Comment obtenir une très bonne note sur TripAdvisor? Part I : un modèle LDA

Pierre Ghewy, Sébastien Chabrier, Christophe Benavent

MANAGEMENT & DATASCIENCE

COMMENT OBTENIR UNE TRÈS BONNE NOTE SUR TRIPADVISOR ?

PIERRE GHEWY, SÉBASTIEN CHABRIER & CHRISTOPHE BENAVENT

Publié dans Management & Datascience Vol.3 N°3, le 24 octobre 2019
Catégorie : Culture Data
DOI : https://doi.org/10.36863/mds.a.210.

Part I - un modèle LDA

RÉSUMÉ

A l'heure où Thomas Cook vient de défaillir, les plateformes de réservation dominent le trafic de la recherche d'information. Pour être dans les premiers, la maîtrise des notes et la qualité des avis de consommateur est indispensable. Le 5 étoiles sur 5 est nécessaire. La dictature du classement impose aux hôtels de comprendre ce qui fait obtenir la meilleure note.

AVIS CONSOMMATEUR | MÉTHODE LDA | TRIP ADVISOR

Citation : Ghewy, P., Chabrier, S., & Benavent, C. (Oct 2019). Comment obtenir une très bonne note sur TripAdvisor ? - Part I - un modèle LDA. *Management et Datascience*, *3*(3). https://doi.org/10.36863/mds.a.210..

Les auteurs :

- **Pierre Ghewy**
  - (Pas d'affiliation)
- **Sébastien Chabrier**
  - (Pas d'affiliation)
- **Christophe Benavent**
  (c.benavent@gmail.com) - (Pas d'affiliation) - ORCID : https://orcid.org/0000-0002-7253-5747 [https://orcid.org/0000-0002-7253-5747]

Liens d'intérêts : Le ou les auteurs déclarent ne pas avoir connaissance de conflit d'intérêts impliqués par l'écriture de cet article.

Financement : Le ou les auteurs déclarent ne pas avoir bénéficié de financement pour le travail mis en jeu par cet article.

TEXTE COMPLET

On propose dans cette note une méthode pour comprendre ce qui fait obtenir une excellente note à partir des thématiques évoquées dans le texte des avis écrits par les touristes. Il s'agira donc d'abord d'identifier des thèmes, souvent appelés topics, et de les relier de manière intelligente aux notes décernées. On devrait ainsi mieux comprendre ce qui fait une bonne note.

L'idée est simple : elle consiste à expliquer la note par la proportion de k contenus (ou en anglais *topic analysis*) évoqués dans les avis, ou plus précisément par la probabilité que le commentaire évoque l'une ou l'autre des thématiques identifiées. Le but est donc de transformer un texte en un vecteur de probabilités qui rende compte de la distribution des sujets dans le texte. Dans un second temps (l'objet de la seconde partie du billet), on construira un modèle prédictif de la note à partir du texte.

La méthode LDA

Pour atteindre ce résultat, la méthode LDA d'analyse de topics est idéale (Blei, Ng, et Jordan 2003) [https://www.zotero.org/google-docs/?iAZuGC] dans la mesure où un de ses résultats est l'estimation de la fréquence estimée (ou probabilité) qu'un thème appartienne au document (l'avis) (matrice theta de documents et k topics). C'est ce que fait exactement le modèle LDA. Il nous donnera aussi la probabilité qu'un mot appartienne à l'un ou l'autre des topics. (matrice beta des n termes et des k topics) qui s'ajuste au mieux à la distribution des mots dans les documents (matrice des w documents et n termes). Généralement un échantillonneur de Gibbs est employé pour estimer les probabilités a posteriori theta et beta qui représentent successivement la distributions des thèmes au sein de chaque document, et la probabilité d'apparition des mots conditionnée à leur appartenance à un thème particulier.

Préparation des données

Avant de mettre en oeuvre le modèle, il va falloir préparer le corpus. C'est une étape nécessaire pour obtenir des résultats clairs.

