Université Numérique Cheikh Hamidou KANE

**POLE STN**

**Master 1 Big Data Analytics**

****

**Projet : Big Data**

**THEME :** **Automatisation de l'extraction de faits à partir de textes (Sujet, prédicat, objet)**

**Encadreur** :

Présenté Par

Ibou NGOM Fatoumata sow Mamadou Niane DIOP

Mai 2024

**SOMMAIRE**

[Introduction 0](#_Toc166717807)

[I. Problématique 0](#_Toc166717808)

[II. Description des données 2](#_Toc166717809)

[III. Prétraitement des données 2](#_Toc166717810)

[A.Le web scraping 2](#_Toc166717811)

[a. Nettoyage des données 2](#_Toc166717812)

[b. Normalisation des données 2](#_Toc166717813)

[c. Suppression des doublons 2](#_Toc166717814)

[d. Gestion des valeurs manquantes 2](#_Toc166717815)

[e. Extraction d'entités 3](#_Toc166717816)

[f. Lissage des données 3](#_Toc166717817)

[B. L'extraction de faits (sujets, prédicats, objets) 3](#_Toc166717818)

[a. Segmentation du texte 3](#_Toc166717819)

[b. Tokenisation : 3](#_Toc166717820)

[c. Lemmatisation ou racinisation 3](#_Toc166717821)

[d. Identification des entités nommées (NER) 3](#_Toc166717822)

[e. Analyse grammaticale 4](#_Toc166717823)

[f. Extraction de phrases clés 4](#_Toc166717824)

[IV. Etude technique du projet 4](#_Toc166717825)

[a. Le code pour l’extraction 5](#_Toc166717826)

[b. Le code pour les fake news 6](#_Toc166717827)

[Conclusion 7](#_Toc166717828)

[ANNEXE : 8](#_Toc166717829)

Introduction

L'extraction automatique de triplets sujet-prédicat-objet à partir de textes est une tâche cruciale mais complexe du traitement automatique du langage naturel. Par ailleurs dans le monde numérique d'aujourd'hui, où l'information abonde à une vitesse vertigineuse, extraire des faits clés à partir de textes est devenu un défi crucial. L'automatisation de cette tâche, en identifiant les sujets, prédicats et objets au sein des documents textuels, est au cœur de ce projet. Cette discipline du traitement automatique du langage naturel (TALN) a le potentiel de transformer la façon dont nous traitons et comprenons les vastes quantités de données textuelles disponibles.

L'objectif principal de ce projet est de développer un système sophistiqué capable d'extraire de manière efficace et précise les éléments essentiels - les sujets qui effectuent des actions (prédicats) sur des objets - à partir de textes variés et souvent complexes. Cette automatisation n'est pas seulement un défi technique, mais aussi une nécessité dans un monde où les données textuelles sont omniprésentes, des médias sociaux aux bases de données académiques.

En explorant une gamme de techniques allant des méthodes traditionnelles basées sur des règles aux approches modernes basées sur l'apprentissage automatique et les réseaux de neurones, ce projet vise à construire un système polyvalent et adaptable. Une attention particulière sera accordée à l'optimisation de la précision et de la vitesse d'extraction.

L'importance de cette automatisation va au-delà de la simple commodité. Dans des domaines tels que la recherche scientifique, l'analyse de marché, la veille concurrentielle et même l'assistance virtuelle, l'efficacité de l'extraction de faits à partir de textes peut faire la différence entre des décisions éclairées et des conjectures approximatives.

Ce projet représente donc une initiative cruciale dans le domaine du TALN, ouvrant la voie à une meilleure utilisation des ressources textuelles massives disponibles et à des avancées significatives dans divers domaines d'application. Dans les sections suivantes, nous explorerons les méthodes, les outils et les défis associés à cette entreprise passionnante.

1. Problématique

L'extraction automatique de triplets sujet-prédicat-objet est souvent un aspect crucial dans ce projet. Cette tâche peut être complexe en raison de plusieurs défis :

Ambiguïté linguistique :

Les langues naturelles sont souvent ambiguës, avec des constructions syntaxiques et des mots pouvant avoir plusieurs interprétations. Cela rend difficile la détermination précise des sujets, prédicats et objets dans certaines phrases.

