Categorizing Written Responses with a Dictionary Approach

Maxime St-Jean

2023-04-11

Contents

This memo describes and provides code for a dictionary-based method for categorizing written free responses in the 2019 CES. It is intended to provide a detailed explanation of our methods and to enable another researcher to replicate our approach with similar data.

Background

The 2019 CES was a survey of 37,822 respondents which asked about respondent's demographics as well as political and sociological attitudes, opinions, and behaviors. Respondents were asked in one question to write-in what the most important issue to them personally in the 2019 federal election was. Our goal was to put these free responses into broader categories, such as issues relating to the economy, environment, etc.

Approach

We approached the categorization task by first creating a list of keywords for each category we thought would be common. The keywords were generated by reviewing the literature political issue categories and identifying commonly used terms and phrases. For example, the environment dictionary started with words like "climate", "change", "oil", etc. These keywords formed the original dictionaries used.

We then ran the question responses through the dictionaries, and the number of words in a response that matched words in a specific dictionary was recorded. So for each category, a new variable was created which contains a number for each response corresponding to how many "hits" the response had for that category. We also created a dummy variable for each category which is entered as a 1 if there was any matches, and 0 if there were none.

Following the original categorization, we focused on responses that had not yet been categorized. These responses were pulled and sorted into alphabetical order so that common words could easily be seen. Common words were either added to existing category dictionaries if applicable, or a new category dictionary was created. This was repeated a couple times until there were no more terms widely used across the uncategorized responses.

Note: the text to be searched was both in English and in French, so dictionaries contain both English and French words, as well as common misspellings of words in both languages. The final dictionaries used are included in the code below.

Acknowledgements

Many thanks to Thomas Galipeau and Nadjim Fréchet for writing the original draft of the code.