Le corpus est constitué des commentaires publiés sur un comparateurs d'hôtels couvrant l'ensemble de la polynésie et moissonnés par une méthode développée au Cetop (Centre d'étude Touristique de la Polynésie Française) de l'Université de Polynésie Française.

La méthode utilisée demande d'abord d'annoter le texte par une analyse des éléments du langage qui vise d'une part à lemmatiser les mots, puis à les étiqueter sur la base de leurs formes morpho-syntaxiques par une méthode de POS (*Part of Speech*) conduite avec le package cleanNLP (Arnold 2017) [https://www.zotero.org/google-docs/?UDu8jS].

```
# chargement des bibliothèques nécessaires
library(readr) #pour lire le fichier csv
library(tidyverse)#le package général dont ggplot2 pour les graphiques
library(gridExtra)# juste pour assembler les graphes en une figures
library(cleanNLP) # pour le POS
library(text2vec) #pour le LDA
library(reshape2) #pour travailler les fichiers de données
#lire le fichier de données
df <- read_csv("commentaire.csv")
#concaténer le titre et l'avis pour avoir plus de texte
text<-paste0(df$Titre,". ",df$Commentaire)
# activer clean_nlp avec le parser par défaut, le configurer en français
cnlp_init_udpipe(model_name = "french")
# l'annotation proprement dite
#obj <- cnlp_annotate(text, as_strings = TRUE)
# comme l'opération est longue, stocker le fichier sur un fichier en dur
#saveRDS(obj,"avis2018.rds")
# lire le fichier d'annotation quand nécessaire
obj<-readRDS(file="avis2018.rds")
```

Concrètement, on sélectionne les noms communs, qui indiquent ce dont on parle, les adjectifs et adverbes, qui indiquent comment on en parle, et les verbes, qui associent aux objets le sens d'une action. Il ne reste plus qu'à analyser la distribution des POS pour les mots les plus fréquents et celle des adjectifs, verbes et noms communs.

```
#extraction et création du fichier des lemmes et de leur POS
Vocab<-cnlp_get_token(obj)
Table <- with(Vocab, table(upos))
ling<-as.data.frame(Table)
# Production du graphique des POS
g1<-ggplot(ling,aes(x=reorder(upos,Freq),y=Freq))+
   geom_bar(stat="identity",fill="darkgreen")+
   coord_flip()+theme_minimal()+theme(text = element_text(size=9),axis.title.y=element_blank())+
   labs(title = "Catégories morpho-syntaxiques",x="catégories",y="fréquence des tokens")
g1
# Production du graphique des lemmes (noms communs)
Vocab1<-subset(Vocab, upos=="NOUN")
Table <- with(Vocab1, table(lemma))
ling<-as.data.frame(Table) %>% filter(Freq>400)
g2<-ggplot(ling,aes(x=reorder(lemma,Freq),y=Freq))+
  geom_bar(stat="identity",fill="brown1")+coord_flip()+
  theme_minimal()+ theme(text = element_text(size=9),axis.title.y=element_blank())+
  labs(title = "Noms communs",x="Noms commun",y="Fréquence des tokens")
g2
```

Le résultat apparaît dans la figure suivante, limitée aux noms commun qui ont une fréquence de 400, aux adjectifs et adverbes (>300) et les verbes (>200)

Figure 1 : Catégories morpho-syntaxiques : les noms communs, adjectifs et verbes les plus fréquents du corpus

La figure 1 donne le résultat de cette analyse en classant les termes par ordre de fréquence. Le mot hôtel apparait plus de 3 000 fois sur un ensemble de 5 000 documents, il risque de ne pas être informatif. À l'opposé, certains termes apparaissent rarements, il ne sont pas plus informatifs.