Variation de la structure grammaticale :

Les phrases peuvent avoir différentes structures grammaticales en fonction de la langue, du style de rédaction et du contexte. Il est donc nécessaire de concevoir des systèmes d'extraction de triplets robustes qui puissent s'adapter à cette diversité.

Traitement des expressions référentielles :

Les pronoms, les références déictiques et les descriptions référentielles peuvent rendre difficile l'association des éléments dans une phrase avec les entités appropriées.

Sens contextuel :

L'extraction de triplets nécessite souvent la compréhension du sens global d'une phrase ou d'un paragraphe pour identifier correctement les relations entre les éléments. Cela peut impliquer la résolution de l'ambiguïté sémantique et la prise en compte du contexte.

Traitement des entités nommées :

Les noms propres et les entités nommées peuvent être importants dans l'extraction de triplets, mais ils peuvent également être difficiles à extraire et à associer aux bons prédicats et objets.

1. Description des données

Les données que nous utilisons sont issues du web scraping de plusieurs plateformes de presse sénégalaises telles que Senego, Sunugal24, Senegal7, Seneplus, entre autres. Nous avons réussi à extraire un total de 864 lignes comprenant les contenus des différents articles de presse, les noms des auteurs, ainsi que les dates de publication associées.

Ces données représentent une ressource précieuse pour diverses applications, notamment dans le domaine du machine learning, où elles pourraient être utilisées pour des tâches telles que la détection de fausses informations (fake news). Dans la partie consacrée à l'étude technique, nous allons mettre en œuvre une application pratique de ces données. Vous pouvez trouver le notebook correspondant dans la section dédiée.

1. Prétraitement des données
2. Le web scraping

Le prétraitement dans le cadre du web scraping est une étape cruciale visant à nettoyer et à préparer les données extraites pour qu'elles soient utilisables dans des analyses ultérieures. Voici quelques-unes des tâches typiques effectuées lors du prétraitement :

1. Nettoyage des données

Supprimer les balises HTML, les publicités, les scripts JavaScript et tout autre contenu non pertinent qui peut être inclus dans les pages web. Cela permet d'obtenir uniquement le contenu textuel des articles, des titres, des dates, etc.

1. Normalisation des données

Uniformiser le format des données, par exemple en convertissant toutes les dates dans un format standard, en mettant en minuscules tous les mots, en supprimant les espaces superflus, etc.

1. Suppression des doublons

Identifier et éliminer les entrées en double qui pourraient résulter de la structure du site web ou du processus de scraping lui-même.

1. Gestion des valeurs manquantes

Traiter les cas où certaines informations essentielles sont manquantes dans les données extraites. Cela peut impliquer de remplir les valeurs manquantes avec des estimations ou de supprimer les entrées concernées.

1. Extraction d'entités

Identifier et extraire des informations spécifiques telles que les noms d'auteurs, les URL des articles, les catégories thématiques, etc., en utilisant des techniques telles que la reconnaissance d'entités nommées (NER).

1. Lissage des données

Réduire le bruit dans les données en appliquant des techniques telles que la suppression des caractères spéciaux, la lemmatisation ou la racinisation des mots, afin de regrouper les termes similaires.

1. L'extraction de faits (sujets, prédicats, objets)

Dans le contexte de l'automatisation de l'extraction de faits à partir de textes, le prétraitement est crucial pour préparer les données textuelles en vue de l'extraction des faits, incluant les sujets, prédicats et objets. Son objectif est de rendre les données textuelles plus organisées, cohérentes et adaptées à l'analyse par des techniques d'extraction automatisée, comme l'analyse de dépendances syntaxiques ou les modèles de langage statistiques.

Voici un exemple d'étapes de prétraitement typiques :

1. Segmentation du texte

Diviser le texte en phrases ou en segments cohérents, ce qui facilite l'identification des relations entre les entités.

1. Tokenisation :

Diviser chaque phrase ou segment en mots ou en "tokens" individuels. Cela permet de traiter chaque mot comme une unité distincte pour l'analyse ultérieure.

