

Détection de dialecte et détection d'offensive dans le langage arabe

Réalisé par :

Mustapha Ahricha & Omar Boudellah

Encadré par :

Pr Jihad Zahir : Professeur, Département Informatique, FSSM, UCA

le 10/01/2024

Table de matière

Introduction :	1
1. Objectifs du projet.....	1
2. Les données utilisé.....	1
la réalisation de projet.....	3
1. Prétraitement des données.....	3
2. L'algorithme utilisé.....	3
3. Déploiement.....	6

Introduction :

1. Objectifs du projet

Le principal objectif de ce projet de traitement du langage naturel (NLP) est de développer deux modèles performants destinés à la détection de dialecte et à l'analyse de l'offensive dans des phrases en langue arabe. Ces modèles seront conçus pour répondre à des

besoins spécifiques en matière de compréhension et d'interprétation des textes, en se concentrant sur deux aspects clés de l'analyse linguistique.

Détection de Dialecte :

L'objectif premier de ce projet est de concevoir un modèle capable de détecter le dialecte utilisé dans une phrase en langue arabe. Compte tenu de la diversité linguistique dans le monde arabe, cette fonctionnalité est cruciale pour comprendre et interpréter correctement les messages textuels. Le modèle sera formé sur un ensemble de données diversifié, représentatif des différents dialectes arabes, afin d'assurer une généralisation efficace lors de l'application.

Analyse de l'Offensive :

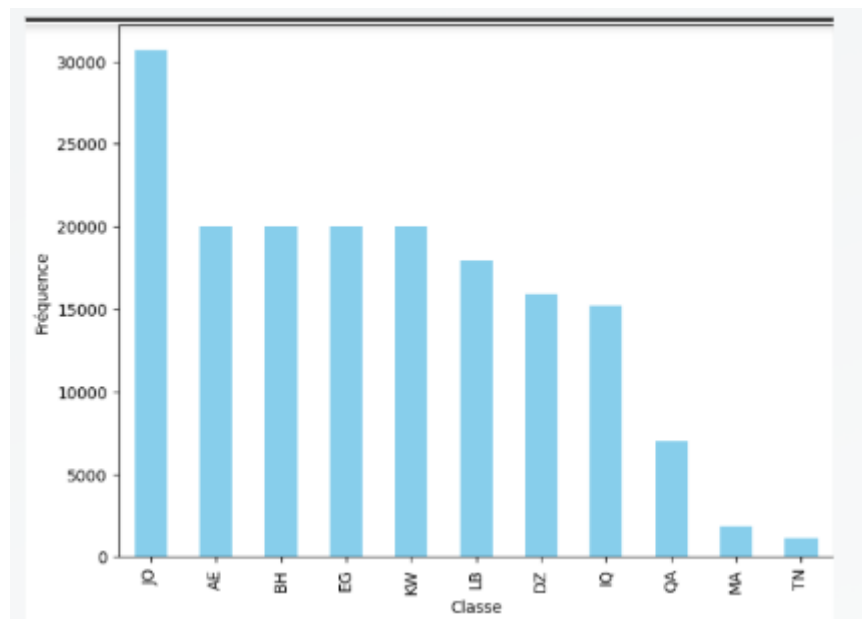
Le deuxième objectif majeur du projet est de créer un modèle spécialisé dans l'analyse de l'offensive dans des phrases arabes. Cette tâche implique la conception d'un modèle de classification capable de déterminer si une phrase est offensante ou non. La sensibilité culturelle et contextuelle inhérente à l'arabe rend cette tâche particulièrement complexe et nécessite une approche soigneusement adaptée pour garantir des résultats précis.

2. Les données utilisé

Pour la détection de dialecte, nous utilisons un fichier CSV contenant 365,719 lignes et deux colonnes.

	text	dialect
0	...ياخي المدرب اختاره والمدرب دخله والمدرب بارك ا	AE
1	...شو الي قاعد يجري فالنصر يا أخوه خسر المباراة س	AE
2	الي يبحث عن مشكلة الوصل راح يحصلها فالجولان	AE
3	...انا مش معترض على تغيير عامر الي دخل مكان عامر ا	AE
4	...تراجع مخيف في مستوى الحارس الكبير ماجد ناصر مش	AE