On va donc non seulement sélectionner certaines catégories de lemmes, mais en plus les écrémer en excluant ceux qui sont présents dans 95% des textes et dans moins de 5% des mêmes textes. Dans le code, on emploie les ressources "tidyverse", dont on trouvera une bonne introduction, en français, réalisée par Julien Barnier (https://juba.github.io/tidyverse/index.html [https://juba.github.io/tidyverse/index.html]).

```
tf <- cnlp_get_token(obj) %>%
  filter(upos %in% c("ADJ", "NOUN","VERB")) %>%
  cnlp_utils_tfidf(min_df = 0.05, max_df = 0.95, type = "tf", tf_weight = "raw")
```

Un modèle à huit topics

On peut alors introduire ce tableau dans le modèle LDA, et après un processus d'essais-erreurs, 8 topics ont été déterminés comme suffisants. L'identification du nombre est souvent la tâche la plus difficile, même s'il est possible de déterminer un optimum en calculant tous les modèles de 5 à 50 topics, ou plus (avec "ldatuning" de Nikita Murzintcev, https://cran.r-project.org/web/packages/ldatuning/vignettes/topics.html [https://cran.r-project.org/web/packages/ldatuning/vignettes/topics.html]).

On utilise le package "text2vec" de Dmitriy Selivanov (http://text2vec.org/ [http://text2vec.org/]) pour estimer le modèle à huit topics qui est retenu. Les autres hyper-paramètres (qui règlent le fonctionnement de l'algorithme) sont les probabilités a priori qu'un document soit relatif à un topic et qu'un mot apparaissent dans un topic, le nombre d'itérations maximums et la tolérance (écart minimum d'une itération à l'autre).

```
lda_model = LDA$new(n_topics = 8, doc_topic_prior = 0.1, topic_word_prior = 0.01) #spécification des paramètres
set.seed(67)#pour retrouver la même solution
doc_topic_distr = lda_model$fit_transform(x = tf, n_iter = 1000,
                          convergence_tol = 0.001, n_check_convergence = 25,
                          progressbar = FALSE) #le modèle LDA
```

Il ne reste plus qu'à examiner les résultats. On les représente graphiquement dans l'ordre de leur pertinence du terme w pour le topic k, ajusté par un paramètre lambda dont on choisit le plus souvent la valeur d'environ 0,3 (voir (Sievert 2014) [https://www.zotero.org/google-docs/?brSUdr] – auteur de la librairie LDAvis qui génère un diagramme interactif qui représente les topics et leurs proximités dans un plan, et facilite fortement l'interprétation) :

pertinence(term w | topic t) = λ * p(w | t) + (1 – λ) * p(w | t)/p(w)

où p(w | t) est la probabilité qu'un mot w appartiennent au topic t, et p(w) la probabilité de trouver le mot dans le corpus. Autrement dit si λ est très inférieur à 1, les mots rares et fortement associés au topic auront une plus grande pertinence comparés à des termes plus fréquents.

La solution obtenue dans la figure 2 est aisément interprétable.

```
# On extrait les 20 mots les plus pertinents du corpus en fonction d'un paramètre lambda (lambda = 1 probabilités)
lda_res<-as.data.frame(lda_model$get_top_words(n = 20, lambda = 0.25))
#on calcule les rangs des termes
lda_res$rank<-as.numeric(row.names(lda_res))
#On transforme le tableau de données pour obtenir un tableau long approprié pour ggplot (avec  la fonction melt de reshape2)
lda_res<-melt(lda_res,id.vars = c("rank"))
g6<-ggplot(lda_res, aes(x=variable, y= rank, group =  value , label = value)) +
   scale_y_reverse() +
   geom_text(aes(color=variable,size=sqrt(26-rank))) +
   theme_minimal()+scale_color_hue()+
   guides(color=FALSE,size=FALSE)+labs(x="topics", y="par ordre de pertinence")
g6
```

Les 8 topics peuvent se décrire de la manière suivante, ils reflètent les expériences vécues de manière flagrante.

