TP3

Claude Alie

Introduction

Ce travail vise à examiner différents types de dictionnaires susceptibles d'être utilisés dans le cadre d'un projet d'analyse de texte automatisé, et d'en observer les avantages et les inconvénients. Trois types de dictionnaires seront considérés, à savoir un dictionnaire construit par mots les plus fréquents, un dictionnaire construit par analyse thématique non supervisée, et un dictionnaire préconstruit et déjà validé. La base de données utilisée devra permettre l'utilisation d'un dictionnaire préconstruit et validé et être assez large pour permettre des traitements d'apprentissage non-supervisé. Notre choix a porté sur le corpus LIPAD (@beelen2017), qui recense les débats de la chambre des communes du Canada, dont nous avons extrait les textes de deux périodes comparables peu susceptibles de produire des changements thématiques importants. Ce corpus est vaste et permet l'utilisation du dictionnaire Lexicoder, largement utilisé pour des analyses à thèmes politiques. Il s'agit ici de voir si ces trois dictionnaires produisent des résultats (intuitivement) cohérents, sinon pourquoi, et que faudrait-il faire pour améliorer leurs résultats.

Méthode

Le corpus LIPAD (Canadian Hansard Dataset) (Beelen K, Thijm TA, Cochrane C, et al., 2017) utilisé recense tous les débats du parlement Canadien depuis 150 ans et en offre une transcription en format .csv accompagnée de diverses métadonnées, dont la date (speechdate), le thème principal abordé par l'intervenant (maintopic), le thème secondaire (subsubtopic), le parti politique de l'intervenant (speakerparty), son nom (speakeroldname), sa position (speakerposition), et le texte de son intervention (speechtext), entre autres. Nous utilisons les données du mois d'octobre des années 2016 et 2017. Ces années non-électorales intra-mandat libéral sont peu susceptibles d'entraîner des changements thématiques notables et ne sont pas perturbées par un événement majeur (comme la pandémie).

Le dictionnaire par mots les plus fréquents a été élaboré en construisant une matrice de traits de documents DFM (document-feature matrix) pour obtenir le nombre d'occurrences des mots des textes de la base de données après nettoyage. Les 50 mots les plus fréquents ont été regroupés par thèmes et convertis en dictionnaire. Le dictionnaire par analyse thématique non-supervisée

a été construit en utilisant le même DFM et en le traitant par STM (@roberts2013), une méthode de modélisation par thème moins contraignante et plus adaptée aux sciences humaines que sa variante plus connue LDA (Latent Dirichlet Allocation). Tirant partie des métadonnées du corpus et les utilisant comme covariables, elle vise à identifier, sans étiquetage préalable, les principaux thèmes du texte (sujets latents) et les mots qui y sont associés en analysant les probabilités de cooccurrences textuelles. Finalement, le dictionnaire pré-construit, nous utilisons le Lexicoder Topic Dictionaries (LTD), développé par Albugh, Quinn, Sevenans et Soroka (@albugh2013), qui vise à capturer les sujets abordés dans les contenus d'actualités, les débats législatifs et les documents politiques. Les sujets couverts par ce dictionnaire sont (en anglais): aboriginal, civil_rights, land-water-management, agriculture, crime, culture, defence, education, energy, environment, finance, fisheries, foreign_trade, forestry, government_ops, healthcare, housing, immigration, intergovernmentalconstitutional_natl_unity, intl_affairs, labour, macroeconomics, prov_local, religion , social_welfare , sstc, transportation. Utilisant principalement les librairies quanteda, tidyverse et clessnverse, les données LIPAD de chaque jour d'octobre ont d'abord été lues et liées (binding) pour les années 2016 et 2017. Puis elles ont été nettoyées et formatées en fonction des dictionnaires utilisés. Pour les dictionnaires construits par fréquence et par STM, une liste de mots outils a d'abord été extraite de la bibliothèque quanteda (snowdown). Les textes sont ensuite transformés en corpus et en DFM (Document-Feature Matrix) en y intégrant les variables maintopic, subtopic et speechtext. Les mots du texte de l'intervention (speechtext) ont ensuite été tokenisés, mis en minuscules et en radicaux (stemming avec * pour chercher tous les mots ayant le radical recherché) et nettoyés (élimination des nombres, de la ponctuation, des mots outils et des mots dont l'occurrence est moindre que 2). Pour la construction du dictionnaire par fréquences, une variable de proportion thématique a été créée à partir de la variable de fréquence (fréquence du mot / somme totale des mots pour chaque parti politique et pour chaque année) pour former un data.frame contenant les 50 mots les plus fréquents, classés par ordre de fréquence. Ceux-ci ont ensuite été regroupés selon leurs affinités thématiques (identifiées manuellement) en quatre thèmes : enjeux liés au parlement (parliament), à la nation (country), aux problèmes du peuple (issues) et aux actions pour les résoudre (actions). Les mots associés à chaque thème ont été radicalisés et généralisés (par *).

```
- [parliament]:
  - govern*, minist*, member*, speaker*, mr*, bill*, year*, today*, say*, colleagu*
- [country]:
  - nation*, canada, peopl*, canad*
- [issues]:
  - liber*, right*, job*, famili*, tax*, question*, $, import*, communit*, work*, liber, right,
job, famili, tax, question, $, import, community, work
- [actions]:
  - make, go*, veri*, like*, want*, now, need*, support*, take*, issu*, get*, plan*, help*, way
*, new*, chang*, time*
```