1. Lemmatisation ou racinisation

Réduire les mots à leur forme de base (lemme) ou à leur racine, afin de normaliser le texte et de regrouper les variantes de mots similaires.

1. Identification des entités nommées (NER)

Utiliser des techniques de NER pour repérer et étiqueter les entités importantes dans le texte, telles que les noms de personnes, d'organisations, de lieux, etc. Ces entités peuvent être utilisées comme sujets ou objets dans l'extraction des faits.

1. Analyse grammaticale

Analyser la structure grammaticale des phrases pour identifier les relations syntaxiques entre les mots. Cela peut aider à déterminer les prédicats qui expriment les relations entre les sujets et les objets.

1. Extraction de phrases clés

Identifier les phrases les plus importantes ou les plus représentatives du texte, qui contiennent probablement les informations les plus pertinentes pour l'extraction des faits.

1. Etude technique du projet
2. Les étapes de réalisation technique

Dans cette quatrième partie de notre projet sur l'automatisation de l'extraction de faits à partir de textes (sujet, prédicat, objet), notre travail a commencé par la création d'un programme Python dédié à l'extraction automatique du contenu textuel à partir d'un site web. Ce code Python est conçu pour récupérer les données pertinentes, les nettoyer et les structurer afin de créer un ensemble de données exploitable, également appelé dataset.

Après avoir réussi l'extraction de plusieurs ensembles de données et avoir créé notre dataset, notre prochaine étape consistera à utiliser ces données pour développer une application de machine learning. En collaboration avec notre encadreur, nous avons décidé de nous concentrer sur un aspect crucial : la détection de fausses informations, également connues sous le terme de "fake news".

Pour ce faire, nous avons élaboré un deuxième code Python qui exploite les données extraites pour entraîner un modèle de machine learning capable de déterminer si une information donnée est potentiellement une fausse nouvelle. Cette tâche nécessite un processus de classification où le modèle apprend à distinguer entre les informations véridiques et les informations trompeuses en se basant sur les caractéristiques et les contextes des articles de presse.

Cette phase de notre projet vise à combiner les techniques d'extraction de données avec les capacités de traitement du langage naturel et de machine learning pour aborder un défi important : la lutte contre la désinformation en ligne.

Explication code

1. Le code pour l’extraction

Le code ci-dessus comprend plusieurs étapes pour l'automatisation de l'extraction de faits à partir de textes, en mettant l'accent sur la détection d'incohérences dans les articles. Ce code illustre une approche complète pour collecter des données textuelles à partir du web, les analyser linguistiquement, détecter les incohérences dans le contenu des articles, et créer un dataset équilibré pour l'entraînement de modèles de machine learning.

Voici une explication synthétique des principales parties du code :

1. **Installation des bibliothèques nécessaires** : Les bibliothèques requises telles que spaCy, NLTK et newspaper3k sont installées pour le traitement du langage naturel, le scraping web et d'autres opérations.
2. **Scraping des articles** : Le code commence par le scraping des articles à partir d'un site web donné, en utilisant la bibliothèque newspaper3k. Les données des articles sont récupérées, y compris le titre, le contenu, la date de publication et les auteurs.
3. **Analyse linguistique avec spaCy** : Les données textuelles sont analysées avec le modèle de langue française pré-entraîné de spaCy pour extraire des triplets sujet-verbe-objet à partir du contenu des articles.
4. **Détection d'incohérences** : Une fonction est définie pour détecter les incohérences dans le contenu des articles. Cela inclut la vérification de la grammaire, de l'orthographe et de la cohérence thématique du texte.
5. **Équilibrage du dataset** : Les articles sont étiquetés comme cohérents ou incohérents en fonction des résultats de la détection d'incohérences. Ensuite, un dataset équilibré est créé en sélectionnant un sous-ensemble aléatoire de données cohérentes pour égaler le nombre de données incohérentes.
6. **Sauvegarde du dataset** : Le nouveau dataset avec les étiquettes équilibrées est enregistré dans un fichier Excel.
7. Le code pour les fake news

Le code est une implémentation en Python pour détecter les fake news à partir de données textuelles en utilisant un modèle de classification Passive Aggressive Classifier. Ce code fournit une base solide pour détecter les fake news à partir de données textuelles en utilisant des techniques de machine learning. Il montre également comment prétraiter les données textuelles pour les rendre compatibles avec les algorithmes de classification.