5 ligne de notre données



AE	20000
BH	20000
TN	20000
SY	20000
SA	20000
QA	20000
PL	20000
OM	20000
MA	20000
LY	20000
LB	20000
KW	20000
JO	20000
IQ	20000
EG	20000
DZ	20000
MSA	20000
SD	14481
YE	11238

les dialectes avec le nombre de ligne de données correspondant

	Tweet	class
0	اسغي يا شعب تونس تدعوا بالاسلام كفار الحمد لله ن	hate
1	قطع يد السارق توفرت الشروط شرط الحد الأدنى قيم	normal
2	تلوموش لطفي لعبدلي شرف	normal
3	مستغرب شعب يسمع تفاهة شانولي الدرجة الشعب تاف	normal
4	هههه غزلتني مافهمتش شمدخلها الموضوع تنتظر وحده	normal

la réalisation de projet

1. Prétraitement des données

Le prétraitement des données pour la détection de dialecte dans la création de notre modèle performant. Voici les étapes spécifiques du prétraitement des données que nous avons implémentées :

1. Nettoyage du Texte :

- Suppression des caractères spéciaux, des symboles de ponctuation, et des chiffres pour garantir une cohérence dans les données.

2. Tokenisation :

- Division des phrases en mots individuels (tokens) pour faciliter l'analyse du texte.

3. Lemmatisation :

- Réduction des mots à leur forme de base (lemme) pour standardiser le vocabulaire et améliorer la généralisation du modèle.

4. Suppression des Mots Vides :

- Élimination des mots couramment utilisés mais peu informatifs (mots vides) pour réduire la dimensionnalité du modèle.

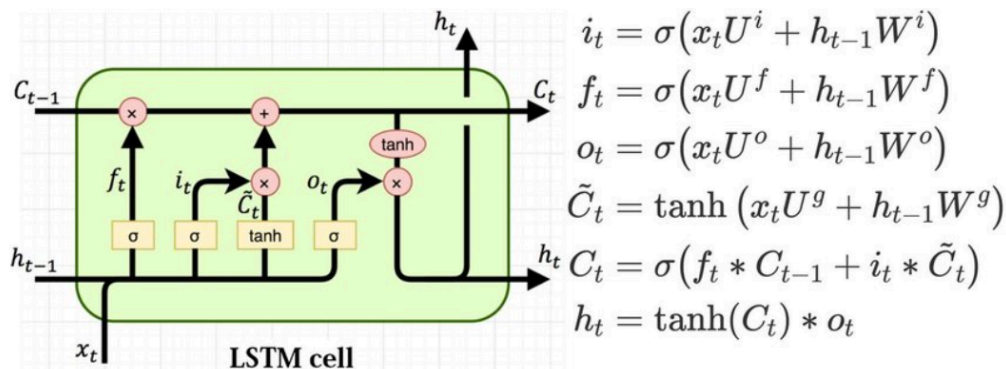
5. Encodage du Texte :

- Transformation des mots en représentations numériques pour être utilisées comme entrées pour le modèle de détection de dialecte.



2. L'algorithme utilisé

LSTM (Long Short-Term Memory) :



Structure de la cellule LSTM et équations qui décrivent les portes d'une cellule LSTM.

Les réseaux de neurones récurrents (RNN) sont puissants pour traiter des données séquentielles, mais ils peuvent être limités lorsqu'il s'agit de capturer des dépendances à long terme. L'algorithme LSTM, introduit pour remédier à ces limitations, est une variation des RNN conçue pour gérer efficacement les séquences de données avec des dépendances temporelles étendues.

Principales Caractéristiques :

1. Cellule Mémoire :

- L'élément clé de l'architecture LSTM est la cellule mémoire, qui permet de stocker et d'accéder à l'information sur une longue période. Cela permet au LSTM de capturer des dépendances temporelles à long terme.

2. Portes de l'Unité LSTM :

- Les unités LSTM sont équipées de trois portes :
- Porte d'Entrée (Input Gate) : Régule les informations qui entrent dans la cellule mémoire.
- Porte de Sortie (Output Gate) : Régule la sortie de la cellule mémoire vers la couche suivante.
- Porte Oubli (Forget Gate) : Contrôle la suppression sélective d'informations de la cellule mémoire.

3. Apprentissage des Dépendances Temporelles :

- Les LSTM sont capables d'apprendre des dépendances temporelles complexes grâce à la combinaison de ces portes, permettant ainsi la préservation ou l'oubli sélectif de l'information au fil du temps.

4. Gradient Flotant :

- Un problème fréquent dans les RNN est le problème du gradient qui disparaît ou explose. Les LSTM gèrent ce problème en utilisant des mécanismes de portes pour réguler le flux du gradient pendant la rétropropagation.

