Projet Knowledge Extraction - Partie A

Jacques Gastebois
Master 2 VMI - Université Paris Cité
IFLCE085 - Recherche et extraction sémantique

1 Introduction

Ce projet a pour dessein l'extraction de connaissances structurées à partir d'un corpus de textes non structurés. La première étape, cruciale, est le **preprocessing**, qui transforme les données brutes en un format exploitable. Dans ce rapport, je présenterai le corpus et la méthodologie de preprocessing mise en uvre. J'exposerai ensuite l'analyse statistique détaillée, incluant la distribution des **Part-of-Speech (POS)** et la vectorisation **TF-IDF**. Pour conclure, je justifierai mes choix techniques et leurs impacts sur la suite du projet.

1.1 Le Corpus

Le dataset (`data.csv`) est constitué de **2221 phrases** annotées pour la reconnaissance d'entités nommées (NER).

- **Format d'entrée** : CSV avec colonnes `id`, `words`, `ner_tags`, `text`.
- **Contenu** : Textes encyclopédiques et biographiques.
- **Annotations** : Tags BIO pour les entités Personnes, Lieux, Organisations, etc.

2 Méthodologie de Preprocessing

Contrairement à une approche classique de "Bag of Words", j'ai opté pour un **preprocessing enrichi** préservant la structure séquentielle, indispensable pour le NER.

2.1 Pipeline Mis en Place

J'ai développé le traitement en Python via un notebook Jupyter, selon les étapes suivantes :

1. **Nettoyage (`cleaned_text`)** :
   - Conversion en minuscules pour réduire la dimensionnalité.
   - Suppression des caractères spéciaux non alphanumériques.
   - Normalisation des espaces.
2. **Lemmatization (`lemmatized_text`)** :
   - Utilisation de la librairie **spaCy**.
   - Transformation des mots en leur forme canonique.
3. **Enrichissement du Dataset** :
   - J'ai ajouté les colonnes `cleaned_text` et `lemmatized_text` au fichier original.
   - **Sortie** : `data_preprocessed.csv`.

3 Analyse Statistique du Corpus

L'analyse que j'ai menée dans le notebook `PartieA_Analyse_Statistique.ipynb` a révélé les caractéristiques suivantes :

3.1 Volumétrie et Longueur

— **Nombre de phrases** : 2221.

— **Longueur moyenne** : ∼24 mots par phrase.

— **Impact du preprocessing** : Comme illustré par la Figure 2, la lemmatization réduit significativement la taille du vocabulaire (d'environ **15% à 25%**). Cela permet de "densifier" l'information en regroupant les variantes d'un même mot.

— **Réduction des outliers** : On observe sur la Figure 2 une diminution drastique de la longueur des phrases "outliers" (points situés au-dessus des moustaches). Cela s'explique par le nettoyage qui supprime les caractères spéciaux, la ponctuation excessive et les métadonnées parasites souvent responsables de ces longueurs aberrantes dans le texte brut.

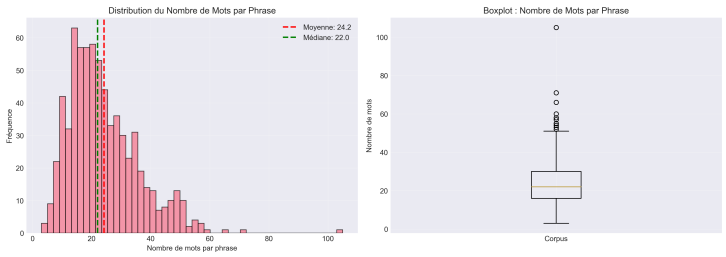

FIGURE 1 – Distribution des longueurs de phrases et de mots

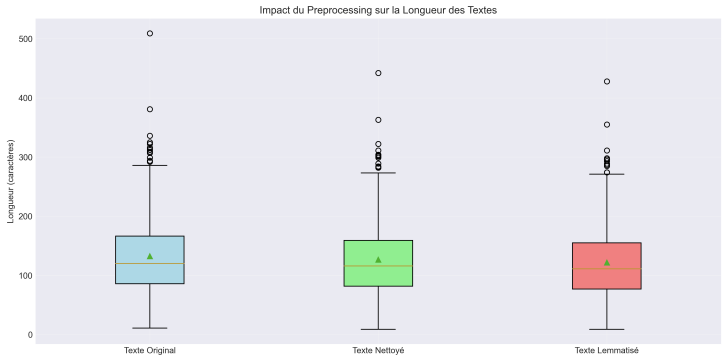

FIGURE 2 – Impact du preprocessing sur la taille du vocabulaire et la longueur des textes

3.2 Analyse Lexicale

L'analyse des mots les plus fréquents et du nuage de mots met en évidence les thématiques dominantes.

