Projet_fin_module

July 6, 2025

1 Projets de Fin de Module – Introduction au Traitement Automatique du Langage Naturel (NLP)

2 DIT– 2024/ 2025

2.1 Consignes générales

Chaque groupe d'étudiant sélectionnera un sujet parmi ceux proposés.
Le projet devra être réalisé en Python (Jupyter Notebook fortement recommandé).
Un rapport détaillé et un code proprement commenté sont attendus.
Une soutenance orale de 10 minutes conclura le module.

2.2 ## Date limite de dépot des sujets: le 14/07/ 2025

2.3 Sujet 1 : Classification de la polarité des avis clients

2.3.1 Contexte

Les avis en ligne représentent une source précieuse d'informations pour les entreprises et les consommateurs. Savoir automatiquement déterminer si un avis est positif ou négatif permet d'automatiser la veille de réputation.

2.3.2 Objectifs pédagogiques

- Appliquer le prétraitement de texte
- Mettre en œuvre des techniques de vectorisation
- Entraîner et évaluer des modèles de classification
- Interpréter et présenter des résultats

2.3.3 Description détaillée

Développez un système de classification automatique qui détermine si un avis client est positif ou négatif.
Vous utiliserez un jeu de données d'avis en français.
Vous devrez : - Nettoyer et prétraiter les textes (tokenisation, stop words, lemmatisation) - Explorer les données (statistiques descriptives, visualisation des mots fréquents, distribution des classes) - Transformer les textes en vecteurs numériques (TF-IDF, Bag-of-Words, embeddings) - Entraîner au moins deux modèles de classification (ex : logistic regression, SVM, Random Forest) - Comparer les performances des modèles (accuracy, F1-score, matrice de confusion) - Interpréter les résultats, identifier les points d'amélioration

2.3.4 Suggestions de jeux de données

- Allociné Reviews
- French Amazon Reviews

2.3.5 Ressources/outils recommandés

- Python (pandas, scikit-learn, nltk, spaCy)

2.3.6 Livrables attendus

- Notebook ou script Python documenté
- Rapport (structure conseillée : introduction, méthodologie, résultats, analyse critique, conclusion)
- Visualisations pertinentes (ex : nuages de mots, graphes de performance)
- (Optionnel) Application ou interface simple pour tester votre modèle

2.3.7 Pistes d'amélioration (bonus)

- Tester des embeddings plus avancés (Word2Vec, FastText, BERT)
- Proposer une visualisation interactive des résultats

2.3.8 Critères d'évaluation

- Qualité du nettoyage et du prétraitement
- Choix et justification des modèles
- Pertinence des analyses et visualisations
- Clarté du rapport et de la présentation orale

2.4 Sujet 2 : Reconnaissance d'entités nommées (NER) dans des articles de presse

2.4.1 Contexte

Les entités nommées (noms de personnes, lieux, organisations, etc.) sont omniprésentes dans les textes d'actualité. La NER est une tâche clé du NLP pour l'extraction d'informations.

2.4.2 Objectifs pédagogiques

- Comprendre l'annotation et l'utilisation de corpus NER
- Mettre en œuvre un pipeline d'extraction d'entités
- Évaluer un système NER
- Interpréter et visualiser les résultats

2.4.3 Description détaillée

Créez un système capable d'identifier automatiquement les entités nommées dans des articles de presse en français.
Vous devrez : - Charger le corpus annoté - Nettoyer et préparer les textes - Explorer les types et fréquences d'entités - Tester l'extraction avec des outils existants (spaCy, Stanza, HuggingFace) -

Évaluer les résultats (précision, rappel, F1-score) - Visualiser les entités extraites dans des exemples d'articles - (Optionnel) Expérimenter l'apprentissage supervisé ou le fine-tuning

2.4.4 Suggestions de jeux de données

- LeNER-Br (corpus NER français)
- WikiNER (French)

2.4.5 Ressources/outils recommandés

- spaCy (modèle fr_core_news_md ou lg)
- Stanza
- Transformers (HuggingFace)
- Tutoriels spaCy NER : Documentation

2.4.6 Livrables attendus

- Notebook Python documenté
- Rapport incluant : exploration du corpus, méthodologie, résultats, visualisations, difficultés rencontrées
- (Optionnel) Démo ou outil interactif de NER

2.4.7 Pistes d'amélioration (bonus)

- Adapter ou fine-tuner un modèle sur le corpus fourni
- Comparer différents outils ou modèles

2.4.8 Critères d'évaluation

- Qualité de l'exploration et du prétraitement
- Approche méthodologique et justification des choix
- Clarté et pertinence des visualisations
- Analyse critique des résultats

2.5 Sujet 3 : Résumé automatique de textes d'actualité

2.5.1 Contexte

Le résumé automatique permet de condenser un texte long en quelques phrases, utile pour la veille informationnelle ou la navigation rapide dans de grands volumes de données.

