Résumé automatique

Projet en «NLP : Natural Language Processing »

Réalisé par :

Abdallah ben othmen trabelsi

Proposé par :

Dr. Abir MASMOUDI

Année universitaire 2022-2023

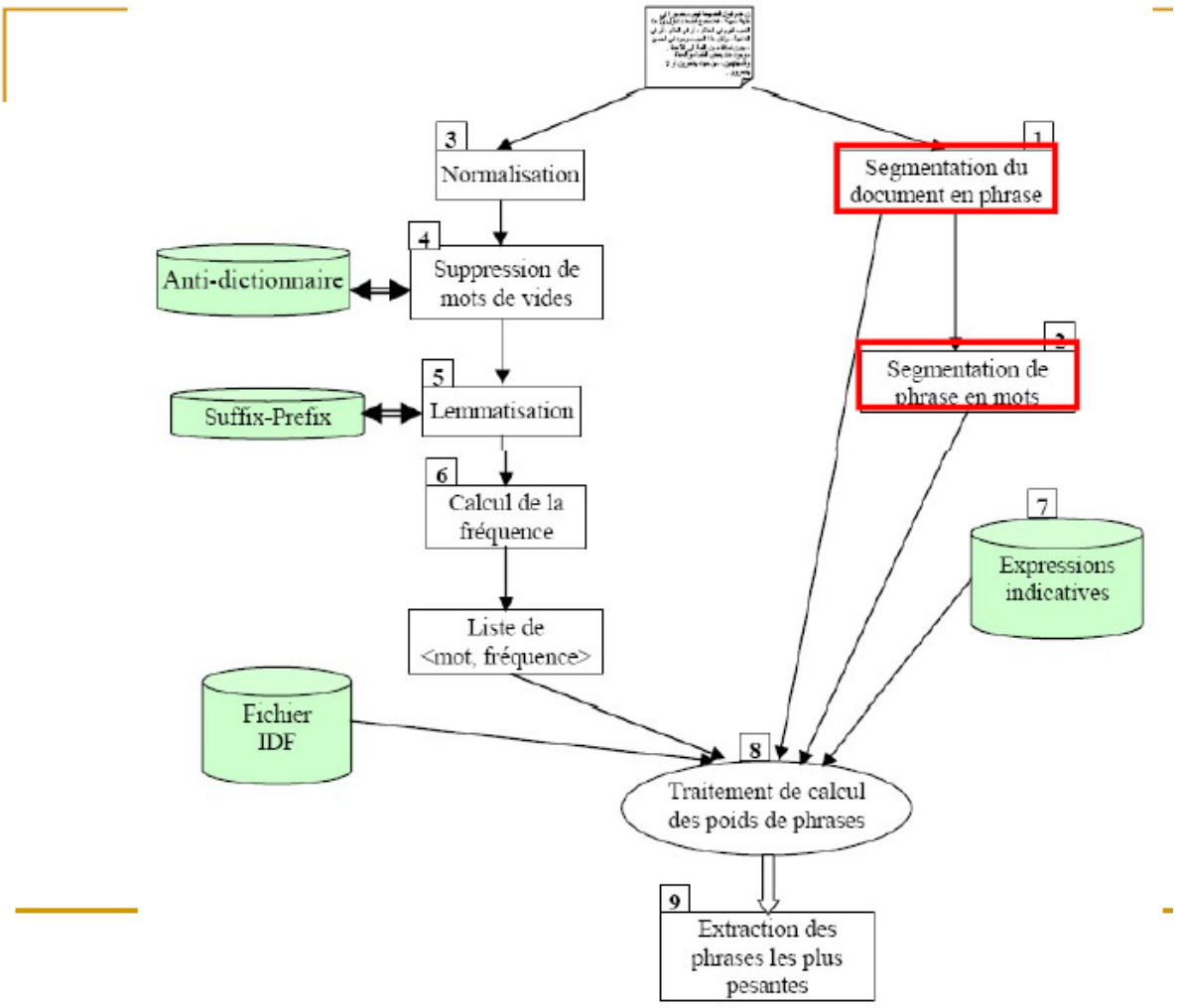
Résumé automatique

I. Contexte et objectif

En se basant sur la méthode de résumé automatique présentée dans le cours, développer un système de résumé automatique qui se base sur la fréquence des mots et les mots des titres pour extraire les phrases les plus pertinentes (importantes). Le système affiche à la fin les phrases les plus pertinentes.

Langage de programmation : Python

Rappel de la méthode de résumé automatique vue dans le cours :



II. Travail demandé:

Dans ce travail, nous proposons de suivre les étapes suivantes pour réaliser ce projet.

