

▼ Sentiment Analysis

Application sur synthèse rapport COR

L'analyse de sentiment consiste à déterminer l'attitude ou l'émotion d'un document à l'égard d'un sujet donné (retraite).

```
# Installation librairie PyPDF2 pour gérer fichier PDF
```

```
!pip install PyPDF2
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: PyPDF2 in /usr/local/lib/python3.8/dist-packages (3.0.1)
Requirement already satisfied: typing_extensions>=3.10.0.0 in /usr/local/lib/python3.8/dist-packages (from PyPDF2) (4.5.0)
```

```
!pip install textblob-fr
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: textblob-fr in /usr/local/lib/python3.8/dist-packages (0.2.0)
Requirement already satisfied: textblob>=0.8.0 in /usr/local/lib/python3.8/dist-packages (from textblob-fr) (0.15.3)
Requirement already satisfied: nltk>=3.1 in /usr/local/lib/python3.8/dist-packages (from textblob>=0.8.0->textblob-fr) (3.7)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from nltk>=3.1->textblob>=0.8.0->textblob-fr) (4.64.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.8/dist-packages (from nltk>=3.1->textblob>=0.8.0->textblob-fr) (1.2.0)
Requirement already satisfied: click in /usr/local/lib/python3.8/dist-packages (from nltk>=3.1->textblob>=0.8.0->textblob-fr) (7.1.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.8/dist-packages (from nltk>=3.1->textblob>=0.8.0->textblob
```

```
# Chargement librairies
```

```
import PyPDF2 # lecture fichier PDF
from textblob import TextBlob # Calcul polarité du document
from textblob_fr import PatternAnalyzer # Pour calcul de polarité avec texte en français
import re # Chargement Regular Expressions
import matplotlib.pyplot as plt
```

```
# Lecture du document PDF et extraction du texte
```

```
with open ("Synthèse.pdf", "rb") as f:
    pdf_reader = PyPDF2.PdfReader(f)
    text = ''
    for page in range(len(pdf_reader.pages)):
        page_obj = pdf_reader.pages[page]
        text += page_obj.extract_text()
```

```
# Visualisation du texte
```

```
text
```

```
' Rapport annuel du COR - Septembre 2022 \n \n \n1 \nSynthèse \n \n \n1. Une dynamique des dépenses
de retraite globalement toujours contenue par rapport à \n1'évolution de la richesse nationale, mais
un niveau en augmentation par rapport aux \ndernières projections \n Les dépenses du système de retra
ite rapportées au PIB constituent un indicateur déterminant \npour évaluer la soutenabilité financière
du système de retraite ; il exprime, de manière globale \net synthétique, le niveau des prélèvements q
u'il faut opérer sur la richesse produite par les actifs pour assurer l'équilibre. \nDépenses du syst
ème de retraite en % du PIB observées et projetées \n \n \n \nSources : rapports à la CCSS 2002 -20
```

```
# Suppression de l'entête du document
```

```
text = re.sub(r'Rapport annuel du COR - Septembre 2022', '', text)
text
```

```
' \n \n \n1 \nSynthèse \n \n \n1. Une dynamique des dépenses de retraite globalement toujours con
tenue par rapport à \n1'évolution de la richesse nationale, mais un niveau en augmentation par rapport
aux \ndernières projections \n Les dépenses du système de retraite rapportées au PIB constituent un i
ndicateur déterminant \npour évaluer la soutenabilité financière du système de retraite ; il exprime,
de manière globale \net synthétique, le niveau des prélèvements qu'il faut opérer sur la richesse prod
uite par les actifs pour assurer l'équilibre. \nDépenses du système de retraite en % du PIB observée
s et projetées \n \n \n \nSources : rapports à la CCSS 2002 -2021 : projections COR - septembre 202
```

▼ Pipeline proposée pour Sentiment Analysis

Voici les étapes couramment suivies pour réaliser une analyse de sentiment :