La création du dictionnaire par STM a impliqué au préalable une analyse STM avec 4 thèmes (K = 4), qui a nécessité 11 itérations et produit les regroupements suivants :

```
Topic 1 Top Words:
         Highest Prob: canadian, member, speaker, mr, need, go, us
         FREX: canadian, member, speaker, us, said, trade, retir
         Lift: aid, fundrais, round, –mean, 10-billion, 155-million, 30-plus
         Score: canadian, member, speaker, mr, us, oh, take
Topic 2 Top Words:
         Highest Prob: peopl, bill, year, one, liber, becaus, support
         FREX: one, becaus, good, made, even, action, system
         Lift: algoma, crtc, long-stand, moustach, tune, bay, behaviour
         Score: peopl, one, becaus, year, bill, right, support
Topic 3 Top Words:
         Highest Prob: work, time, make, want, countri, today, nation
         FREX: work, countri, issu, women, money, ride, worker
         Lift: anniversari, asid, chair, solv, 100th, 27-year-old, 30-billion
         Score: work, countri, issu, time, want, today, women
Topic 4 Top Words:
         Highest Prob: govern, canada, minist, hous, can, veri, tax
         FREX: agreement, look, econom, ensur, across, cpp, emiss
         Lift: abbotsford, downgrad, essex, mail, pulp, tend, attempt
         Score: canada, minist, govern, hous, busi, famili, like
```

Chaque thème s'est vu attribué une étiquette significative après avoir considérés les mots retenus par l'algorithme en fonction de leur probabilité (Highest Prob), rang (FREX) et deux scores (Lift et Score). Le dictionnaire a été formé en associant les mots retenus à leurs thèmes respectifs, à savoir politique canadienne (can_politics), problèmes sociaux et législatifs (soc_issues_snd_legis), travail et économie (work_and_economy), et gouvernement et poliques (gov_and_policy). Les mots de chaque thème ont ensuite été radicalisés et généralisés (par *), comme suit :

```
Dictionary object with 4 key entries.
- [can_politics]:
  - aid*, billion*, canad*, fundrais*, go*, mean*, member*, million*, mr*, need*, oh*, retir*, roun
said, speaker*, take, trade*, us
- [soc_issues_and_legis]:
  - action*, algoma*, bay*, becaus*, behaviour*, bill*, crtc, even, good*, liber*, long-stand*, mad
moustach*, one*, peopl*, right*, support*, system*, tune*, year*
- [work_and_economy]:
  - 100th*, anniversari*, asid*, billion*, chair*, countri*, issu*, make*, money*, nation*, ride*,
v*, time*, today*, want*, wom*, work*, worker*, year-old
- [gov_and_policy]:
  - govern*, canad*, minist*, hous*, can, veri*, tax*, agreement*, look*, econom*, ensur*, across,
p, emiss*, abbotsford*, downgrad*, essex*, mail*, pulp*, tend* [ ... and 4 more ]
```