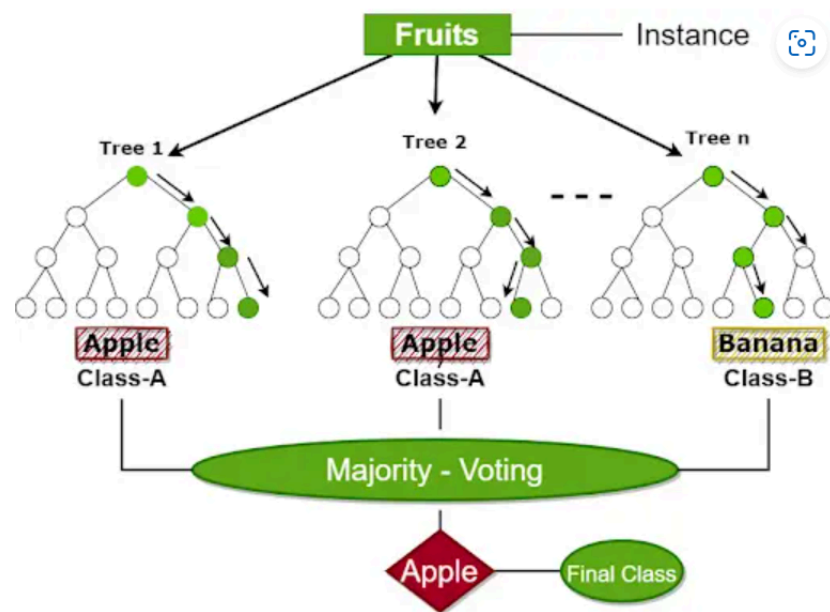
5. Applications :

- Les LSTM sont largement utilisés dans des domaines tels que la traduction automatique, la reconnaissance vocale, la génération de texte, et d'autres tâches liées à la séquence.

6. Bidirectionnels et Empilements :

- Les LSTM peuvent être utilisés dans des configurations bidirectionnelles pour capturer des informations des séquences dans les deux sens. Ils peuvent également être empilés pour former des architectures plus complexes.

Random Forest :



L'algorithme Random Forest est une technique d'apprentissage ensembliste, largement utilisée pour des tâches de classification et de régression. Il repose sur le concept de forêt d'arbres de décision, où chaque arbre individuel est construit de manière aléatoire. L'agrégation des prédictions de ces arbres conduit à un modèle robuste et performant.

Principales Caractéristiques :

1. Ensemble d'Arbres de Décision :

- Random Forest combine plusieurs arbres de décision pour former une forêt. Chaque arbre est construit de manière indépendante et de manière aléatoire.

2. Bagging (Bootstrap Aggregating) :

- Chaque arbre est entraîné sur un sous-ensemble aléatoire des données, choisi avec remplacement. Cela garantit la diversité des arbres et renforce la robustesse du modèle.

3. Variables Aléatoires :

- À chaque étape de la construction d'un arbre, un sous-ensemble aléatoire des caractéristiques est utilisé pour prendre des décisions. Cela évite la domination d'une caractéristique particulière dans la forêt.

4. Vote Majoritaire :

- Lorsqu'il s'agit de la classification, chaque arbre donne sa prédiction, et la classe finale est déterminée par un vote majoritaire. Pour la régression, la moyenne des prédictions est utilisée.

5. Réduction de l'Overfitting :

- La diversité des arbres et la sélection aléatoire de caractéristiques contribuent à réduire le surajustement, ce qui rend Random Forest robuste face à des données bruitées ou avec des caractéristiques redondantes.

6. Importance des Caractéristiques :

- Random Forest fournit une mesure de l'importance de chaque caractéristique, permettant ainsi d'identifier celles qui contribuent le plus à la prédiction.

La régression logistique :

La régression logistique est un algorithme d'apprentissage supervisé utilisé pour résoudre des problèmes de classification. Contrairement à son nom, la régression logistique est principalement utilisée pour la classification binaire, c'est-à-dire la prédiction de deux classes distinctes. Elle peut également être étendue à la classification multi-classes.

une description générale de l'algorithme de régression logistique :

1. Formulation du Problème : La régression logistique est utilisée pour résoudre des problèmes de classification où la variable cible (la variable que nous essayons de prédire) est binaire. Par exemple, cela peut inclure la prédiction de "Oui" ou "Non", "Fraude" ou "Non-fraude", etc.