1. **Pré-traitement du document.** Il est important de nettoyer les données en enlevant les caractères inutiles tels que les signes de ponctuation, les nombres, etc ;
2. **Tokenization.** Diviser le document en tokens (mots, phrases, séquences, etc.) pour une analyse plus fine ;
3. **Étiquetage des sentiments.** Assigner une étiquette de sentiment (positive, négative, neutre) à chaque token en utilisant des dictionnaires de sentiment, des algorithmes de classification automatique ou d'autres techniques ;
4. **Agrégation des sentiments.** Une fois que tous les tokens ont été étiquetés, il est nécessaire d'agréger les sentiments pour obtenir une vue d'ensemble du document. Cela peut se faire en comptant simplement le nombre de tokens positifs, négatifs et neutres, ou en utilisant des

algorithmes plus avancés pour pondérer les sentiments en fonction de leur importance relative ;

5. **Interprétation des résultats.** Enfin, les résultats de l'analyse de sentiment peuvent être interprétés et utilisés pour comprendre l'attitude générale de l'auteur à l'égard du sujet.

#chargement librairie NLTK

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import string
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Package omw-1.4 is already up-to-date!
True
```

Pré-traitement du document

Création d'une fonction *clean_text* qui enlève les caractères inutiles du texte.

```
def clean_text(text):
    # suppression des nombres
    text_nonum = re.sub(r'\d+', '', text)
    # suppression de la ponctuation et conversion en minuscule
    text_nopunct = "".join([char.lower() for char in text_nonum if char not in string.punctuation])
    # suppression caractères non alphanumérique ou soulignés
    text_only_alphanum = re.sub(r'[\W_]', ' ', text_nopunct)
    # substitution espaces multiples en espaces simples
    # Suppression également des espaces blancs de début et de fin
    text_no_double_space = re.sub('\s+', ' ', text_only_alphanum).strip()
    # suppressions mots spécifiques
    text_last = re.sub('synthèse|retraite|cor|rapport|convention|insee|système|régime', '', text_no_double_space)
    return text_last
cleaned_text = clean_text(text)
print(cleaned_text)
```

une dynamique des dépenses de globalement toujours contenue par à l'évolution de la richesse nationale mais un niveau en augment

Tokenization

Décomposition du texte en éléments simples (tokens) pour permettre l'analyse

Tokenization : diviser le texte en mots individuels

```
tokens = word_tokenize(cleaned_text)
print(tokens[0:15])

['une', 'dynamique', 'des', 'dépenses', 'de', 'globalement', 'toujours', 'contenue', 'par', 'à', 'l', 'évolution', 'de', 'la', 'ric
```

Supprimer les stop words

```
stop_words = set(stopwords.words('french')) # Changer 'french' pour utiliser une autre langue
filtered_tokens = [token for token in tokens if not token in stop_words]
print(filtered_tokens[0:20])

['dynamique', 'dépenses', 'globalement', 'toujours', 'contenue', 'évolution', 'richesse', 'nationale', 'niveau', 'augmentation', 'd
```

Étiquetage des sentiments et agrégation des sentiments

A chaque token on lui associe une étiquette de sentiment (positive, négative, neutre) en utilisant des dictionnaires de sentiment (ici) Ensuite, on agrège les sentiments de tous les tokens (moyenne).

Analyse de sentiment en utilisant TextBlob

```
filtered_tokens_str = ' '.join(filtered_tokens)
```

(0.05625714285714284, 0.13782857142857144)

Le score de subjectivité du document est : 0.13782857142857144
Le document est plutôt objectif.

[illegible]

```
words_freq = blob.word_counts
```

```
sort_words_freq = sorted(words_freq.items(), key=lambda item: item[1])
len(sort_words_freq)
selection = sort_words_freq[610:625]
```

```
ind = []
fre = []
for item in selection:
    ind.append(item[0])
    fre.append(item[1])
```

```
plt.rcParams["figure.figsize"] = [12, 6]
plt.rcParams["figure.autolayout"] = True
```

```
plt.bar(ind, fre)
```

```
plt.show()
```