- Importer les bibliothèques (NLTK et autres si nécessaires)

```
import bs4 as bs
import urllib.request
import re
import nltk
nltk.download()
```

- Chargement du texte à partir du technique web scraping

Récupérer des articles de Wikipedia (SCRAPING)

```
import bs4 as bs
import urllib.request
import re

scraped_data = urllib.request.urlopen('https://en.wikipedia.org/wiki/Natural_language_processing')
article = scraped_data.read()

parsed_article = bs.BeautifulSoup(article,'lxml')

paragraphs = parsed_article.find_all('p')

article_text = ""

for p in paragraphs:
    article_text += p.text
```

- Suppression des crochets et des espaces supplémentaires

```
# Removing Square Brackets and Extra Spaces
article_text = re.sub(r'\[[0-9]*\]', ' ', article_text)
article_text = re.sub(r'\s+', ' ', article_text)
```

✓ 0.7s

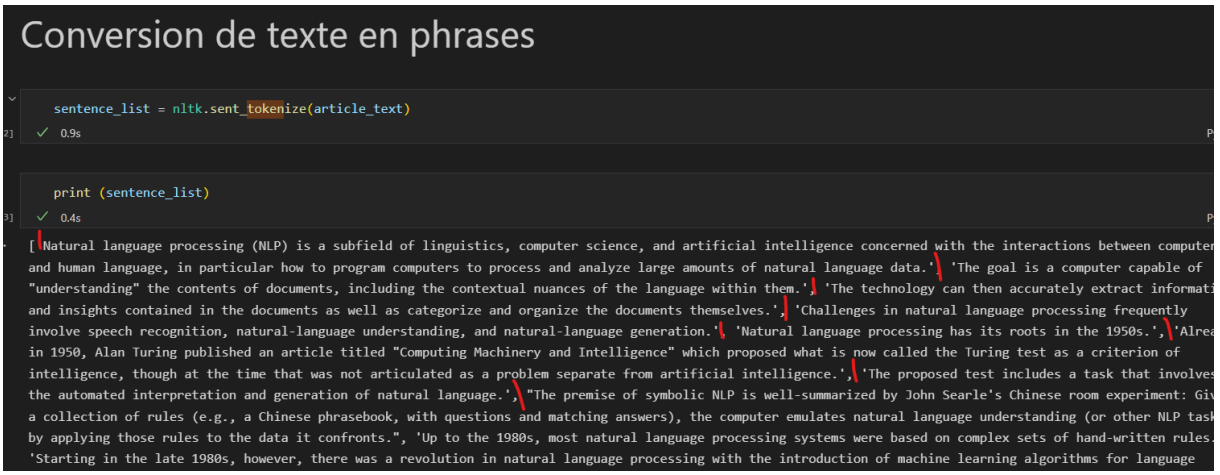
- Suppression des caractères spéciaux et des chiffres

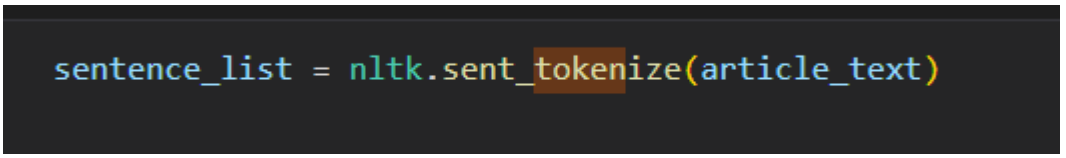
Preprocessing

```
# Removing special characters and digits
formatted_article_text = re.sub('[^a-zA-Z]', ' ', article_text )
formatted_article_text = re.sub(r'\s+', ' ', formatted_article_text)
```

✓ 0.7s

- Segmenter les phrases en mots tokenistion (fonction prédéfinie tokenize())





- Détecter les stopWords (mots vides) (fonction prédéfinie pour les
- stopWords) & Calculer pour chaque mot sa fréquence (sauvegarder les résultats dans une structure)

```
stopwords = nltk.corpus.stopwords.words('english')

word_frequencies = {}
for word in nltk.word_tokenize(formatted_article_text):
    if word not in stopwords:
        if word not in word_frequencies.keys():
            word_frequencies[word] = 1
        else:
            word_frequencies[word] += 1
```

✓ 0.8s

```
word_frequencies
```

✓ 0.6s

```
Output exceeds the size limit. Open the full output data in a text editor
{'Natural': 0.07142857142857142,
 'language': 1.0,
 'processing': 0.5714285714285714,
 'NLP': 0.6071428571428571,
 'subfield': 0.03571428571428571,
 'linguistics': 0.32142857142857145,
 'computer': 0.14285714285714285,
 'science': 0.10714285714285714,
 'artificial': 0.07142857142857142,
 'intelligence': 0.10714285714285714,
```

- Calculer les poids des phrases en fonction de la présence de mots

```
sentence_scores = {}
for sent in sentence_list:
    for word in nltk.word_tokenize(sent.lower()):
        if word in word_frequencies.keys():
            if len(sent.split(' ')) < 30:
                if sent not in sentence_scores.keys():
                    sentence_scores[sent] = word_frequencies[word]
                else:
                    sentence_scores[sent] += word_frequencies[word]
```

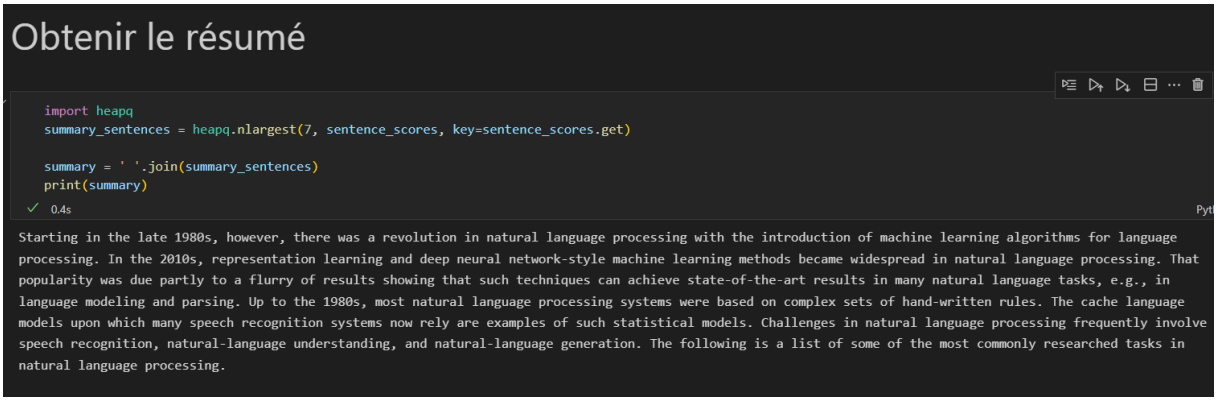
✓ 0.6s

```
sentence_scores
```

✓ 0.3s

```
Output exceeds the size limit. Open the full output data in a text editor
{'The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them.': 1.6785714285714286,
 'The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.':
0.6428571428571428,
 'Challenges in natural language processing frequently involve speech recognition, natural-language understanding, and natural-language generation.':
2.821428571428572,
 'Natural language processing has its roots in the 1950s.': 2.25,
 'The proposed test includes a task that involves the automated interpretation and generation of natural language.': 2.107142857142857,
 'Up to the 1980s, most natural language processing systems were based on complex sets of hand-written rules.': 3.071428571428572,
 'Starting in the late 1980s, however, there was a revolution in natural language processing with the introduction of machine learning algorithms for language
processing.': 5.214285714285714,
 "This was due to both the steady increase in computational power (see Moore's law) and the gradual lessening of the dominance of Chomskyan theories of linguisti
(e.g.": 0.8928571428571428,
 'transformational grammar), whose theoretical underpinnings discouraged the sort of corpus linguistics that underlies the machine-learning approach to language
```

- fréquents au sein de chaque phrase (sauvegarder les valeurs des poids)
- Trier les phrases selon leurs poids
- Réordonner les phrases selon leur ordre chronologique du texte original



III. Travail à faire

- Comme convenu lors de la séance du 19/10/22, vous êtes amenés à présenter, lors de la séance du 09/11/22, un programme Python qui réalisera le résumé d'un texte en utilisant plusieurs critères de sélection des phrases. Vous suivez globalement les étapes qui ont été données dans le chapitre "Résumé Automatique (Bis)".
- Le travail sera élaboré par monôme ou par binôme.
- Vous allez présenter vos travaux le Mercredi 09/11/22.
- Un compte-rendu comportant les différentes parties du programme Python que vous allez coder ainsi que des images écran illustratives de l'exécution doit être envoyé par mail au plus tard le 08/11/22 à 20h.

NB : Le compte rendu dans un fichier pdf. Dans ce compte rendu, vous mettez les étapes qu'assure votre système. Pour chaque étape, vous donnez une brève description et vous illustrez son résultat à travers une image écran.