2. Hypothèse : L'algorithme repose sur l'hypothèse que la relation entre les variables explicatives (features) et la variable cible peut être modélisée en utilisant une fonction logistique.

3. Fonction Logistique : La fonction logistique, également appelée fonction sigmoïde, est utilisée pour modéliser la probabilité qu'une observation appartienne à une classe particulière. La fonction logistique prend en entrée une combinaison linéaire des variables explicatives et produit une sortie entre 0 et 1.

La formule de la fonction logistique est la suivante :

$$\text{Logistic Function}(z) = \frac{1}{1 + e^{(-z)}}$$

Ici, z représente la combinaison linéaire des variables explicatives et des poids associés.

4. Entraînement : L'objectif de l'entraînement est d'ajuster les poids de manière à minimiser la divergence entre les prédictions de la régression logistique et les valeurs réelles de la variable cible. Cela est généralement accompli en utilisant une fonction de coût, telle que la log-vraisemblance négative (negative log-likelihood), qui mesure la différence entre les prédictions du modèle et les valeurs réelles.

5. Fonction de Coût : La fonction de coût mesure la performance du modèle en attribuant un coût à des prédictions incorrectes. Le but de l'entraînement est de minimiser cette fonction de coût.

6. Gradient Descent : L'optimisation des poids se fait souvent à l'aide d'algorithmes d'optimisation, tels que la descente de gradient (gradient descent). L'idée est d'ajuster progressivement les poids dans la direction qui minimise la fonction de coût.

7. Prédiction : Une fois le modèle entraîné, il peut être utilisé pour faire des prédictions sur de nouvelles données. La probabilité de l'appartenance à la classe positive est calculée à l'aide de la fonction logistique, et une décision de classification est prise en fonction d'un seuil (généralement 0.5).

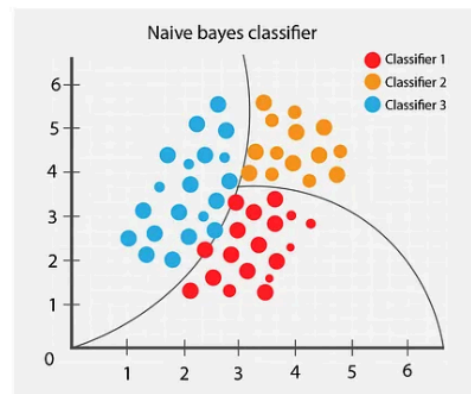
Naive Bayes :

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$



L'algorithme Naive Bayes est un algorithme d'apprentissage supervisé basé sur le théorème de Bayes. Il est particulièrement utilisé pour les problèmes de classification et est connu pour sa simplicité, sa rapidité d'entraînement et ses performances souvent surprenantes malgré ses hypothèses "naïves". L'approche "naïve" provient de l'hypothèse selon laquelle les caractéristiques (features) utilisées pour décrire les observations sont conditionnellement indépendantes, ce qui simplifie le calcul des probabilités.

une description générale de l'algorithme Naive Bayes :

1. Formulation du Problème : L'algorithme Naive Bayes est utilisé pour résoudre des problèmes de classification où l'objectif est de prédire la classe d'une observation en fonction de ses caractéristiques. Il est largement utilisé dans le domaine du traitement du langage naturel (NLP) pour la classification de texte.

2. Hypothèse Naïve : L'hypothèse naïve est que toutes les caractéristiques utilisées pour décrire une observation sont indépendantes les unes des autres, étant donné la classe de l'observation. Cette hypothèse simplifie les calculs en permettant de traiter chaque caractéristique de manière isolée.

3. Théorème de Bayes : L'algorithme Naive Bayes utilise le théorème de Bayes pour estimer les probabilités conditionnelles des classes étant donné les caractéristiques. Le théorème de Bayes s'exprime comme suit :

$$P(C | X) = P(X | C) P(C) / P(X)$$

- $P(C | X)$: Probabilité de la classe (C) étant donné les caractéristiques (X).
- $P(X | C)$: Probabilité des caractéristiques (X) étant donné la classe (C).
- $P(C)$: Probabilité a priori de la classe (C).
- $P(X)$: Probabilité marginale des caractéristiques (X).

4. Estimation des Probabilités : Pour entraîner le modèle, il faut estimer les probabilités $P(X | C)$ et $P(C)$ à partir des données d'entraînement.

