

TP 3 - Analyse des enjeux des Premières nations et des Inuits et Métis dans les débats de la Chambre des Communes

Yun Shen (Yoshi) Li

Introduction

Les débats des politiciens de la Chambre des Communes et le Sénat du Canada m'ont toujours paru très intéressant (et ce, pas forcément pour le gouvernement canadien) car ils ont tous des intérêts différents, mais au fin fond travaillent tous pour le même gouvernement. Puisque les données du LIPAD sont disponibles, j'aimerais les utiliser pour analyser un débat spécifique de la Chambre: First Nations, Inuit and Métis Children, Youth and Families Act du 3 mai 2019.

Le Canada a une très longue histoire de maltraitance contre les Premières nations, ainsi que les peuples Inuits et Métis, même avec les nombreuses lois et actes passés en leur noms. Avec les données du LIPAD et la transcription du débat, j'aimerais analyser les enjeux reconnus lors de cette session dans la Chambre, dans le but de chercher l'importance des Premières nations dans l'agenda du gouvernement canadien. Plus précisément, j'aimerais aussi voir la différence dans les opinions des députés des différentes parties politiques.

Comme mentionné ci-dessus, les données utilisées proviennent du LIPAD et consiste d'une transcription en .csv du débat à la Chambre des Communes le 3 mai 2019. La discussion repose sur un nombre de sujets tels que les relations internationales, l'industrie d'agriculture, liberté de la presse, la sécurité publique, mais c'est principalement sur l'inattention du gouvernement par rapport aux promesses non-tenues aux Premières nations, les Inuits et les Métis. Donc, puisque le débat de cette journée repose principalement sur eux, mon analyse va reposer sur les enjeux que les députés apportent sur la table pour ensuite déterminer un degré final d'importance dans leur agenda.

Données utilisées

La transcription du débat dans la Chambre des Communes du Canada le 3 mai 2019 ('2019-05-03.csv') sera utilisé pour mon analyse; le .pdf de la transcription est d'environ 50 pages en

anglais depuis le site officiel de la Chambre des Communes. Les arguments et présentations en français ont été traduits en anglais, donc le document est entièrement en anglais, me permettant ainsi de faciliter l'analyse. Ma méthode commence d'abord avec l'importation de Lexicoder comme dictionnaire car il contient une énorme base de mots et de phrases qui seront indispensables dans ma recherche, et afin de ne pas avoir besoin de créer entièrement mon propre dictionnaire. J'ai aussi choisi de créer un petit dictionnaire qui comprend des mots spécifiques aux enjeux des peuples indigènes canadiens, que je vais combiner avec Lexicoder.

Nettoyage

Dans le nettoyage de mes données et des dictionnaires, j'ai tout d'abord enlevé les variables non-pertinentes à mon analyse et j'ai laissé `speechtext` (transcription de ce qui a été dit par les députés) et `speakerparty` (le parti politique auquel le député appartient), qui sont les deux que j'aimerais analyser pour répondre à ma question. De plus, j'ai enlevé les stopwords en anglais qui sont inutiles dans mon analyse. Ensuite, j'ai transformé la colonne entière de `speechtext` en minuscule avec `tolower()`, annulant ainsi le problème de case sensitivity. Finalement, j'ai enlevé les valeurs NA qui sont également inutiles. J'avais voulu utiliser les méthodes de stemming et lemmatization, mais je ne savais pas par où commencer puisque le document est aussi large.

Dictionnaire

J'avais choisi d'utiliser le dictionnaire Lexicoder pour sa facilité d'utilisation ainsi que pour les termes qui sont présents dedans. Ce dernier a été utilisé nombreuses fois pour des analyses textuelles, tout comme j'essaye de faire, et il contient aussi des termes qui seront pertinentes dans ma recherche. En le combinant avec mon petit dictionnaire "homemade" des termes que je perçois comme pertinents pour analyser les enjeux des peuples autochtones canadiens (`first_nations_dictionary`), j'ai créé un nouveau dictionnaire que je crois confondre à ce que j'aimerais réaliser. Lexicoder manquait des termes qui sont pertinentes et même nécessaires pour décrire les enjeux des autochtones du Canada.

