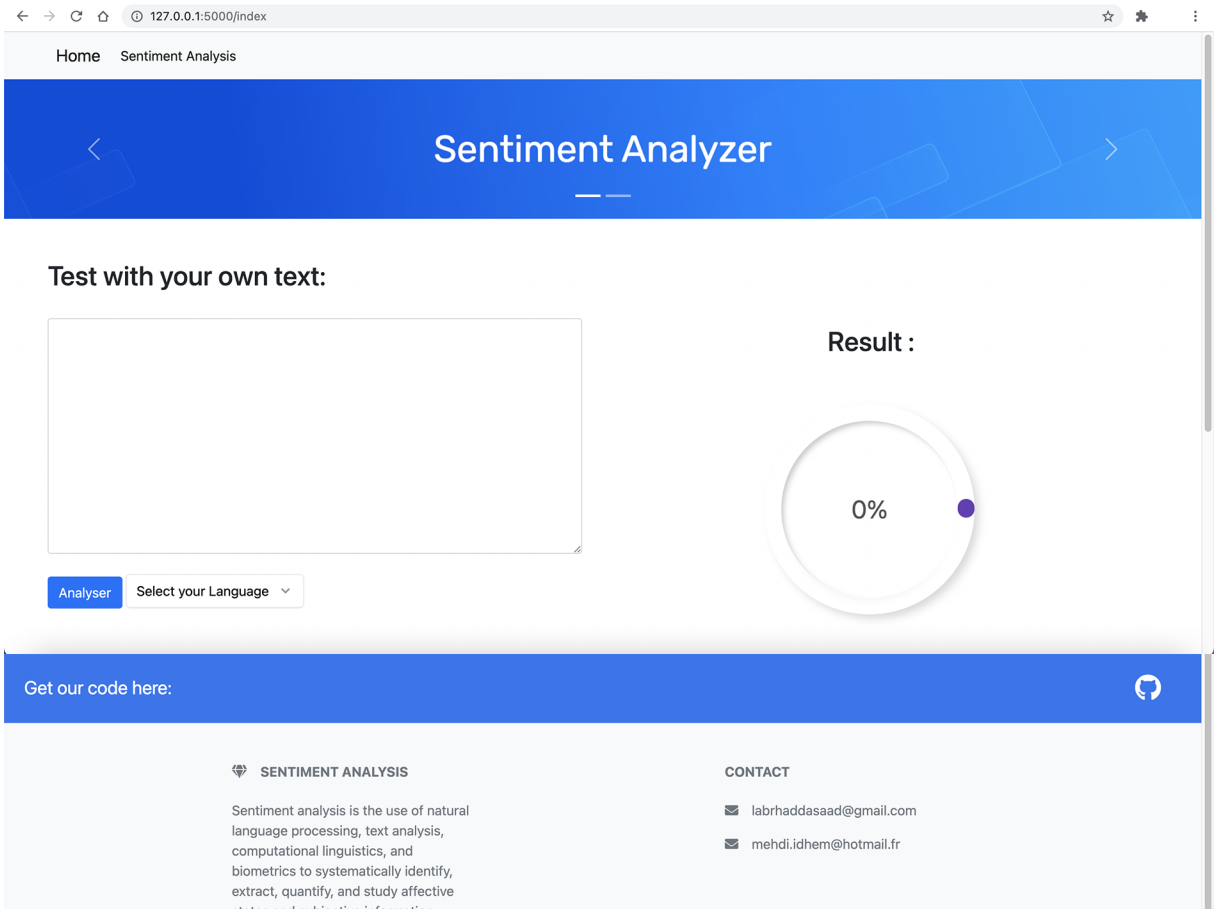


Figure 5: page principale

4.2 Traitement du texte:

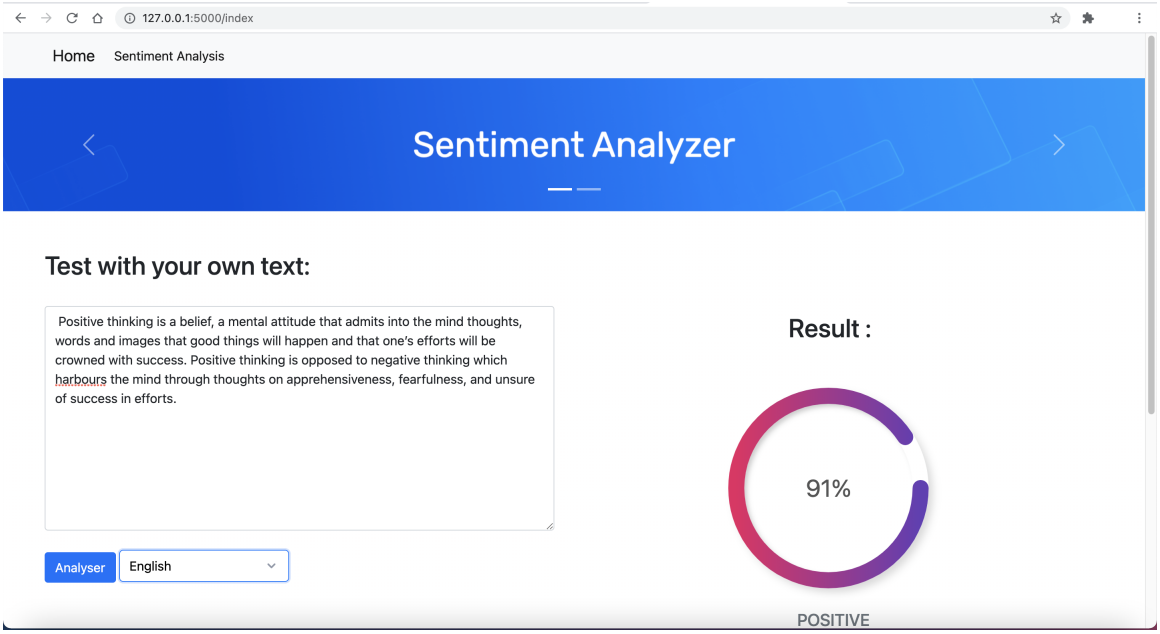


Figure 6: Traitement d'un texte anglais

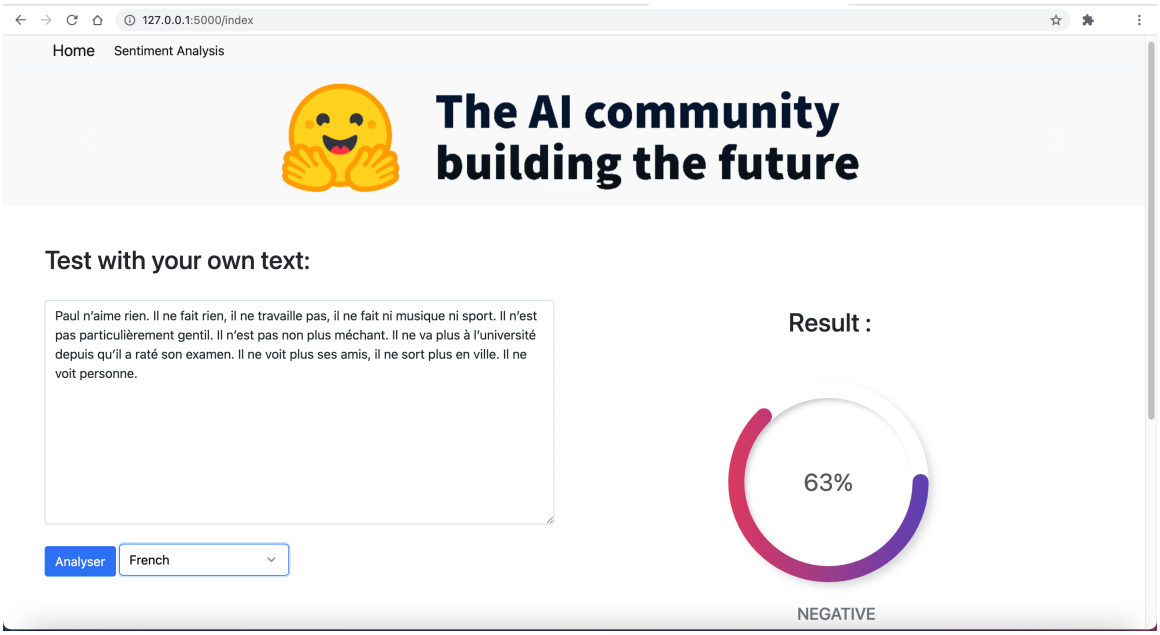


Figure 7: Traitement d'un texte français

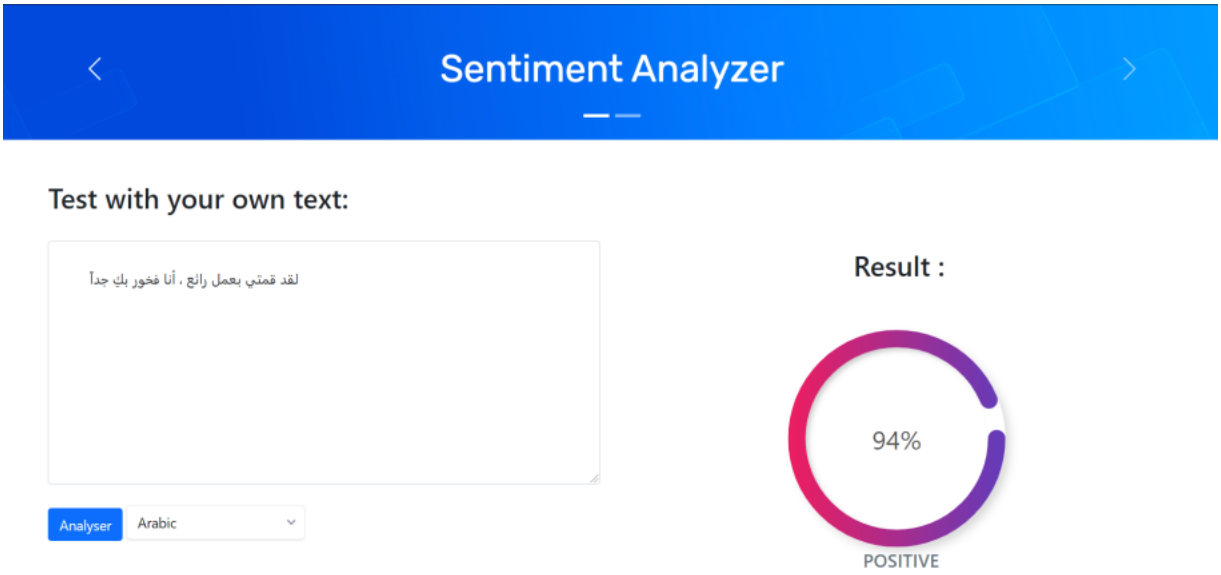


Figure 8: Traitement d'un texte arabe

5 Conclusion

En guise de conclusion, force est de constater que le projet permet à l'étudiant de mettre en relief ce qu'il a acquis comme capacités techniques, intellectuelles et professionnelles enseignées au cours de sa formation.

Après la réalisation pratique, et le test des différentes opérations, nous affirmons que les objectifs tracés au début du projet sont réalisés, puisque le travail a été effectué avec anxiété et conformément aux règles de l'art.

Toutefois, et en tant que débutants, nous avons rencontré quelques problèmes durant la conception de la commande et la réalisation pratique, mais grâce à ce que nous avons appris durant ce mois , nous avons pu faire face à ces difficultés.

Ainsi, nous trouvons que notre expérience était bénéfique pour notre formation puisqu'il nous a permis d'avoir un aperçu fructueux sur le domaine de NLP . De même, elle nous a permis de mettre à profit les études des semestres précédents et de parfaire nos connaissances dans des matières déjà étudiées.

Enfin, l'un des avantages majeurs de ce projet réside dans le travail en binôme. Il nous a permis d'apprendre la méthode de répartition des tâches entre les membres du groupe,l'organisation du travail de sorte à s'enrichir mutuellement en se partageant les idées.

6 Code Source

[1] https://github.com/LABSAAD/projet-nlp.git

[2] https://github.com/MehdiLaghezali/project-nlp