Classification de Texte

Faculté de Sciences et Techniques
Université de Nouakchott

Name: Mohamed EL Moustapha EL ARBY
Student Number: C17596

Ce notebook contient la documentation du projet de classification de texte en 3 catégories (pos, neg, neutr) en utilisant un modèle SVM codé manuellement (sans scikit-learn).

Introduction

La classification de texte est une tâche fondamentale en traitement automatique des langues (TAL) qui consiste à attribuer automatiquement une étiquette à une entité textuelle, à partir d'un ensemble d'exemples étiquetés, à reconnaître des motifs linguistiques caractéristiques (mots, expressions, structures grammaticales) et à associer de nouveaux textes à une classe prédéterminée (par exemple : positif, négatif, neutre).

Dans ce projet, nous nous intéressons spécifiquement à la classification de textes portant sur l'impact des médias sociaux en Mauritanie, et plus particulièrement à la classification des textes issus des plateformes d'échange (comme Facebook). L'objectif principal est d'explorer comment les données circulant sur ces réseaux sociaux peuvent contribuer à améliorer la gouvernance et renforcer la relation Gouvernement-Citoyen (G2C) en Mauritanie.

Au cours de la première phase de recherche, nous avons collecté et annoté manuellement des données issues de Facebook, en nous concentrant sur les commentaires rédigés en dialecte HASSANIYA. Ces données ont servi à évaluer diverses techniques de classification de texte, notamment l'impact du traitement en continu (streaming) sur la classification des dialectes mauritaniens. Les résultats préliminaires ont été publiés dans une revue internationale indexée.

1 Objectif

1. Maîtriser l'implémentation d'un SVM linéaire "from scratch"

2. Mettre en place une chaîne de traitement de texte adaptée
- Nettoyage, tokenisation, suppression des stop-words (français + liste Hassaniya personnalisée), vectorisation (bag-of-words ou TF-IDF) et conversion en matrice dense.

3. création et exploiter un jeu de données
- Recueillir les avis issus de Facebook, les annoter en trois classes (pos, neg, neutr), et s'assurer de la qualité et de la répartition équilibrée des classes.

4. Évaluer les performances et interpréter les résultats
- Calculer précision globale du modèle.

2 Librairies utilisées connexes avec son axe de recherche

Nous utilisons les modules suivants :

1. NumPy
   - Manipulations matricielles, opérations vectorielles, calculs de produit scalaire et mise à jour des poids du SVM.
   - Axe recherche : permet d'implémenter facilement les formules de l'apprentissage supervisé (descente de gradient) sans détails d'optimisation externes.

2. Pandas
   - Chargement et exploration du jeu de données, DataFrame pour visualiser la distribution des classes et découpage train/test.
   - Axe recherche : facilite l'analyse exploratoire, notamment pour détecter des biais linguistiques propres au dialecte Hassaniya.

3. Scikit-learn (prétraitement et évaluation uniquement)
   - train_test_split pour séparer en ensembles d'entraînement et de test.
   - CountVectorizer (ou TfidfVectorizer) pour transformer les textes en matrice creuse (TF-IDF).
   - classification_report, confusion_matrix pour obtenir les métriques standard (précision, rappel, F1-score) et matrice de confusion.
   - Axe recherche: fournit des métriques comparables aux études publiées pour le dialecte mauritanien.

4. NLTK (pour stop-words)
   - Extraction de stop-words en extension pour les particules Hassaniya.
   - Axe recherche : dans le contexte Hassaniya, avons chargé la liste de stop-words aral'avons combinéeombiné avec la liste de stop-words par des particules locales.

3 Données exploitées

Nous avons utilisé notre propre dataset collecté par nous-mêmes, qui contient : 1. Nombre d'instances : 1851
2. Nombre de lignes : 1853
3. Nombre de classes : 3

- Positif
- Négatif
- Neutre

4. Liste des attributs :

- id : identifiant unique pour chaque commentaire
- annotation : indique si l'entrée a été annotée (1) ou non (0)
- created$_at$: $horodatageindiquantlemomentoùlecommentaireaétéstockétext$: $contenucompletducommentaire$

- annotation$_result$: $labeldesentimentattribué(positif, négatifouneutre)$

4 Résultats et interprétations

Après entraînement et évaluation du classifieur SVM "from scratch" sur les commentaires en dialecte Hassaniya, nous obtenons les résultats suivants :

Précision globale (accuracy) :

Le modèle atteint une précision d'environ 36% sur l'ensemble de test, ce qui signifie que plus d'un commentaire sur trois est correctement classé dans sa catégorie (positif, négatif ou neutre).

Interprétation des résultats :

La précision globale 36% reste modeste, mais dépasse le seuil de 35% fixé comme objectif minimal.

5 Conclusion

Ce mini-projet a permis :

- De comprendre en profondeur le fonctionnement d'un SVM linéaire et de l'implémenter sans dépendance à scikit-learn.
- De mettre en œuvre une chaîne de traitement texte de bout en bout (nettoyage, vectorisation, entraînement, évaluation).
- D'identifier les forces et limites d'un tel modèle dans un contexte de dialecte local.