Analyse textuelle et extraction d'informations – Année 1923

Introduction

L'objectif de ce travail est de **traiter automatiquement un corpus de textes anciens** (les petites annonces de Bruxelles en 1923) afin d'en **extraire des informations pertinentes** à l'aide d'outils de **traitement automatique du langage naturel (TALN)**.

Le projet vise à :

- Nettoyer et uniformiser les fichiers texte du corpus ;
- Générer un **nuage de mots** pour visualiser les termes les plus fréquents ;
- Extraire les **mots-clés principaux** avec la méthode YAKE ;
- Identifier les **entités nommées** (personnes, organisations, lieux) grâce à SpaCy ;
- Réaliser une **analyse de sentiment** sur un échantillon de phrases pertinentes.

2. Structure du code

Le code est organisé en plusieurs parties, chacune correspondant à une étape du traitement.

2.1 Chargement et préparation du corpus

Cette première partie lit tous les fichiers `.txt` correspondant à l'année choisie (ici 1923). Chaque fichier est ouvert, converti en minuscules, et ajouté dans une seule grande variable `texte_total`.

CHEMIN = "/Users/walazerelli/tac-25-26/data/txt"

ANNEE_CHOISIE = "1923"

Un compteur affiche le nombre de fichiers lus.

Résultat obtenu : 100 fichiers ont été correctement chargés.

2.2 Nettoyage du texte

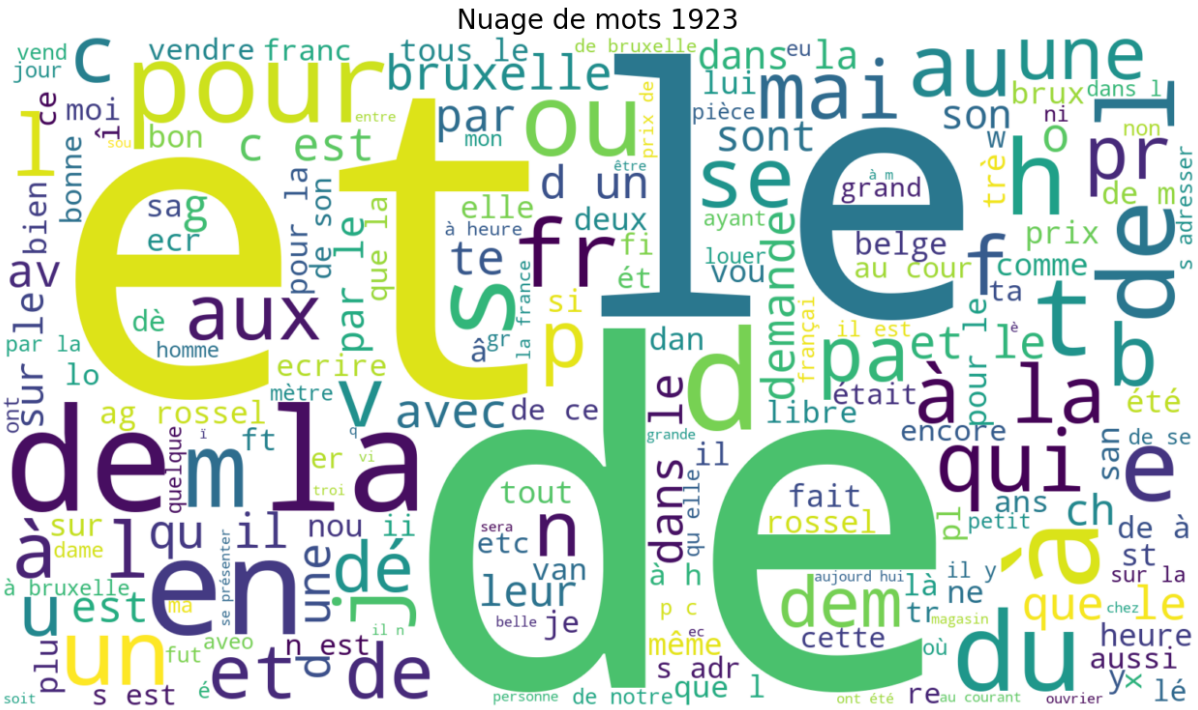
Le texte est ensuite **nettoyé** :

- Suppression des chiffres, symboles et ponctuation ;
- Uniformisation en minuscules ;
- Suppression des espaces multiples.

Le texte final propre (`texte_total_clean`) est prêt pour l'analyse.

2.3 Génération du nuage de mots

Un **nuage de mots** est généré avec la bibliothèque `WordCloud`.
Les mots fréquents comme “et”, “de”, “la” sont exclus grâce à une liste de *stopwords* enrichie.



Résultat obtenu :
L’image montre que les mots les plus fréquents sont *dem*, *pour*, *prix*, *bruxelles*, *bon*, *franc* et *jeune fille*.
Cela indique la présence de nombreuses annonces liées à l’emploi et aux échanges commerciaux.

2.4 Extraction de mots-clés (YAKE)

Le module `yake` est utilisé pour identifier automatiquement les expressions les plus caractéristiques du corpus.

Top 10 des mots-clés obtenus :

1. bons gages
2. dem jeune fille
3. rossel bruxelles
4. dem bon ouvrier
5. dem bons ouvriers
6. bas prix
7. demande bons ouvriers
8. neuve bruxelles
9. dem bon demi
10. haut prix

Ces expressions représentent les thèmes dominants du corpus.

2.5 Extraction d'entités nommées (SpaCy)

L'analyse avec **SpaCy** permet d'extraire les entités (personnes, lieux, organisations).
Les entités inutiles sont filtrées et normalisées pour éviter les doublons.

Résultats principaux :

Type	Exemple d'entités extraites
ORG	Reich, Ford, Fiat, Croix Rouge, Banque Nationale
GPE	Bruxelles, Congo Belge, Yougoslavie

2.6 Analyse de sentiment (TextBlob)

Les phrases les plus pertinentes ont été sélectionnées selon les mots-clés et les entités détectées.
Une analyse de **polarité** (positif/négatif) et de **subjectivité** a ensuite été appliquée.

Exemples de phrases analysées :

- « Cinéma nous cherchons d'urgence débutants des… » → Polarité : 0.20
- « Prémédité son crime car un témoin entendu... » → Polarité : 0.26
- « Bronchite, asthme, douleurs dans la poitrine… » → Polarité : 0.50

Les scores moyens restent faibles, car la majorité des textes sont informatifs et non émotionnels.

3. Résultats globaux

- **100 fichiers** lus et nettoyés
- **Nuage de mots** révélant les termes principaux : *dem*, *prix*, *bruxelles*
- **10 mots-clés majeurs** extraits automatiquement
- **Top 20 entités** dont *Reich*, *Ford*, *Fiat*, *Croix Rouge*
- **10 phrases analysées** pour la polarité et la subjectivité
- Les textes sont globalement **neutres** avec une très faible charge émotionnelle

4. Conclusion

Le code mis en place permet d'exécuter **toute la chaîne de traitement textuel automatique**, depuis la récupération du corpus jusqu'à la visualisation et l'analyse linguistique.
Chaque étape est automatisée, reproductible et donne des résultats cohérents avec la nature du corpus.

L'ensemble du travail offre donc une **vision complète du traitement d'un corpus textuel avec Python**, en combinant **nettoyage, visualisation, extraction d'information et analyse linguistique**.