

Extraction de connaissances avancée *“Analyse d’opinion”*

Carbonnel Jessie Nguyen Daniel Pibre Lionel

Université de Montpellier 2

18 Décembre 2014

Sommaire

- 1 Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
- 4 Visualisation
- 5 Classification
- 6 Conclusion et perspectives

Sommaire

- 1 Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
- 4 Visualisation
- 5 Classification
- 6 Conclusion et perspectives

Introduction

Sujet : Classification des opinions sur les commentaires des applications de Google Play Store.

Problématique : Prédire la note que l'utilisateur va donner à une application à partir de son commentaire.

Sommaire

- 1 Introduction
- 2 Constitution du corpus**
- 3 Prétraitement et génération des fichiers ARFF
- 4 Visualisation
- 5 Classification
- 6 Conclusion et perspectives

Structure des données récupérées

NomApplication:Ebook et PDF Reader

IdApplication:books.ebook.pdf.reader

CategorieApplication:Livres et références

NoteApplication:4,3 NombreVotants:43 379

TitreCommentaire:Ebook Pelerin
Commentaire: Super installation, ai acheté un ebook chez Bayard.

Suis pas déçu. DateCommentaire:26 juillet 2014

NoteCommentaire:5

Sommaire

- 1 Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
 - Prétraitement
 - Génération des fichiers ARFF
- 4 Visualisation
- 5 Classification

TreeTagger

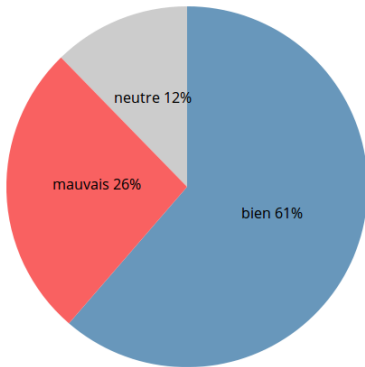
Utilisation de TreeTagger afin d'avoir la classe grammaticale des mots ainsi que leur forme lemmatisée.

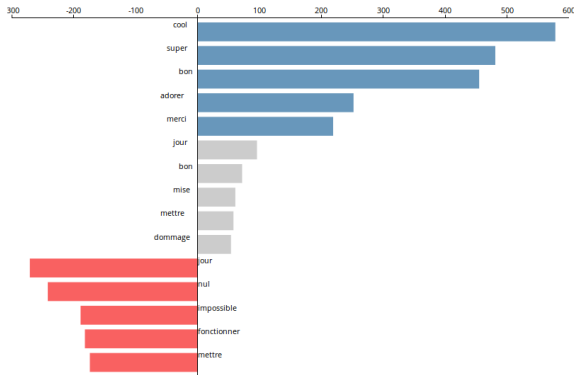
Structure de sortie de TreeTagger

Mot	Classe grammaticale	Mot lemmatisé
dès	PRP	dès
que	KON	que
je	PRO :PER	je
lance	VER :pres	lancer
l'	DET :ART	le
application	NOM	application
j'	PRO :PER	je
adore	VER :pres	adorer
cyprien	ADJ	cyprien
...

Sommaire

- 1 Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
- 4 Visualisation**
 - Camembert
 - Histogramme
- 5 Classification





Sommaire

- 1 Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
- 4 Visualisation
- 5 Classification**
- 6 Conclusion et perspectives

Algorithmes utilisés

- NaiveBayes : probabiliste
- J48 : arbre de décision
- JRip : règles d'association
- SMO : machine à vecteurs de support
- IBk : K plus proches voisins

Les problèmes liés au corpus

- Certains commentaires sont écrits en anglais
- Fautes d'orthographe et de frappe

Fautes d'orthographe et de frappe

“Je kiff grave car jadore cyprien et jaimefai lui poser un question :
est ce que tu connais squeezie (ca c oui c sur) norman kihouu tal
blackm”

Commentaire en anglais

“It keeps loosing my books , I have to re-download them every day”

Résultats des instances correctments classifiées

SMO : 73,6%

J48 : 69,3%

NaiveBayes : 67,8%

JRip : 67,7%

IBk : 64,4%

Analyses des résultats

L'algorithme de classification SMO est le plus robuste.
Les autres algorithmes sont plus sensibles au bruit et aux fautes d'orthographe.

Sommaire

- 1 Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
- 4 Visualisation
- 5 Classification
- 6 Conclusion et perspectives**