- V1 – La carte postale : les pilotis, la plage, le bungalow plante le décors.
- V2 – le rapport qualité-prix : Il n'y a aucune ambiguïté, s'ajoute une mention géographique (proche de l'aéroport)
- V3 – Le dithyrambe : le magnifique, le superbe, l'exceptionnel, le parfait.
- V4 – Un paradis polynésien de fruits délicieux et de poissons.
- V5 – Une pension accueillante et simple qui met à disposition des vélos.
- V6 – un séjour agréable et une équipe chaleureuse.
- V7 – Les aménités de la chambre et de la vue.
- V8 – L'interaction client : marquée par de nombreux verbes (demander, devoir, faire, savoir).

Figure 2 : Solution à 8 topics du modèle LDA. Les termes sont classés par pertinence.

Les profils thématiques des notes

Il ne reste plus qu'à représenter la teneur des discours en fonction de la note donnée.

Chaque texte est donc désormais décrit par un vecteur de 8 variables qui reflètent la probabilité qu'il traite de l'un ou l'autre des 8 thèmes définis. On peut interpréter ces valeurs comme la proportion des thématiques dans le discours qui donne la composition thématique du discours de chaque avis.

On va donc les représenter par un simple diagramme en barres superposées. Pour chaque note on a directement la composition moyenne des textes qui ont 1 étoile, 2, … et 5 étoiles.

```
#extraction des variables topic
topic<- as.data.frame(doc_topic_distr)
#ajout des variables topics au fichier de données
df<-cbind(df,topic)
#renommer les variables
df$Cartepostale<-df$V1
df$RapportQltePx<-df$V2
df$Dithyrambe<-df$V3
df$Paradis<-df$V4
df$Pension<-df$V5
df$Sejour<-df$V6
df$Chambre<-df$V7
df$InteractionClient<-df$V8
# fabrication du bar plot
df$Note <-as.factor(df$Note)
foo<-   aggregate(cbind(Cartepostale,RapportQltePx,Dithyrambe,Paradis,Pension,Sejour,Chambre,InteractionClient)~Note,data=df,FUN
="mean")
foo<-melt(foo)
g5<-ggplot(foo, aes(x=Note, y=value,group=variable))+
   geom_bar(stat="identity",aes(fill=variable))
   +theme_minimal()
g5
```

Le résultat obtenu par cette dernière procédure est donné dans la figure trois. Si on s'intéresse à la meilleure note (5) on voit clairement que c'est la qualité du séjour et l'effet woah que traduit le dithyrambe qui font la différences avec sans doute l'expérience paradisiaque. La question du rapport qualité prix se pose plutôt pour les notes de quatre étoiles. Il est souvent question de pension.

Lorsque la note est faible (1, 2, 3), c'est l'interaction client/personnel qui est dominante dans le discours, dont on a remarqué sa forte composante verbale, signe du péremptoire, un langage à l'impératif !