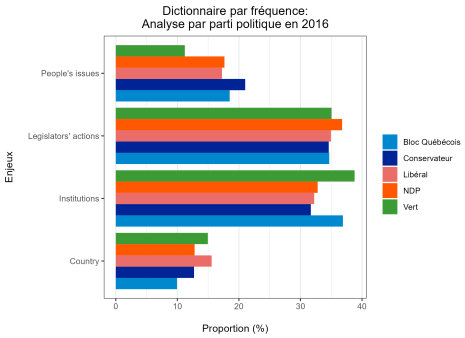
Pour l'analyse par dictionnaire préconstruit Lexicoder LTD, les données ont été préparées en choisissant les variables relatives à la date des interventions (speechdate), au parti de chaque intervenant (speakerparty) et à leurs interventions (speechtext). Elles ont ensuite été nettoyées par mise en minuscules et élimination des données manquantes. Aucune lemmatisation ou radicalisation (stemming) n'a été nécessaire à cette étape, ces opérations ayant déjà été effectuées. Le dictionnaire ainsi traité se présente comme suit :

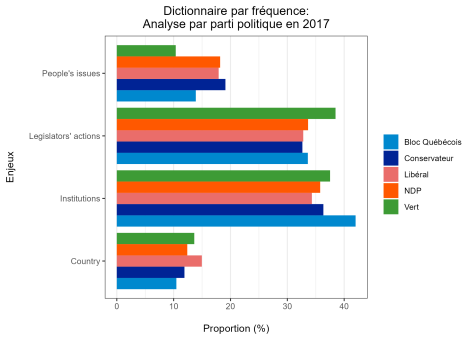
```
Dictionary object with 28 key entries.
- [macroeconomics]:
  - aggregate demand, aggregate supply, business cycle, demand shock, demand side, demand-side, eco
m, employment rate, full employment, food price, industr, keynes, bank of canada, bank of england,
r market, bretton woods, budget, bull market, changing demographic, coinage [ ... and 62 more ]
- [civil_rights]:
  - civil right, ableism, abortion, access to info, african american, anti-choice, anti-semit, bill
1, charter of the french language, biphobi, bisexual, charter of rights, civil libert, disabilit, d
riminat, diversity, equal employm, equal opportunit, equal right, equalit [ ... and 65 more ]
- [healthcare]:
  - aids, alcoholism, allerg, anaesthesiolog, anesthesiolog, cancer, cardiolog, cardiothoracic, car
vascular, cigarette, dermatolog, dietic, disease, disorder, doctor, drug treatment, drug abuse, end
in, gastroenterolog, geriatric [ ... and 108 more ]
- [agriculture]:
  - agricult, cattle, cultivat, grain, wheat, barley, beef, pork, poultry, tractor, thresh, orchard
ood inspect, farm, food import, aquacult, foot and mouth, livestock, crop, agri-food [ ... and 9 mo
]
- [forestry]:
  - agroforest, deforest, reforest, forest, logging, lumber, seedling, timber, tree, softwood, wood
- [labour]:
  - cpp, qpp, qpip, pension, employ, hire, hiring, income, internship, labor, labour, laid off, liv
wage, lockout, lock-out, maternity leave, minimum wage, parental leave, paternity leave, paycheck [
... and 44 more ]
[ reached max_nkey ... 22 more keys ]
```

L'application des dictionnaires aux données s'est effectuée en utilisant la méthode run_dictionary de quanteda en choisissant un dictionnaire, le corpus et le texte du corpus. Dans le cas du dictionnaire LTD, il a de plus fallu choisir un sous-ensemble de thèmes, et nous en avons choisi quatre qui s'apparentent à ceux des deux dictionnaires construits, à savoir : macroéconomie (macroeconomics), bien-être social (social_welfare), affaires intergouvernementales (intergovernmental), et unité canadienne et constitution (national unity & constitution). L'application du dictionnaire a été suivie d'opérations de nettoyage de variables inutiles, de pivotage en forme longue, de regroupement par parti politique, de calcul de proportions, de renommage de variables, de suppression des NA, et de visualisation du résultat. Une analyse a été effectuée pour chaque dictionnaire pour les années 2016 et 2017.

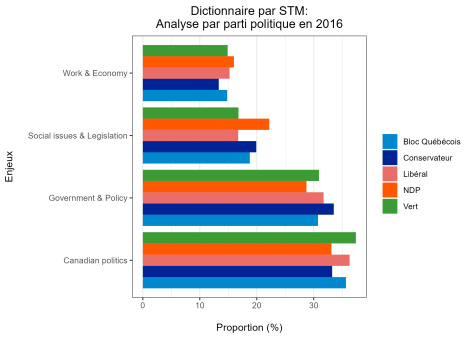
Résultats

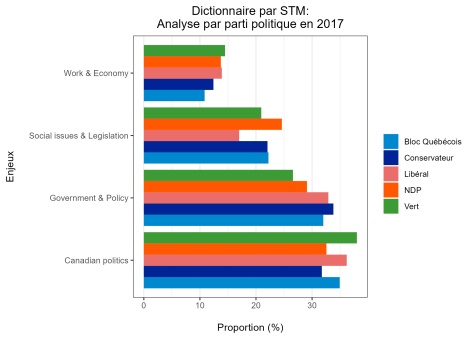
La figure 1 présente les résultats obtenus à partir du dictionnaire par fréquence. On y trouve la proportion d'interventions de chaque parti en fonction des thèmes de ce dictionnaire pour les années 2016 et 2017. En 2016, on observe d'abord que l'importance relative de chacun de ces thèmes varie peu d'une année à l'autre. Les législateurs traitent surtout d'actions à mener (Legislators' actions) et d'affaires institutionnelles, et un peu moins des problèmes du peuple (People's issues) et des thèmes liés au pays (Country). On note que les verts évoquent moins souvent les thèmes liés aux problèmes quotidiens (étant peut-être plus préoccupés par l'écologie?). Les actions à entreprendre intéressent tous les partis. Les institutions fédérales préoccupent davantage le Bloc québécois (peut-être pour les critiquer) et les Verts (peut-être pour les secouer). Sans surprise, le thème du pays (Canada) intéresse beaucoup le parti libéral mais peu le Bloc québécois. Il intéresse aussi les verts (qui ont une perspective globaliste). Ces observations sont grosso modo confirmées en 2017.



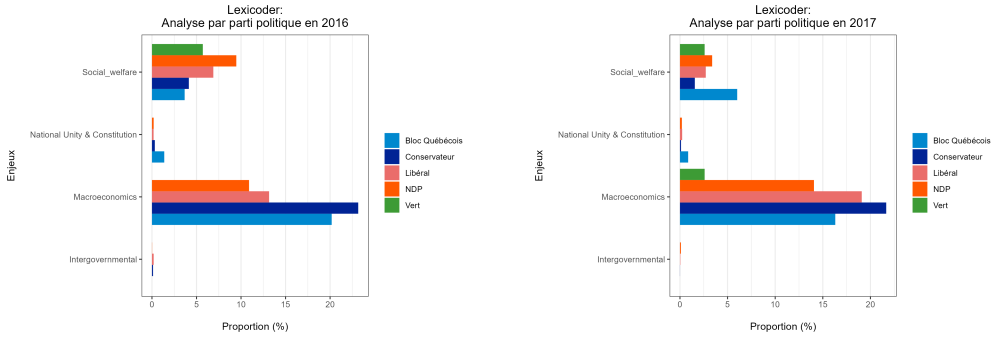


Tout comme le dictionnaire par fréquence, le dictionnaire par STM montre peu de changements d'une année à l'autre, et relativement peu de changements d'un parti à l'autre. Si on est peu surpris que le NPD s'intéresse aux conditions sociales, on l'est davantage que les conservateurs s'intéressent peu à l'économie.





Finalement, le Lexicoder, pour lequel on a choisis des thèmes apparentés à ceux mis à lumière par les deux dictionnaires construits, présente aussi des proportions thématiques par parti qui sont relativement stables d'une année à l'autre. On note que le thème du bien-être social est privilégié par le NPD et le parti libéral, le thème de l'unité nationale et la constitution par le Bloc québécois, le thème macroéconomique par les conservateurs et le bloc québécois, et le thème intergouvernemental, très peu abordé, par les libéraux.



Discussion et conclusion

Ces résultats montrent que tous les dictionnaires sont relativement constants d'une année à l'autre, présentant ainsi une certaine fidélité étant donné que les années étudiées sont peu susceptibles d'avoir produit des changements thématiques notables. Cependant, les résultats du Lexicoder sont davantage conformes à l'intuition pour les différents partis fédéraux sur chacun des thèmes abordés. Ils sont davantage contrastés, montrant de manière assez réaliste que les débats portent surtout sur l'économie et les affaires sociales, et dans une bien moindre mesure sur l'unité nationale et les affaires intergouvernementales. Les partis qui dominent pour chaque thème sont grosso modo ceux qu'on s'attendrait à voir dominer (NPD pour le bien-être sociale, conservateurs pour l'économie). Contrairement au Lexicoder, élaboré en fonction de relations thématiques intrinsèques, les deux dictionnaires construits sont d'abord fondés sur des relations de probabilité d'occurrences (probabilités simples pour le dictionnaire par fréquence d'occurrence et probabilités conditionnelles pour le dictionnaire par STM), auxquels on a subséquemment ajouté une interprétation thématique. Cette opération d'interprétation est délicate et difficile à réussir (ref). Ainsi, le dictionnaire par STM présente deux thèmes

difficiles à différentier, même avec une analyse fine et éclairée (gouvernement et politiques vs. politiques canadiennes). Ces deux dictionnaires présentent néanmoins l'avantage de mettre en lumière les thèmes issus des mots les plus utilisés, et ceux issus des mots les plus associés. Construits pour les fins d'un travail pratique, ils n'ont pas bénéficié d'autant d'attention et de validation que le Lexicoder, et il est impossible de faire l'économie d'un tel travail de peaufinage et de validation (ref). En particulier, concernant le dictionnaire par STM, il n'est pas garanti que K = 4 soit la valeur la plus adéquate.

Annexe: Codes utilisés

Librairies utilisées pour la création des bases de données et des dictionnaires

Lecture des jeux de données du corpus de LIPAD

```
annee = "2016"
path = paste0("_data/lipad/", annee, "/10/")
print(path)
files <- dir_ls(path, regexp = annee) # pour naviguer dans ses dossiers

Data_parl <- files |>
  map(~read_csv(files)) |>
  reduce(bind_rows) |>
  distinct()

names(Data_parl)
```

Préparation des données

Pour le dictionnaire préconstruit Lexicoder

```
# Pour analyse par date et parti politique
Data_parl_pre_clean_1 <- Data_parl |>
  select(speechdate, speakerparty, speechtext) |>
  ## Mise en minuscule ##
  mutate(speechtext = tolower(speechtext)) |>
  ## Suppression des NA's ##
  na.omit()

# Pour analyse par parti politique seulement
Data_parl_pre_clean_2 <- Data_parl |>
```

```
  select(speakerparty, speechtext) |>
  ## Mise en minuscule ##
  mutate(speechtext = tolower(speechtext)) |>
  ## Suppression des NA's ##
  na.omit()
```

Pour les dictionnaires construits par fréquence et par STM

```
stop_words <- read.csv("stop_words.csv", header = FALSE, stringsAsFactors = FALSE)[,1]
stop_words
```

```
# Création et nettoyage du DFM
books_dfm <- corpus(tibble(maintopic = Data_parl$maintopic,
                           subtopic = Data_parl$subtopic,
                           text = Data_parl$speechtext)) %>%
  tokens(remove_numbers = TRUE, remove_punct = TRUE) %>%
  dfm(tolower = TRUE) %>%
  dfm_wordstem() %>%
  dfm_trim(min_termfreq = 2) %>%
  dfm_remove(pattern = stop_words)
books_dfm
```

Création des dictionnaires construits

Dictionnaire Lexicoder

```
lexicoder_en <- dictionary(file = "_dictionary/policy_agendas_english.lcd", format = "yosh
names(lexicoder_en)
print(lexicoder_en)
```

Dictionnaire par fréquence

```
freq_dictionary <- list(parliament   = c("govern*", "minist*", "member*", "speaker*", "mr*
                        country = c("nation*", "canada", "peopl*", "canad*"),
                        issues = c("liber", "right", "job", "famili", "tax", "question", "
                        actions = c("make", "go*", "veri*", "like*", "want*", "now", "need
                        ) |> # on prépare une liste avec des catégories + mots clés
  dictionary() # On transforme la list() en dictionnaire avec la fonction de quanteda
```

```
freq_dictionary
```