Code

The code used in the analysis was written in R and relies on a variety of packages. The code is shown below, and the 2019 CES dataset is available online at http://www.ces-eec.ca/2019-canadian-election-study/. The code is fully documented and includes comments that explain each step of the process.

```
## Packages

library(tidyverse)
library(haven)
library(magrittr)

library(tidytext)
library(quanteda)
library(tm)
library(SnowballC)

library(crayon)
library(progressr)
library(progress)
library(lubridate)
library(quanteda.textstats)
library(tictoc)

####---------------------Dictionary approach-------------------------####

## Functions
runDictionary <- function(
    dataA, # input database where the search will be performed
    word, # name of the column to be searched
    dictionaryA) { # dictionary containing terms to be searched for
  tictoc::tic() # starts a timer (speedtesting the function)
  dataA <- dataA %>%
    mutate(word = {{word}}) # creates a copy of the input column called "word"
  corpusA <- tokens(dataA$word) # tokenizes text in the "word" column
  dfmA    <- dfm(tokens_lookup(corpusA, # checks for frequency of each token
                               dictionaryA,
                               nested_scope = "dictionary"))
  ## Progress bar ##
  pb <- progress_bar$new(
    format = yellow("  downloading [:bar] :percent in :elapsed"),
    total = 100, clear = FALSE, width= 60)
  purrr::walk(1:100, ~{pb$tick(); Sys.sleep(0.01)})
  message(green("100% expressions/words found"))
  tictoc::toc() # ends timer
  dataB   <- convert(dfmA, # puts the frequency count as a data.frame
                     to = "data.frame")
  return(dataB) # returns the count
}


## Set up

ces_19 <- ces_19 %>%
```

```r
  mutate_at(vars(cps19_imp_iss, cps19_imp_iss),
            function(x){gsub('[^ -~]', '', x)}) # removes special characters

ces_19$cps19_imp_iss <- tolower(
  ces_19$cps19_imp_iss) # puts all text to lowercase

## Economy

dictionaryeco <- dictionary( # creates the category-specific dictionary
  list(economy = c("economy", "jobs", "employment", "tax", "taxs", "taxes",
                   "job", "conomie", "con", "l'conomie",
                   "economie", "conomique", "dette", "debt", 'deficit',
                   "impts", "finances", "finance", "impot", "dficits",
                   "budget", "conomiques", "economics", "balanced",
                   "dollars", "deficits",
                   "evonomy", "low income", "spending", "trade",
                   "depenses", "déficit",
                   "dpense", "middle class", "taxing", "wage", "wages",
                   "economic", "conomiques", "budgets", "taxation",
                   "fiscal", "market", "recession", "growth", "loans",
                   "dollars", "budgétaire",
                   "leconomie", "argent", "l'endettement",
                   "living", "cost", "money", "spending",
                   "inequality", "prices", "trade", "inflation",
                   "poor", "enconomie", "ecomomie", "econamy",
                   "emploi", "ecomomy", "econics", "unemployment",
                   "impots", "affordability", "d'impot", "d'impo", "d'impt",
                   "emploie", "economique", "ecomony", "work", "unemployed",
                   "taxe", "taxed", "dficit", "financial", "budgtaire",
                   "l'economie")))

# finding those who mentioned one of the economic words in the 2019 CES
ces_19.econ <- runDictionary( # this creates a dataframe with a column counting
  dataA = ces_19,             # the number of times a word in the dictionary
  word = cps19_imp_iss,       # is mentioned.
  dictionaryA = dictionaryeco)

ces_19$economy <- ces_19.econ$economy # puts the column into the orginal data
ces_19 <- ces_19 %>%
  mutate(economy.dum = ifelse(economy >= 1, 1, 0)) # adds a binary column

# The same process as above is repeated for each category

## Enviro
dictionaryenviro <-dictionary(
  list(enviro = c("climate", "change", "envi", "pipelines", "oil", "carbon",
                  "pipeline", "environnement", "environmental", "environment",
                  "climate change", "l'environnement", "warming",
                  "l'environnement", "climatiques", "lenvironement", "ges",
                  "rechauffement", "gas", "enviroment", "water",
                  "sustainability", "enviromental", "environnement",
                  "enviroment", "écologie",
                  "l'envéronnement", "l'ecologie", "l'environnemen",
```

```
                  "l'environnemenr", "l'environnent", "emvironnement",
                  "environnemen5", "ecology", "co2", "polluer", "pollute",
                  "pollution", "planet", "nergtiques", "energy", "carbone",
                  "greener", "green", "climatique",
                  "environnment", "enviournment", "climat", "envioroment",
                  "earth", "cologie", "environnementaux",
                  "ecologie", "enviornment", "enviro",