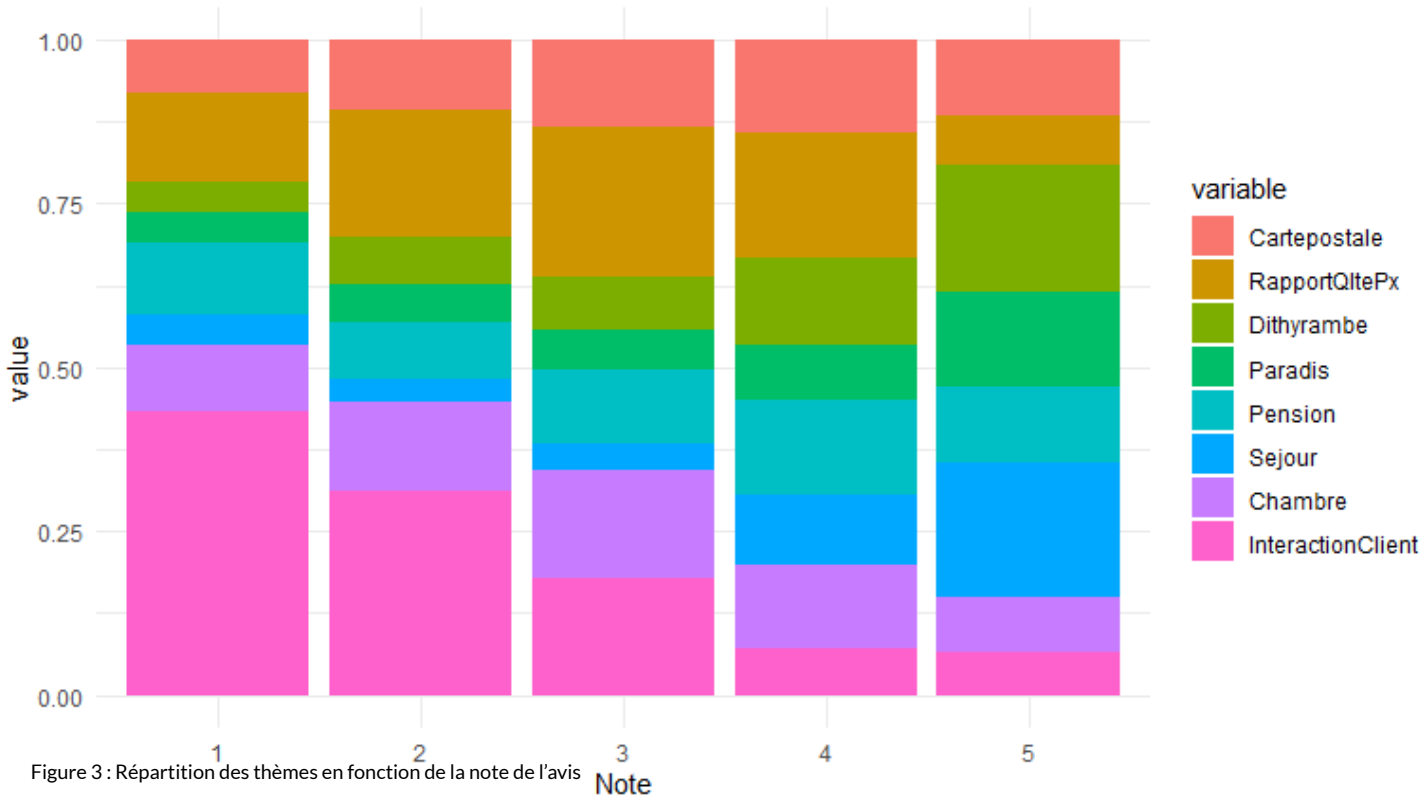

Figure 3 : Répartition des thèmes en fonction de la note de l'avis

Pour conclure

Le lecteur attentif aura bien sûr remarqué à la lecture des résultats factuels une vérité bien plus universelle que celle de ce set de données. On aura ainsi, avec la méthode LDA, vérifiée dans le texte, l'idée que seul l'expérience donne l'exceptionnel : un séjour très agréable et chaleureux, un goût de paradis, l'heureuse surprise d'un cadre magnifique forment le discours des meilleures notes.

Au cran d'en dessous c'est le trivial rapport qualité prix qui l'emporte. Quant aux notes, on en distingue la médiocrité par la proportion de la récrimination.

Naturellement on peut aller plus loin. Puisque de manière purement descriptive il semble qu'à chaque niveau de note correspondent une forme de discours tant d'un point de vue du style (surutilisation de verbes ou d'adjectifs) que du point de vue topical (le thème des contenus sémantiques), On peut se demander s'il n'est pas possible de construire un modèle prédictif.

Prédire la note en fonction du contenu du texte. Ce sera l'objet de la seconde partie de cette note.

BIBLIOGRAPHIE

- Arnold, Taylor. 2017. « A Tidy Data Model for Natural Language Processing Using CleanNLP ». *The R Journal* 9 (2): 248. https://doi.org/10.32614/RJ-2017-035.
- Blei, David M., Andrew Y. Ng, et Michael I. Jordan. 2003. « Latent Dirichlet Allocation ». *J. Mach. Learn. Res.* 3 (mars): 993–1022.
- Sievert, Carson. 2014. « LDAvis: A method for visualizing and interpreting topics ». *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. Baltimore, Maryland, USA (juin): 63–70.