3.3 Analyse POS (Part-of-Speech)

Choix de la méthode : Utilisation du *tagger* probabiliste de la librairie **spaCy** (`en_core_web_sm`).

Avantages :

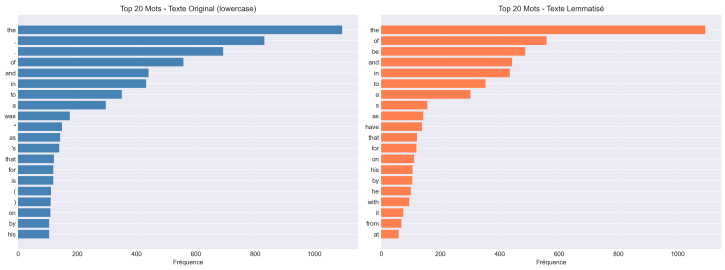

FIGURE 3 – Mots les plus fréquents (hors stop-words)

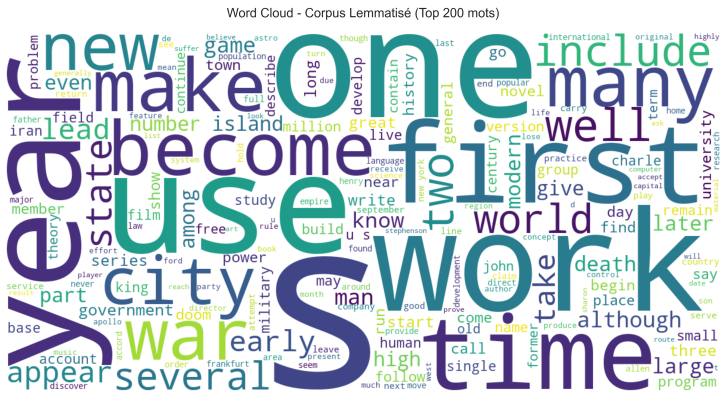

FIGURE 4 – Nuage de mots du corpus

— Rapidité d'exécution et robustesse sur l'anglais standard.

— Fournit des étiquettes universelles (UPOS) faciles à interpréter.

Inconvénients :

— Sensible aux erreurs de capitalisation (le passage en minuscules peut créer des ambiguïtés, ex : "Apple" vs "apple").

Sur 17 151 tokens, la distribution est la suivante :

Sur 17 151 tokens, la distribution est présentée en Figure 5. On note une prédominance des noms et noms propres (30.5%), typique de textes biographiques.

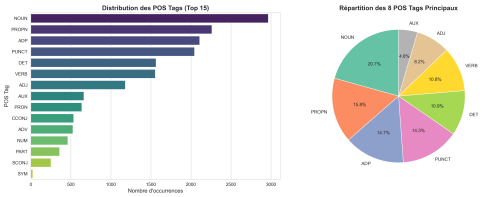

FIGURE 5 – Distribution des catégories POS (Top 15)

3.4 Vectorisation TF-IDF

Choix de la méthode : Vectorisation **TF-IDF** (Term Frequency-Inverse Document Frequency).

Avantages :

— Permet de pondérer les termes selon leur importance relative (pénalise les mots très fréquents et peu informatifs comme "the", "is").

— Réduit l'impact des "stop words" sans les supprimer brutalement.

Inconvénients :

— Perte de la sémantique séquentielle (approche "Bag of Words").

— Génère des matrices très creuses (haute dimensionnalité).

Exemple d'application du corpus : Considérons la phrase suivante issue du dataset (en-doc6361-sent4) : *"The study of logic led directly to the invention of the programmable digital electronic computer, based on the work of mathematician Alan Turing and others."*

Après prétraitement (nettoyage, lemmatisation), les termes comme *"programmable"*, *"digital"*, *"electronic"* et *"computer"* reçoivent des scores TF-IDF élevés car ils sont spécifiques à ce document par rapport au reste du corpus, tandis que des mots comme *"the"* ou *"of"* ont un score proche de zéro.

J'ai obtenu une matrice de **700 documents** × **2493 features**.

Caractéristiques :

— **Vocabulaire** : 2493 termes uniques.

— **Densité** : 0.77% (La matrice est très creuse, signifiant que chaque document ne contient qu'une infime fraction du vocabulaire global, ce qui est typique des données textuelles).