Dictionnaire par STM

Analyse STM

```
# Analyse thématique

library(stm)
library(beepr)
library(lda)
library(topicmodels)
library(textclean)

# Calculate with stm
#stm_model <- stm(documents = books_dfm, K = 4)
#beepr::beep()

#book_topics <- labelTopics(stm_model)
#book_topics

#write_rds(
#  book_topics,
#  file = "book_topics.rda"
#)

book_topics <- read_rds(
  file = "book_topics.rda"
)
```

```
# Dictionnaire "home made"

stm_dictionary <- list(can_politics      = c("aid*", "billion*", "canad*", "fundrais*", "g
                                             "mean*", "member*", "million*", "m
                                             "retir*", "round*", "said", "speak
                                             "trade*", "us"),
                       soc_issues_and_legis = c("action*", "algoma*", "bay*", "
                                                "behaviour*", "bill*", "crtc"
                                                "good*", "liber*", "long-stan
```

```
                                                    "moustach*", "one*", "peopl*"
                                                    "support*", "system*", "tune*
                                  work_and_economy = c("100th*", "anniversari*", "asid*",
                                                       "countri*", "issu*", "make*", "money*",
                                                       "solv*", "time*", "today*", "want*", "w
                                                       "worker*", "year-old"),
                                  gov_and_policy = c("govern*", "canad*", "minist*", "hous
                                                     "tax*", "agreement*", "look*", "econo
                                                     "across", "cpp", "emiss*", "abbotsfor
                                                     "essex*", "mail*", "pulp*", "tend*",
                                                     "minist*", "govern*", "hous*", "busi*
                                  ) |>
  dictionary() # On transforme la list() en dictionnaire avec la fonction de #quanteda
names(stm_dictionary)
stm_dictionary
```

Analyses de dictionnaire

Dictionnaire par fréquence: Analyse par parti politique

```
title <- paste0("Dictionnaire par fréquence:\nAnalyse par parti politique en ", annee)
Plot1 <- run_dictionary(data       = Data_parl_pre_clean_2,
                text       = speechtext,
                dictionary = freq_dictionary) |>
  # On reprend les mêmes colonnes pour l'analyse #
  bind_cols(Data_parl_pre_clean_2) |>
  select(-c(doc_id,speechtext)) |>
  # Pivote la base de données pour voir la proportion par speaker #
  pivot_longer(!speakerparty, names_to = "categorie", values_to="n") |>
  ungroup() |>
  group_by(speakerparty, categorie) |>
  summarise(n=sum(n)) |>
  mutate(prop = round(n/sum(n),4)*100,
         speakerparty = case_when(speakerparty == "Conservative"          ~ "Conservateur"
                                  speakerparty == "Green Party"           ~ "Vert",
                                  speakerparty == "New Democratic Party"  ~ "NDP",
                                  speakerparty == "Liberal"               ~ "Libéral",
                                  T ~ as.character(speakerparty)),
         categorie = case_when(categorie == "parliament" ~ "Institutions",
                               categorie == "country" ~ "Country",
```

```
                                categorie == "issues" ~ "People's issues",
                                categorie == "actions" ~ "Legislators' actions",
                                T ~ as.character(categorie))) |>
  na.omit() |>
  # À des fins d'exemple on garde seulement certains thèmes + on enlève les indépendants #
  filter(categorie %in% c("Institutions", "Country", "People's issues", "Legislators' acti
         !speakerparty == "Independent") |>
  ggplot(aes(x = categorie, y = prop, fill = speakerparty)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual("", values = c("#0088CE", "#002395", "#EA6D6A", "#FF5800", "#3D9B35"))
  coord_flip() +
  labs(x = "Enjeux\n",
       y = "\nProportion (%)",
       title = title) +
  theme_bw() +
  ## theme() en fonction des dimensions dans Quarto ##
  #theme(title = element_text(size = 15),
  #      legend.text = element_text(size = 12),
  #      axis.text = element_text(size = 12, color = "black"))
  theme(plot.title = element_text(hjust = 0.5, size = 13),
        plot.subtitle = element_text(hjust = 0.5, size = 11),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 9),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()
        )
path <- paste0("_figures/Plot_freq_", annee, ".png")
ggsave(path, plot = Plot1)
```

