Introduction Constitution du corpus Prétraitement et génération des fichiers ARFF Classification Conclusion et perspectives

Extraction de connaissances avancée "Analyse d'opinion"

Nguyen Daniel Pibre Lionel Carbonnel Jessie

Université de Montpellier 2

18 Décembre 2014



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



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



Introduction Constitution du corpus Prétraitement et génération des fichiers ARFF Classification 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.



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



AVIS

Lex Bee *****

Nice free PDF reader This PDF reader really has got fine assets. I specially appreciate the maintained preset

**1 10946

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1 140

**1



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



- Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
 - Prétraitement
 - Génération des fichiers ARFF
- Wisualisation
- Classification



Prétraitement

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





Génération

Programmation d'un parser en java.

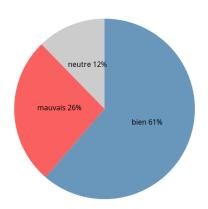
Quatre fichiers de sortie :

- Texte Brut
- Texte Brut Lemmatisé
- Texte Nettoyé
- Texte Nettoyé Lemmatisé
- Les quatre mêmes fichiers mais en corrigeant les mots adorer, aimer et kiffer

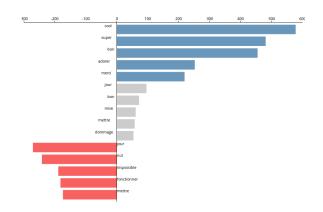


- Introduction
- 2 Constitution du corpus
- 3 Prétraitement et génération des fichiers ARFF
- 4 Visualisation
 - Camembert
 - Histogramme
- Classification











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



Algorithmes utilisés

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



Méthodes utilisées

- Utilisation de la représentation sac de mots
- Première exécution avec la présence des mots (représentation binaire)
- Deuxième exécution avec l'occurence des mots (représentation fréquentiste)



Les problèmes liés au corpus

- Certains commentaires sont écrits en anglais
- Fautes d'orthographe et de frappe
- ⇒ Les mots avec des fautes ou en anglais ne sont pas reconnus par TreeTagger



Introduction
Constitution du corpus
Prétraitement et génération des fichiers ARFF
Visualisation
Classification
Conclusion et perspectives

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

Texte Brut Texte Brut Lemmatisé

SMO: 73,1% SMO: 73.6%

J48: 67.8% J48: 68.9%

IBk: 65,6% IBk: 65.9%

JRip: 64,8% JRip: 67,8%

NaiveBayes: 64,2% NaiveBayes: 65,9%



Analyses des résultats

- L'algorithme SMO est le plus robuste.
- Text brut ⇒ Les mêmes mots sont écrits différemment
- Texte brut lemmatisé ⇒ Les mots non reconnus disparaissent
- Texte lemmatisé ⇒ Certains commentaires peuvent être vides



Résultats des instances correctments classifiées sur le texte corrigé

SMO: 73,2%

J48: 67.8%

IBk: 65.6%

JRip: 65,2%

NaiveBayes: 64,3%

Texte corrigé ⇒ Meilleurs résultats



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



Introduction Constitution du corpus Prétraitement et génération des fichiers ARFF Visualisation Classification Conclusion et perspectives

Conclusion

- Suivre un processus de fouille de données de A à Z
- Voir les différents résultats suivant le corpus
- L'impact des fautes d'orthographe sur les résultats



Perspectives

- Utiliser des outils de TALN pour corriger le corpus et traduire les commentaires écrits en anglais
- Faire de nouvelles visualisations sur les résultats obtenus
- Tester d'autres algorithmes