- Estimation de $P(X | C)$: Pour les caractéristiques discrètes, cela peut être fait en comptant le nombre d'occurrences de chaque combinaison de caractéristiques et de classes. Pour les caractéristiques continues, on peut utiliser des modèles de distribution (par exemple, la distribution normale).

- Estimation de $P(C)$: La probabilité a priori de chaque classe peut être estimée en comptant la fréquence des occurrences de chaque classe dans les données d'entraînement.

5. Prédiction : Une fois le modèle entraîné, il peut être utilisé pour prédire la classe d'une nouvelle observation en utilisant le théorème de Bayes pour calculer les probabilités conditionnelles et en choisissant la classe ayant la probabilité la plus élevée.

6. Types de Naive Bayes : Il existe plusieurs variantes de l'algorithme Naive Bayes en fonction de la nature des caractéristiques et des distributions. Les trois types les plus courants sont le Naive Bayes multinomial (pour les caractéristiques discrètes), le Naive Bayes gaussien (pour les caractéristiques continues suivant une distribution normale), et le Naive Bayes Bernoulli (pour les caractéristiques binaires).

L'algorithme Naive Bayes est souvent utilisé dans des applications telles que la classification de documents, la détection de spam, et d'autres tâches de classification où les caractéristiques sont relativement indépendantes. Malgré ses simplifications, il peut fournir des résultats compétitifs dans de nombreuses situations.

3. Déploiement

On a créé une interface par l'utilisation de streamlit. Cette interface permet de faire un test de l'offensive dans les textes arabes et permet aussi de classer le dialectal arabe.

Alors comme le random forest est donné le meilleur précision (accuracy) pour le test d'offensive Alors on va déployer ce modèle pour le test d'offensive.

Et pour le dialect on a trouver que LSTM après l'ajustement des paramètres que ce modèle est donné des bonnes classification de dialecte alors on va déployer ce modèle pour le test de dialecte.

Le test de dialect : Voici l'interface pour détecter le dialect dans le langage arabe, ici un interface pour entrer un texte ou importer un fichier csv

The image displays two screenshots of a web application interface. The top screenshot shows the initial state where the 'Choisissez un projet' dropdown is open, listing 'Test dialecte' and 'Test d'offensive'. The main content area is titled 'Test de dialecte et test d'offensive' and includes a welcome message, a text input field for dialect testing, a file upload section for CSV files (with a 'Browse files' button), and a button to 'Afficher les prédictions pour le test de dialecte'. The bottom screenshot shows the same interface after a CSV file named 'output1 (1).csv' (11.3MB) has been uploaded. The file is listed below the upload section, and the 'Afficher les prédictions' button is now visible. The prediction results section at the bottom shows the word 'Egypt'.

Exemples de test le dialect un par entrer un texte et le deuxième par importer un csv :

Choisissez un projet

Test dialecte

Test de dialecte et test d'offensive

Bienvenue dans le test de dialecte!

Entrez du texte pour le test de dialecte

فأش كتوصل 4 د الليل وبالي ماتعشيش

Choisissez un fichier CSV pour le test de dialecte


Drag and drop file here
Limit 200MB per file

Browse files

Afficher les prédictions pour le test de dialecte

Prédictions pour le test de dialecte

Morocco

Le test d'offensive : voici l'interface et un exemple d'une phrase contient l'offensive

Choisissez un projet

Test d'offensive


Test de dialecte et test d'offensive

Bienvenue dans le test d'offensive!

Entrez du texte pour le test d'offensive

تفو عطفك يا الكتب و الله انت ماتسبش الخير لي درت ففقد انسان نكاهه و حمار

Choisissez un fichier CSV pour le test d'offensive


Drag and drop file here
Limit 200MB per file

Browse files

Afficher les résultats pour le test d'offensive

Résultats pour le test d'offensive

Ce texte contient l'offensive

Autre phrase de test mais cette fois une phrase normal

Choisissez un projet

Test d'offensive

Test de dialecte et test d'offensive

Bienvenue dans le test d'offensive!

Entrez du texte pour le test d'offensive

مصطفى تاه تاه مسكين أين اختفى هذا الفنان لماذا لم يعد يشارك في التمثيل ولو انه فنان مسرحي الأول في زمن الإبداع العراقي

Choisissez un fichier CSV pour le test d'offensive


Drag and drop file here
Limit 200MB per file

Browse files

Afficher les résultats pour le test d'offensive

Résultats pour le test d'offensive

Ce texte est normal