Dictionnaire par STM: Analyse par parti politique

```
title <- paste0("Dictionnaire par STM:\nAnalyse par parti politique en ", annee)
Plot2 <- run_dictionary(data       = Data_parl_pre_clean_2,
                text       = speechtext,
                dictionary = stm_dictionary) |>
  # On reprend les mêmes colonnes pour l'analyse #
  bind_cols(Data_parl_pre_clean_2) |>
  select(-c(doc_id,speechtext)) |>
  # Pivote la base de données pour voir la proportion par speaker #
  pivot_longer(!speakerparty, names_to = "categorie", values_to="n") |>
```

```
  ungroup() |>
  group_by(speakerparty, categorie) |>
  summarise(n=sum(n)) |>
  mutate(prop = round(n/sum(n),4)*100,
         speakerparty = case_when(speakerparty == "Conservative"          ~ "Conservateur"
                                  speakerparty == "Green Party"           ~ "Vert",
                                  speakerparty == "New Democratic Party"  ~ "NDP",
                                  speakerparty == "Liberal"               ~ "Libéral",
                                  T ~ as.character(speakerparty)),
         categorie = case_when(categorie == "soc_issues_and_legis" ~ "Social issues & Legi
                               categorie == "can_politics" ~ "Canadian politics",
                               categorie == "work_and_economy" ~ "Work & Economy",
                               categorie == "gov_and_policy" ~ "Government & Policy",
                               T ~ as.character(categorie))) |>
  na.omit() |>
  # À des fins d'exemple on garde seulement certains thèmes + on enlève les indépendants #
  filter(categorie %in% c("Social issues & Legislation", "Canadian politics", "Work & Econ
         !speakerparty == "Independent") |>
  ggplot(aes(x = categorie, y = prop, fill = speakerparty)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual("", values = c("#0088CE", "#002395", "#EA6D6A", "#FF5800", "#3D9B35"))
  coord_flip() +
  labs(x = "Enjeux\n",
       y = "\nProportion (%)",
       title = title) +
  theme_bw() +
  ## theme() en fonction des dimensions dans Quarto ##
  #theme(title = element_text(size = 15),
  #      legend.text = element_text(size = 12),
  #      axis.text = element_text(size = 12, color = "black"))
  theme(plot.title = element_text(hjust = 0.5, size = 13),
        plot.subtitle = element_text(hjust = 0.5, size = 11),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 9),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()
        )
path <- paste0("_figures/Plot_stm_", annee, ".png")
ggsave(path, plot = Plot2)
```

Lexicoder: Analyse par parti politique

```r
title <- paste0("Lexicoder:\nAnalyse par parti politique en ", annee)
Plot3 <- run_dictionary(data       = Data_parl_pre_clean_2,
                text       = speechtext,
                dictionary = lexicoder_en) |>
  # On reprend les mêmes colonnes pour l'analyse #
  bind_cols(Data_parl_pre_clean_2) |>
  select(-c(doc_id,speechtext)) |>
  # Pivote la base de données pour voir la proportion par speaker #
  pivot_longer(!speakerparty, names_to = "categorie", values_to="n") |>
  ungroup() |>
  group_by(speakerparty, categorie) |>
  summarise(n=sum(n)) |>
  mutate(prop = round(n/sum(n),4)*100,
         speakerparty = case_when(speakerparty == "Conservative"          ~ "Conservateur"
                                  speakerparty == "Green Party"           ~ "Vert",
                                  speakerparty == "New Democratic Party"  ~ "NDP",
                                  speakerparty == "Liberal"               ~ "Libéral",
                                  T ~ as.character(speakerparty)),
         categorie = case_when(categorie == "macroeconomics" ~ "Macroeconomics",
                               categorie == "social_welfare" ~ "Social_welfare",
                               categorie == "intergovernmental" ~ "Intergovernmental",
                               categorie == "constitutional_natl_unity" ~ "National Unity
                               T ~ as.character(categorie))) |>
  na.omit() |>
  # À des fins d'exemple on garde seulement certains thèmes + on enlève les indépendants #
  filter(categorie %in% c("Macroeconomics", "Social_welfare", "Intergovernmental", "Nation
         !speakerparty == "Independent") |>
  ggplot(aes(x = categorie, y = prop, fill = speakerparty)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual("", values = c("#0088CE", "#002395", "#EA6D6A", "#FF5800", "#3D9B35"))
  coord_flip() +
  labs(x = "Enjeux\n",
       y = "\nProportion (%)",
       title = title) +
  theme_bw() +
  ## theme() en fonction des dimensions dans Quarto ##
  theme(plot.title = element_text(hjust = 0.5, size = 13),
        plot.subtitle = element_text(hjust = 0.5, size = 11),
        axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 9),
        panel.grid.major.y = element_blank(),
```

```
        panel.grid.minor.y = element_blank()
        )
path <- paste0("_figures/Plot_lexicoder_", annee, ".png")
ggsave(path, plot = Plot3)
```