Résultats

Je vais être honnête, j'ai eu tellement de bugs dans mon code que je ne sais même plus ce que j'ai fait...alors je vais devoir arrêter mon tp ici :(mais bon, voilà mon code >.<

```
#importing packages to prepare for text analysis

library(fs)
library(crayon)
library(quanteda)
```

```

library(tidyverse)
library(clessnverse)
library(dplyr)
install.packages("textTinyR")
library(textTinyR)

#importation de données et dictionnaire

debate_trans <- read.csv("/Users/yoshili/fas_1001_Li/lipad/2019/5/2019-5-3.csv")

lexicoder_eng <- dictionary(file = "/Users/yoshili/fas_1001_Li/policy_agendas_english.lcd")

speechtext_list <- as.list(debate_trans$speechtext)
debate_dictionary <- dictionary(speechtext_list) #transform debate_trans into dictionary

#création d'un autre dictionnaire

first_nations_dictionary <- list(issues = c("indigenous child", "families", "separation",
      "family reunification", "reform", "Jordan's pr
      "child and family services", "youth", "apprehe
      truth = c("truth and reconciliation commitee", "reconcili
      "keeping families together", "cultural experien
      "legislation", "bills", "laws", "injustice",
      "historical injustice", "self-determination", "
      aboriginal = c("first nations", "Inuit", "Métis")) |>
  dictionary()

#fusion de lexicoder et nouveau dico

first_nations_dictionary_m <- first_nations_dictionary |> stack() #dictionary to databse

lexicoder_eng_m <- lexicoder_eng |> stack() #dictionary into database

new_dico <- bind_rows(first_nations_dictionary_m, lexicoder_eng_m) |> #combine into one di
  unstack(values~ind) |> #transformation into list
  dictionary() #then dictionary

#nettoyage de base et sélection des données

debate_trans_preclean <- debate_trans |>
  select(speechtext, speakerparty) |> #garder seulement speechtext et speakerparty

```

```

mutate(speechtext = tolower(speechtext)) |> #converting speechtext values into lowercase
na.omit() #omitting NA values

#removing stopwords

english_stopwords <- stopwords(kind = "en")
debate_dictionary <- removeWords(debate_dictionary, english_stopwords)

#text analysis by speaker party

run_dictionary(data = debate_trans_preclean,
               text = speechtext,
               dictionary = lexicoder_eng_m) |>
bind_cols(debate_trans_preclean) |>
select(-c(doc_id,speechtext)) |>
pivot_longer(!speakerparty, names_to = "categorie", values_to = "n") |>
#assigning speakerparty to a categorie, values with n
#pivot_longer()shapes words from wide to long, basically shortening it to its root
ungroup() |>
group_by(speakerparty, categorie) |> #grouping speakerparty and categorie
summarise(n=sum(n)) |> #sum of values in category 'n'
mutate(prop = round(n/sum(n),3)*100, #creating variable prop (proportion) and calculating
       speakerparty = case_when(speakerparty == "Conservative" ~ "Conservateur",
                                speakerparty == "Green Party" ~ "Partie vert",
                                speakerparty == "New Democratic Party" ~ "NDP",
                                speakerparty == "Liberal" ~ "Libéral",
                                T ~ as.character(speakerparty)), #converting speakerparty
       categorie = case_when(categorie == "issues" ~ "Enjeux",
                             categorie == "truth" ~ "Vérité",
                             categorie == "aboriginal" ~ "Autochtone",
                             T ~ as.character(categorie))) |> #converting lexicoder categorie
na.omit() |> #omitting NA values
filter(categorie %in% c("Enjeux", "Vérité", "Autochtone"),
       !speakerparty == "Independent") |>
ggplot(aes(x = categorie, y = prop, fill = speakerparty)) + #creating point graph,
geom_bar(stat = "identity", position = "dodge") + #fitting regression lines
scale_fill_manual(values = c("skyblue", "purple", "pink")) +
coord_flip() + #flipping coordinates
labs(x = "Enjeux discutés",
     y = "Proportion en %",
     title = "Importance des enjeux des Premières nations, Inuit et Métis") +

```

```
theme_light() +  
theme(title = element_text(size = 20),  
      legend.text = element_text(size = 20),  
      axis.text = element_text(size = 20, color = "black"))
```