

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

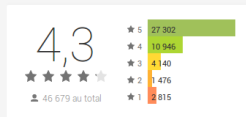
Sujet : Classification des opinions sur les commentaires des applications de Google Play Store.

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

Avis



Lex Bee ★★★★★

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



Mado Line ★★★★★

Y a mieux Trop compliqué pour chercher son fichier, ne prend pas en charge .epub et ne distribue pas



Frédéric Galet ★★★★★

Pas mal du tout Manque juste une prise en charge automatique du format du bouquin et de l'interface de lecture



Harouna Hassane ★★★★★

Excellent pour être gratuit. Reste à automatiser un paramétrage à l'ensemble des livres.

Donner votre avis



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
...

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

Commentaire étiqueté (1/2)

```
17 TitreCommentaire    NOM <unknown>
18 :    PUN :
19 Application NOM application
20 très    ADV très
21 bien    ADV bien
22 Commentaire ADJ <unknown>
23 :    PUN :
24 Application NOM application
25 magnifique ADJ magnifique
26 DateCommentaire:19 ADJ <unknown>
27 septembre    NOM septembre
28 2014    NUM @card@
29 NoteCommentaire:5 ABR <unknown>
```

Commentaire étiqueté (2/2)

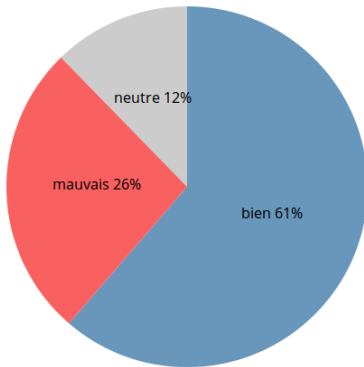
```
17 TitreCommentaire    NOM <unknown>
18 :    PUN :
19 Application NOM application
20 très    ADV très
21 bien    ADV bien
22 Commentaire ADJ <unknown>
23 :    PUN :
24 Application NOM application
25 magnifique ADJ magnifique
26 DateCommentaire:19 ADJ <unknown>
27 septembre    NOM septembre
28 2014    NUM @card@
29 NoteCommentaire:5 ABR <unknown>
```

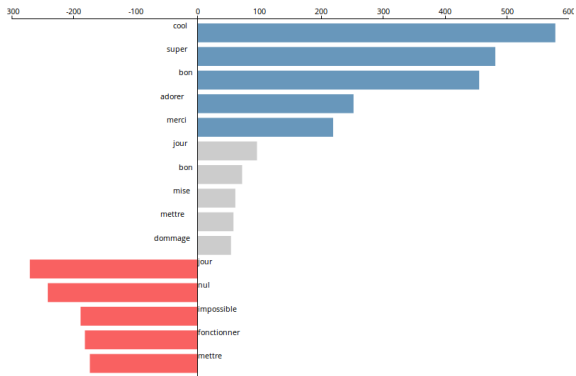
Commentaire dans ligne d'un fichier ARFF

```
6 bien, ': application très bien : application magnifique '
```

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 (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'occurrence 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

Fautes d'orthographe et de frappe

“Je kiff grave car jadore cyprien et jaimefai lui poser un question :
est ce que tu connais squeeze (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 \Rightarrow Les mêmes mots sont écrits différemment
- Texte brut lemmatisé \Rightarrow Les mots non reconnus disparaissent
- Texte lemmatisé \Rightarrow 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 \Rightarrow Meilleurs résultats

Sommaire

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