— **Paramètres** :

  — `max_df=0.8` : J'ai exclu les termes présents dans plus de 80% des documents. Ces mots sont considérés comme des "stop words" spécifiques au corpus (trop fréquents pour être discriminants).

  — `min_df=2` : J'ai ignoré les termes apparaissant dans moins de 2 documents. Cela permet d'éliminer le bruit (fautes de frappe, hapax legomena) et de réduire la dimensionnalité de la matrice sans perdre d'information généralisable.

  — `ngram_range=(1,2)` : J'ai inclus non seulement les mots uniques (unigrammes) mais aussi les paires de mots consécutifs (bigrammes). Cela permet de capturer des concepts composés (ex : "machine learning", "new york") qui ont un sens propre différent de la somme de leurs parties.

La figure 6 présente les termes aux scores TF-IDF les plus élevés.

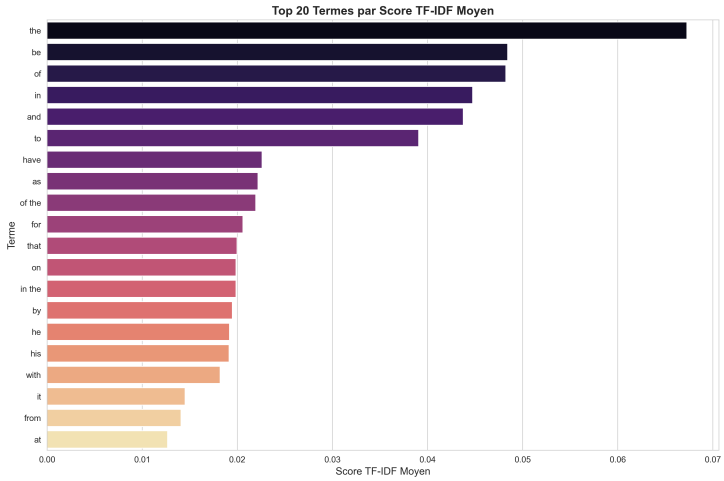

FIGURE 6 – Top 20 termes par score TF-IDF moyen

Interprétation : La Figure 6 montre que les termes ayant les scores TF-IDF les plus élevés restent des mots très courants de la langue anglaise ("the", "be", "of", "in"). Cela indique que ces termes sont présents dans une grande majorité des documents avec une fréquence élevée. Bien que le TF-IDF pénalise théoriquement les mots présents partout (IDF faible), leur fréquence brute (TF) est ici si importante qu'elle compense cette pénalité. Pour des tâches ultérieures plus fines, il serait pertinent d'envisager une liste de "stop words" plus agressive pour faire ressortir des termes plus spécifiques au domaine.

4 Analyse Critique des Choix

4.1 Choix 1 : Lemmatization vs Stemming

— **Choix** : Lemmatization (spaCy).

— **Avantages** : Préserve le sens sémantique, indispensable pour l'extraction de relations.

— **Inconvénients** : Coûteux en calcul, perte de certaines nuances flexionnelles.

4.2 Choix 2 : Lowercasing

— **Choix** : Tout mettre en minuscule.

— **Avantages** : Réduit la taille du vocabulaire.

— **Inconvénients** : **Perte de la capitalisation**, indice crucial pour le NER.

— **Mitigation** : J'ai conservé la colonne `text` originale pour pallier ce défaut.

4.3 Choix 3 : Conservation de la Structure Séquentielle

— **Choix** : Pas de vectorisation TF-IDF pour l'export final.

— **Avantages** : Indispensable pour le NER et l'extraction de relations.

— **Inconvénients** : Fichiers plus volumineux.

5 Impact sur la Suite du Projet

5.1 Impact sur la Partie II (NER)

Le preprocessing fournit une base propre. Toutefois, pour maximiser les résultats, il me faudra utiliser :

1. Le **texte original** pour la capitalisation.
2. Le **texte lemmatisé** pour les embeddings sémantiques.

5.2 Impact sur la Partie C (Extraction de Relations)

L'extraction de relations bénéficie de la lemmatization qui normalise les verbes et simplifie les arbres de dépendance.

6 Conclusion

En conclusion, je ne ferai qu'insister sur le fait que le preprocessing réalisé est un compromis entre réduction de bruit et préservation de l'information. Ce socle de données enrichi est robuste et adapté aux défis que j'aurai à relever par la suite : la reconnaissance fine d'entités et l'extraction de relations.