                  "enviormental", "enironment",
                  "fossiles", "fossil", "environement", "environmentalism",
                  "l'cologie", "l'environement", "pipe")))

ces_19.enviro <- runDictionary(dataA = ces_19,
                               word = cps19_imp_iss,
                               dictionaryA = dictionaryenviro)

ces_19$enviro <- ces_19.enviro$enviro
ces_19 <- ces_19 %>%
  mutate(enviro.dum = ifelse(enviro >= 1,1,0))


## Immigration

dictionaryimmigration <- dictionary(
  list(immigration = c("immigration", "illgale", "illégale",
                       "minority", "discrimination",
                      "immigrants", "immigrant", "langue",
                      "l'imigration", "d'immigrant", "foreign",
                      "immigrations", "imagration", "imigration",
                      "immegrants", "l'immigration",
                      "emigration", "refugee", "refugees",
                      "immagration", "immgration")))

ces_19.immigration <- runDictionary(dataA = ces_19,
                                    word = cps19_imp_iss,
                                    dictionaryA = dictionaryimmigration)

ces_19$immigration <- ces_19.immigration$immigration
ces_19 <- ces_19 %>%
  mutate(immigration.dum = ifelse(immigration >= 1, 1, 0))


## Healthcare

dictionnaryhealthcare <- dictionary(
  list(healthcare = c("health", "health-care", "care", "sant", "soins", "life",
                      "mental", "disability","pharmacare",
                      "disabled", "drugs", "drug", "medicare", "santé",
                      "medical", "heath", "prescriptions", "doctors",
                      "sante", "santé", "soin", "santè", "docteur", "medical",
                      "healthcare", "healtcare", "heathcare", "hospitals",
                      "medicine")))

ces_19.healtchcare <- runDictionary(dataA = ces_19,
```

```
                                    word = cps19_imp_iss,
                                    dictionaryA = dictionnaryhealthcare)

ces_19$healthcare <- ces_19.healtchcare$healthcare
ces_19 <- ces_19 %>%
  mutate(healthcare.dum = ifelse(healthcare >= 1, 1, 0))


## Housing

dictionnaryhousing <- dictionary(
  list(housing = c("housing", "affordable", "rent", "homeless", "rental",
                   "unaffordable", "renting","home", "homes", "dwelling",
                   "loyer", "maisons", "sans-abris", "logement",
                   "logements", "rents", "homelessness")))

ces_19.housing <- runDictionary(dataA = ces_19,
                                word = cps19_imp_iss,
                                dictionaryA = dictionnaryhousing)

ces_19$housing <- ces_19.housing$housing
ces_19 <- ces_19 %>%
  mutate(housing.dum = ifelse(housing >= 1, 1, 0))


## Seniors

dictionnaryseniors <- dictionary(
  list(seniors = c("pension", "pensions", "seniors", "senior", "aines",
                   "ages", "cpp", "elderly", "oas", "aging",
                   "senior's", "retirement", "âgées", "ainés", "65",
                   "vieillisse", "viellesse", "vielliesse", "vieux",
                   "ainees", "aine", "vieillissement", "veillesse", "pesion",
                   "age", "old people", "retirees", "retraite", "retired",
                   "senoir")))

ces_19.seniors <- runDictionary(dataA = ces_19,
                                word = cps19_imp_iss,
                                dictionaryA = dictionnaryseniors)

ces_19$seniors <- ces_19.seniors$seniors
ces_19 <- ces_19 %>%
  mutate(seniors.dum = ifelse(seniors >= 1, 1,0))


## Leaders

dictionnaryleaders <- dictionary(
  list(leaders = c("trudeau", "justin", "libéral", "libral", "liberals",
                   "leadership", "leader", "justin", "conservatives",
                   "parties", "leaders", "pm", "andrew", "sheer", "singh",
                   "blanchet", "ndp", "bloc", "green", "paul", "may",
                   "otoole", "toole", "bernier", "pm", "politician",
```

```
                    "trudeau's", "o'toole", "libéraux", "leader", "ford",
                    "scheer", "prime minister", "candidate", "candidates")))

ces_19.leaders <- runDictionary(dataA = ces_19,
                                word = cps19_imp_iss,
                                dictionaryA = dictionnaryleaders)

ces_19$leaders <- ces_19.leaders$leaders
ces_19 <- ces_19 %>%
  mutate(leaders.dum = ifelse(leaders >= 1, 1, 0))


## Ethics

dictionnaryethics <- dictionary(
  list(ethics = c("gouvernement", "corruption", "honesty", "ethics",
                  "transparency", "accountability",
                  "responsibility", "truth", "lies", "lying", "ethical",
                  "transparent", "integretity", "corrupt",
                  "trustworthy", "dishonesty", "liar", "transparence", "moral",
                  "integrity", "honest", "trust", "corruptions",
                  "coruption", "credibility",
                  "greed", "promesses", "honestly", "honnetete",