Voici une explication synthétique des différentes parties du code :

1. **Importation des bibliothèques :** Les bibliothèques nécessaires pour le traitement des données (NumPy, Pandas, Matplotlib, Seaborn) et pour les techniques de machine learning (scikit-learn) sont importées.
2. **Téléchargement du fichier Excel** : L'utilisateur est invité à importer un fichier Excel contenant les données textuelles à analyser.
3. **Prétraitement des données** : Les données sont chargées depuis le fichier Excel et des étapes de prétraitement sont effectuées, telles que la suppression des colonnes inutiles, la vérification de la qualité des données, le nettoyage du texte, la lemmatisation, etc.
4. **Transformation des données textuelles en données numériques :** Les données textuelles sont transformées en vecteurs de caractéristiques utilisant la technique TF-IDF (Term Frequency-Inverse Document Frequency).
5. **Division des données en ensembles** **d'entraînement et de** **test** : Les données sont divisées en ensembles d'entraînement et de test pour évaluer les performances du modèle.
6. **Entraînement du modèle** : Un modèle de classification Passive Aggressive Classifier est entraîné sur les données d'entraînement.
7. **Évaluation du modèle** : Le modèle est évalué sur les données de test en calculant son score de précision et en affichant une matrice de confusion pour visualiser ses performances.

Conclusion

Ce projet d'automatisation de l'extraction de faits à partir de textes représentent une étape significative dans le domaine du traitement automatique du langage naturel (TALN). En développant un système capable d'identifier et d'extraire efficacement les sujets, prédicats et objets à partir de documents textuels, nous avons ouvert la voie à une utilisation plus efficace et pertinente des vastes quantités de données textuelles disponibles.

À travers ce projet, nous avons exploré diverses techniques, des méthodes traditionnelles aux approches modernes basées sur l'apprentissage automatique et les réseaux de neurones. Nous avons également relevé des défis importants, tels que la diversité linguistique et sémantique des textes, ainsi que la nécessité d'une précision et d'une adaptabilité accrues.

Les applications potentielles de cette automatisation sont vastes et variées, allant de la recherche scientifique à l'analyse de marché, en passant par la veille concurrentielle et l'assistance virtuelle. En permettant une extraction rapide et précise des faits clés à partir de textes, notre système contribue à accélérer les processus décisionnels et à améliorer la qualité des analyses dans ces domaines.

Cependant, malgré les progrès réalisés, il reste encore des défis à relever. L'amélioration continue de la précision, de la robustesse et de la capacité à traiter des textes dans différentes langues et contextes reste une priorité. De plus, l'évolution rapide de la technologie et des usages nécessitera une adaptation constante de notre système pour rester pertinent et efficace.

En définitive, ce projet représente une contribution significative à l'avancement du domaine du TALN et ouvre la voie à de nouvelles possibilités dans la manière dont nous traitons et comprenons les données textuelles. En combinant expertise technique et vision stratégique, nous sommes convaincus que cette automatisation aura un impact durable sur de nombreux aspects de notre société et de nos activités.

Ce projet nous a permis de travail en équipe et en distance ce qui constitue un atout dans le milieu professionnel.

ANNEXE :

Les liens utilisés dans ce projet :

* <https://github.com/abiek12/Fake-News-Detection-using-MachineLearning/blob/main/Fake_News_Detector-PA.ipynb>
* <https://thepythoncode.com/article/fake-news-classification-in-python>
* <https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/>
* <https://github.com/Athari22/Fake-News-Detection-Using-Python>
* <https://drive.google.com/drive/folders/1kWorVo4AyCGZLZ5_CjSyMr3QuCaShwLz?usp=drive_link>
* <https://www.seneweb.com/>
* <https://senego.com/>
* <https://www.senenews.com/>
* <https://www.seneplus.com/>
* <https://www.dakaractu.com/>
* <https://senegal7.com/>