ANALYSE DES SENTIMENTS DANS LA LANGUE ARABE

OUARAS Khelil Rafik
Département Informatique
Université D'Alger 01
kikoouaras@gmail.com

RESUME

Ce projet explore l'analyse des sentiments sur les médias sociaux en se concentrant sur les tweets en langue arabe. Nous avons mis en œuvre un système complet, comprenant des étapes de prétraitement telles que la translittération, la suppression de la ponctuation, des stop words, ... et l'utilisation de n-grams. Les résultats obtenus montrent que des choix judicieux dans ces étapes améliorent significativement les performances du système. Notre modèle, basé sur le classificateur CountVectorizer avec des caractéristiques de 3-grams, atteint un F1-score de 65,00% et une précision dépasse de 67.85%. Le MultinomialNB émerge comme le modèle le plus performant parmi ceux évalués.

I. INTRODUCTION

Dans un contexte de connectivité croissante, l'analyse des sentiments émerge comme un outil essentiel pour décoder les interactions humaines en ligne. En exploitant les vastes quantités de données textuelles générées sur les médias sociaux, les commentaires et les revues, ce projet vise à extraire des insights cruciaux sur les opinions et émotions des utilisateurs. Applicable à la surveillance de la réputation en ligne, l'évaluation des produits, et la détection des tendances émotionnelles, l'analyse des sentiments offre des perspectives décisives pour des décisions éclairées dans des domaines variés.

II. BASE DE DONNEES

Dans cette section, nous décrivons la base de données dédiée à la recherche de tweets pour l'Analyse des Sentiments dans la langue Arabe. Ensuite, nous présentons les étapes de prétraitement que nous avons appliquées pour nettoyer les textes bruts extraits de Twitter. La base de données publiquement disponible contient, pour chaque tweet : (i) l'identifiant de l'utilisateur, (ii) l'identifiant du tweet, et (iii) les sentiments multiples indiquant s'il s'agit d'un tweet positif, négatif ou neutre. La base de données comprend 15,548 tweets au total. Nous avons observé que 41.77% de la base de données (6,495 tweets) mentionnant des sentiments sont étiquetés comme "NEU" (Neutre), 40.51% de la base de données (6,298 tweets) mentionnant des sentiments sont étiquetés comme "NEG" (Négatif), et 17.72% de la base de données (2,755 tweets) mentionnant des sentiments sont étiquetés comme "POS" (Positif).

III. ARCHITECTURE DU SYSTEME

Dans notre système, nous avons appliqué quatre étapes de prétraitement. La première étape consiste en l'utilisation du module Aransia pour translittérer les tweets de la langue française en langue dialectique arabe, ainsi que la normalisation. Ensuite, dans la deuxième étape, nous avons utilisé plusieurs traitements, notamment la suppression de la ponctuation, des chiffres, et des stop words en arabe, en utilisant le package stop-words de Python. De plus, nous avons éliminé les URLs, les mentions, les hashtags, les mots réservés de Twitter et les mots d'une seule lettre. La troisième étape de nettoyage de données a également inclus la suppression des emojis, la gestion des espaces (pas plus d'une seule espace), et des caractères doubles dans les mots. Enfin, dans la quatrième étape, nous avons effectué le

stemming en utilisant Tashaphyne, un stemmer et un segmenteur de lumière arabe, pour réduire les mots à leur racine. En parallèle, dans cette approche basée sur l'apprentissage automatique, après le nettoyage des données, nous avons appliqué différentes techniques telles que CountVectorizer, TF-IDF vectoriser et n-grams avec des valeurs de n allant de 1 à 10. Trois processus de tokenization ont été effectués : mots, caractères, et caractères avec limite (considérant l'espace comme un caractère). La classification a été réalisée en utilisant les algorithmes MultinomialNB, RandomForestClassifier, SVM, et Decision Tree.

IV. EXPERIENCES ET RESULTATS

Comme évoqué dans la section précédente, notre système repose sur l'application de multiples combinaisons de nettoyage (étapes de prétraitement) en utilisant des n-grams et des tokenizers. Nous avons sélectionné les quatre modèles, comme indiqué dans le tableau 1. De plus, nous avons présenté dans le même tableau les performances moyennes de toutes les classes ("POS", "NEG", "NEU") pour l'ensemble de test (F1-score, Rappel et Précision).

Model	**Configuration**	**Précision (%)**	**Rappel (%)**	**F1-score (%)**
MultinomialNB	CountVectorizer, max_features = 8000	67.49	65.67	64.33
	CountVectorizer, max_features = 8000, 3-grams	***67.85***	***65.67***	***65.00***
	TfidfVectorizer, max_features = 8000	66.30	57.00	57.67
	TfidfVectorizer, max_features = 8000, 3-grams	67.01	59.00	59.67
Random Forest Classifier	CountVectorizer, max_features = 8000	64.98	59.33	60.00
SVM	CountVectorizer, max_features = 8000, 3-grams	62.06	59.00	59.33
Decision Tree	CountVectorizer, max_features = 8000	55.47	47.67	47.67

Tableau 01 : Performances du système en termes de précision, de rappel et de score F1 pour chaque modèle et leur hyperparamètre.

Il est important de souligner que la taille des n-grams à un impact significatif sur les performances du système. Nous avons réalisé des expériences avec des n-grams de 1 à 10, et il en ressort que le 3-grams présente la meilleure amélioration de la précision. Ces résultats ont été obtenus en utilisant la méthode de séparation des données avec un test_size de 0.2 et un random_state de 42 dans la fonction train_test_split. Les performances détaillées de chaque modèle, y compris MultinomialNB, Random Forest Classifier, SVM et Decision Tree, sont fournies dans le Tableau 01, où les différentes configurations avec CountVectorizer et TfidfVectorizer sont présentées avec leurs scores de précision, de rappel et de F1-score.

V. Conclusion

L'approche adoptée dans ce travail repose tout d'abord sur une série d'étapes de prétraitement appliquées au jeu de données des tweets fourni dans cette tâche, puis sur le classificateur CountVectorizer avec des caractéristiques de n-grams, en plus du module de tokenization. Nous avons démontré que des choix adéquats de combinaison d'étapes de prétraitement et de valeurs de n (n-grams) ont conduit à une amélioration des performances. En comparaison avec la performance moyenne de la tâche, notre système obtient un score F1 de 65,00%, avec une précision dépassant de 67,85%. Le modèle le plus performant s'avère être le MultinomialNB.