                  "honntet", "morality", "moral", "morals",
                  "accountable", "accountibility")))

ces_19.ethics <- runDictionary(dataA = ces_19,
                               word = cps19_imp_iss,
                               dictionnaryethics)

ces_19$ethics <- ces_19.ethics$ethics
ces_19 <- ces_19 %>%
  mutate(ethics.dum = ifelse(ethics >=1,1,0))


## Education (MAXIME CREATED)

dictionnaryeducation <- dictionary(
  list(education = c("education", "ducation", "school", "schools",
                     "educational", "university", "tuition", "student",
                     "students", "schooling", "l'ducation", "l'education")))

ces_19.education <- runDictionary(dataA = ces_19,
                                  word = cps19_imp_iss,
                                  dictionnaryeducation)

ces_19$education <- ces_19.education$education
ces_19 <- ces_19 %>%
  mutate(education.dum = ifelse(education >=1,1,0))


## Crime and guns
```

```
dictionnarycrime <- dictionary(
  list(crime = c("crime", "crimes", "criminal", "criminals", "gang","gangs",
                 "safety", "gun", "firearm", "violence")))

ces_19.crime <- runDictionary(dataA = ces_19,
                              word = cps19_imp_iss,
                              dictionnarycrime)

ces_19$crime <- ces_19.crime$crime
ces_19 <- ces_19 %>%
  mutate(crime.dum = ifelse(crime >=1,1,0))


## Indigenous

dictionnaryindigenous <- dictionary(
  list(indigenous = c("indigenous", "aboriginal", "reconciliation",
                      "first nations",
                      "first nation", "indeginous", "native")))

ces_19.indigenous <- runDictionary(dataA = ces_19,
                                   word = cps19_imp_iss,
                                   dictionnaryindigenous)

ces_19$indigenous <- ces_19.indigenous$indigenous
ces_19 <- ces_19 %>%
  mutate(indigenous.dum = ifelse(indigenous >=1,1,0))


## Other welfare

dictionnarywelfare <- dictionary(
  list(welfare = c("childcare", "children", "daycare", "dental", "welfare",
                   "social services", "social programs", "social service",
                   "social program", "family", "families", "famille", "child",
                   "children's", "basic income", "familiale", "familles",
                   "poverty", "social assistance", "public", "pauvret",
                   "service", "services", "parental")))

ces_19.welfare <- runDictionary(dataA = ces_19,
                                word = cps19_imp_iss,
                                dictionnarywelfare)

ces_19$welfare <- ces_19.welfare$welfare
ces_19 <- ces_19 %>%
  mutate(welfare.dum = ifelse(welfare >=1,1,0))


## Electoral reform

dictionnaryelection <- dictionary(
  list(election = c("election", "electoral", "voting", "voter",
                    "representation", "democracy", "first past the post",
```

```
                    "proportional")))

ces_19.election <- runDictionary(dataA = ces_19,
                                 word = cps19_imp_iss,
                                 dictionnaryelection)

ces_19$election <- ces_19.election$election
ces_19 <- ces_19 %>%
  mutate(election.dum = ifelse(election >=1,1,0))


## Women's issues and abortion

dictionnarywomen <- dictionary(
  list(women = c("women", "women's", "abortion", "abortions","woman",
                 "woman's", "unborn", "reproductive", "femme", "femmes",
                 "gender", "maternity", "womens")))

ces_19.women <- runDictionary(dataA = ces_19,
                              word = cps19_imp_iss,
                              dictionnarywomen)

ces_19$women <- ces_19.women$women
ces_19 <- ces_19 %>%
  mutate(women.dum = ifelse(women >=1,1,0))


## Security/defense and IR

dictionnarysecurity <- dictionary(
  list(security = c("security", "defense", "international", "china",
                    "defence", "war", "wars", "relations", "global", "israel",
                    "u.s.", "segurity", "scurit", "military")))

ces_19.security <- runDictionary(dataA = ces_19,
                                 word = cps19_imp_iss,
                                 dictionnarysecurity)

ces_19$security <- ces_19.security$security
ces_19 <- ces_19 %>%
  mutate(security.dum = ifelse(security >=1,1,0))


## Qubec and Law 21

dictionnaryquebec <- dictionary(
  list(quebec = c("quebec", "21", "qubec", "francophone",  "lacit","laicit",
                  "laicite")))

ces_19.quebec <- runDictionary(dataA = ces_19,
                               word = cps19_imp_iss,
                               dictionnaryquebec)
```

```r
ces_19$quebec <- ces_19.quebec$quebec
ces_19 <- ces_19 %>%
  mutate(quebec.dum = ifelse(quebec >=1,1,0))


## Race

dictionnaryrace <- dictionary(
  list(race = c("race", "racism")))

ces_19.race <- runDictionary(dataA = ces_19,
                             word = cps19_imp_iss,
                             dictionnaryrace)

ces_19$race <- ces_19.race$race
ces_19 <- ces_19 %>%
  mutate(race.dum = ifelse(race >=1,1,0))
```