Rapport Machine Learning – Reconnaissance des émotions dans un texte

Table des matières

[Introduction 1](#_Toc477092978)

[Données d’apprentissage 1](#_Toc477092979)

[Déroulement de l’application 1](#_Toc477092980)

[Nettoyage des donnée 1](#_Toc477092981)

[Recherche des données adéquates 1](#_Toc477092982)

[Model d’apprentissage 1](#_Toc477092983)

[Apprentissage continue 1](#_Toc477092984)

[Manuel 1](#_Toc477092985)

[Automatique 1](#_Toc477092986)

[Analyse et résultat 1](#_Toc477092987)

# Introduction

L’application aura pour but d’identifier les émotions que transcrit une phrase ou un texte. Pour cela, le machine learning sera utilisé pour pouvoir prédire les émotions. Les émotions qui seront identifiées sont listées ci-dessous :

* Tranquilité
* Surprise
* Joie
* Tristesse
* Dégoût
* Colére
* Fureur
* Peur
* Terreur
* Coupure avec ses émotions

Un apprentissage automatique et manuel est aussi introduit dans l’application pour un apprentissage continu.

# Données d’apprentissage

L’application utilise deux tables de données :

* Une table référençant par un id les différentes émotions
* Une table avec les mots et expression lié par un id d’émotion

Ces données ont été récupérées sur internet, dans plusieurs sites. Des listes de mots ou d’expressions liées par des émotions y sont référencées.

# Déroulement de l’application

L’application intègre un menu qui permet :

* De prédire une émotion pour un texte.
* D’ajouter un mot ou une expression liée à une émotion dans la table d’apprentissage.
* De quitter l’application.

## Nettoyage des donnée

La première étape de l’application est le nettoyage de donnée. C’est lui qui permet en grosse partie l’optimisation d’une application de prédiction.

Dans un premier temps, le texte est découpé en différents qui lui composent dans un tableau. Par la suite, nous enlevons chaque ponctuation et chaque « stopword » (pronoms, mot de liaison, etc…) qui n’ont pas de sens à proprement parler dans un texte ou phrase. Ensuite, nous appliquons un « stemming » sur chaque mot du tableau. Il permet d’avoir seulement les racines des mots (il permet un meilleur moyen de recherché les donnée adéquate qui j’expliquerais par la suite)

Nous utilisons la librairie NLTK qui regroupe tout un lot de fonction qui permet l’analyse de texte et ainsi le nettoyage de donnée.

## Recherche des données adéquates

La recherche de donnés se fait en deux temps : pour les mots puis par expressions.

Pour choisir les données adéquates, je regarde dans la base de données de mes mots et expressions si je les retrouve dans le texte. Pour cela, le stemming est important. En l’appliquant, sur chaque mot du texte et pour chaque mot de la BDD. Je peux faire une comparaison en trouvant les mots de la même famille.

Pour les expressions, je fais la même chose sauf qu’à la fin du nettoyage de données, je remets en texte la table des mots nettoyés (et aussi les expressions que je traite) et j’applique une fonction « find() » qui me permet de retrouvé dans un texte une expression.

Par la suite, je stock dans une table tous les mots ou expressions que j’ai trouvé. Ils seront par la suite traités dans un modèle d’apprentissage pour prédire une émotion.

## Model d’apprentissage et résultat

Pour le model d’apprentissage, l’application utilise  « nltk.NaiveBayesClassifier » de la librairie NLTK. C’est un modèle de classification adapté pour les mots. Pour sa création, je lui mets en entrés ma table d’apprentissage. Il va par la suite créer un modèle où il aura classifié chaque mot et expression par émotion.

Pour chaque mot et expression que l’application a trouvé dans le texte, il va lancer une prédiction. J’aurais alors une table avec la liste des émotions pour chaque mot et expression.

S’il y a plusieurs émotions différentes, le programme va choisir l’émotion avec la plus grande occurrence. Nous avons donc à la fin une émotion prédite pour le texte à analyser.

# Apprentissage continue

## Manuel

## Automatique

# Analyse et résultat