2.5.2 Objectifs pédagogiques

- Appréhender les approches extractives et abstractive du résumé
- Utiliser des modèles pré-entraînés pour le résumé
- Évaluer la qualité des résumés générés

2.5.3 Description détaillée

Mettez en place un système générant automatiquement des résumés d'articles d'actualité en français.
Vous devrez : - Préparer le jeu de données (texte, résumé de référence) - Analyser les caractéristiques des textes/résumés - Implémenter une méthode extractive (TextRank, sumy…) et tester une méthode abstractive (T5, mBART) - Comparer les approches (longueur, fidélité, lisibilité) - Évaluer les résultats (score ROUGE, évaluation humaine si possible)

2.5.4 Suggestions de jeux de données

- OrangeSum
- WikiLingua (français)

2.5.5 Ressources/outils recommandés

- sumy, spaCy, HuggingFace Transformers
- Tutoriel sur le résumé automatique : Hugging Face summarization

2.5.6 Livrables attendus

- Notebook Python documenté
- Rapport (analyse comparative des méthodes, exemples de résumés, analyse critique)
- (Optionnel) Application de résumé automatique

2.5.7 Pistes d'amélioration (bonus)

- Évaluer la robustesse sur différents types de textes
- Proposer une interface web simple

2.5.8 Critères d'évaluation

- Qualité des méthodes implémentées
- Pertinence de l'analyse comparative
- Interprétation critique des résultats

2.6 Sujet 4 : Détection automatique de spam dans les SMS ou emails

2.6.1 Contexte

Le tri automatique des messages indésirables est une application classique et toujours d'actualité du NLP.

2.6.2 Objectifs pédagogiques

- Appliquer le NLP à des textes courts et bruités
- Concevoir un classifieur supervisé
- Comprendre les enjeux de déséquilibre des classes

2.6.3 Description détaillée

Développez un système qui distingue automatiquement les messages légitimes des spams dans un jeu de SMS ou d'emails en français.
Vous devrez : - Nettoyer et préparer les messages - Analyser les caractéristiques du spam (longueur, mots fréquents, etc.) - Représenter les messages (TF-IDF, n-grams) - Entraîner plusieurs modèles (Naive Bayes, SVM…) - Gérer le déséquilibre des classes (stratégies de rééchantillonnage, métriques adaptées) - Évaluer et interpréter les performances

2.6.4 Suggestions de jeux de données

- French Spam SMS Dataset
- UCI SMS Spam Collection (à traduire ou adapter)

2.6.5 Ressources/outils recommandés

- scikit-learn, pandas, nltk
- Tutoriel : scikit-learn spam detection

2.6.6 Livrables attendus

- Code Python documenté
- Rapport structuré (exploration, méthodologie, résultats, analyse critique)
- (Optionnel) Interface simple de test

2.6.7 Pistes d'amélioration (bonus)

- Utiliser des embeddings ou modèles avancés
- Expérimenter sur d'autres jeux de données

2.6.8 Critères d'évaluation

- Qualité du traitement des données
- Justesse du choix et de l'évaluation des modèles
- Capacité à interpréter les résultats

2.7 Sujet 5 : Analyse de sujets (topic modeling) sur des tweets ou articles

2.7.1 Contexte

Le topic modeling permet d'identifier automatiquement les grands thèmes d'un corpus textuel, utile pour l'exploration, la veille ou la classification non supervisée.

2.7.2 Objectifs pédagogiques

- Appliquer le NLP à des textes courts ou informels
- Mettre en œuvre des algorithmes de topic modeling
- Analyser et visualiser les résultats

2.7.3 Description détaillée

Découvrez les grands thèmes d'un corpus de tweets ou d'articles en français à l'aide d'algorithmes de topic modeling (LDA, NMF…).
Vous devrez : - Nettoyer et prétraiter les textes (suppression des mentions, hashtags, URLs, lemmatisation) - Représenter les textes (Bag-of-Words, TF-IDF) - Former des modèles de topics (LDA, NMF, BERTopic…) - Visualiser les thèmes (nuages de mots, graphique de distribution) - Interpréter les résultats (associer les topics à des sujets concrets)

2.7.4 Suggestions de jeux de données

- Tweets en français
- Le Monde articles

2.7.5 Ressources/outils recommandés

- Gensim, scikit-learn, pyLDAvis, spaCy
- Tutoriel LDA : Gensim topic modeling

2.7.6 Livrables attendus

- Notebook Python complet et documenté
- Rapport (démarche, visualisations, analyse des thèmes trouvés)
- (Optionnel) Présentation orale ou interactive

2.7.7 Pistes d'amélioration (bonus)

- Expérimenter des méthodes récentes (BERTopic)
- Proposer une interface d'exploration des thèmes

2.7.8 Critères d'évaluation

- Qualité du prétraitement
- Pertinence de la modélisation et de la visualisation
- Clarté de l'interprétation des thèmes

N'oubliez pas :
- De bien commenter votre code
- De structurer vos rapports
- De citer vos sources et outils utilisés
- La créativité et l'autonomie seront valorisées !

Bon courage à tous !

[ ]:

[ ]: