Midterm Assignment - C

Part C (30 points)

This part of the assignment is a more open ended mini-project. You will be graded on the quality of your analysis, discussion, and presentation. You will not be penalised if you find a null result. You may use the range of text analysis concepts introduced in the course up to and including the week 7 materials. **Word limit is 750 words**. You may write as much code & code comments as you like, but ensure all of your results are clearly presented and discussed in text. We advise mixing code chunks with text discussion so the marker can more easily follow the project narrative. You will not be allowed to re-use any substantive analysis from this work in your end of term project, but you may use the same data.

0. Generative AI

You are allowed and encouraged to use generative AI tools like Copilot and ChatGPT for the coding elements of this assignment. If you do so, please indicate in each answer with a triple hash comment ("###") where it has been used to contribute large chunks of code, either through autocompletions or a specific prompt (provide the prompt). It is not necessary to indicate the use of short Copilot autocompletions unless they amount to a substantive proportion of the code in an answer.

1. Mini-project

Use one of the suggested datasets below, or another dataset of your choosing to complete a text analysis mini-project. You may not use a corpus already studied in seminars/assignments, but you may use any of quanteda's built-in support data such as dictionaries.

Do not include your data files from this mini-project in your submission repository, it may cause issues with submission and marking. You will be penalised if you do so. If you have collected your own data, do not include any scraping / API code in this worksheet, simply read the csv/json/txt file. Consult us in office hours if you are having difficulty reading data.

Introduce your data and ideas. Pose one or two research questions that you would like to answer. Read, process, analyse, and visualise the data to answer these questions. Provide a discussion and conclusion of your findings.

Suggested data sources:

- quanteda built-in datasets (https://github.com/quanteda/quanteda.corpora) (excluding those previously used on the course)
- This collection (https://github.com/EmilHvitfeldt/R-text-data?tab=readme-ov-file)
- Hate Speech and Offensive Language (https://github.com/t-davidson/hate-speech-and-offensive-language) (content warning)
- Wikipedia - good vs "promotional" articles (https://www.kaggle.com/datasets/urbanbricks/wikipedia-promotional-articles) (free Kaggle account required for this and others below)
- News category dataset (https://www.kaggle.com/datasets/rmisra/news-category-dataset)
- Reddit depression dataset (https://www.kaggle.com/datasets/infamouscoder/depression-reddit-cleaned/data)
- US economic news articles (https://www.kaggle.com/datasets/heeraldedhia/us-economic-news-articles)

YOUR ANSWER HERE

Introduction:

In this project we aim to analyze the quanteda's builtin dataset named data_corpus_irishbudget2010. It is the collection of Irish budget speeches from the year 2010. The main variables of key intrest here are the textual data(including the entire corpus), party etc. The main focus is on the themes discussed i these documents and identify and patterns. To acheive this we have explored

1. Pre-processing
2. Applying dictionary (To check how a sentiment dictionary (Laver-Garry). be useful here) and EDA
3. Multinational Classification using Naive Bayes

Research Questions:

1. What are the pre-dominant topics discussed in the irish parlament related to 2010 budget?
2. To what extent do different political parties use populist rhetoric in their discourse, as identified by the Laver-Garry sentiment dictionary?
3. Relation between tf-idf, weighted proportions and dictionary lookup?
4. Is it okay to apply supervised learning algorithms like classification on every dataset, what are. some possible limitations and effects on evaluation metrics?

1) Importing:

```
library(ggplot2)
library(quanteda.textplots)
library(quanteda.textmodels)
library(reshape2)
library(quanteda)
## Package version: 3.3.1
## Unicode version: 14.0
## ICU version: 71.1
## Parallel computing: 8 of 8 threads used.
## See https://quanteda.io for tutorials and examples.


#importing data
data(data_corpus_irishbudget2010)
data_corpus <- data_corpus_irishbudget2010
summary(data_corpus_irishbudget2010, 3)
## Corpus consisting of 14 documents, showing 3 documents:
##
##                  Text Types Tokens Sentences year debate number   foren    name
##   Lenihan, Brian (FF)  1953   8641       374 2010 BUDGET     01   Brian Lenihan
##  Bruton, Richard (FG)  1040   4446       217 2010 BUDGET     02 Richard  Bruton
##    Burton, Joan (LAB)  1624   6393       309 2010 BUDGET     03    Joan  Burton
##  party
##     FF
##     FG
##    LAB
#downloading dictionary
download.file('https://raw.githubusercontent.com/quanteda/tutorials.quanteda.io/maste
r/content/dictionary/laver-garry.cat','LaverGarry.cat')
dict_lg <- dictionary(file = "LaverGarry.cat", format = "wordstat")
```

2) pre-processing data

From the below word cloud, we possiblly can infer the answer to the First research question. According to that word cloud, The data seems to have good quality after pre-processing. Further, The topics which were discussed can be

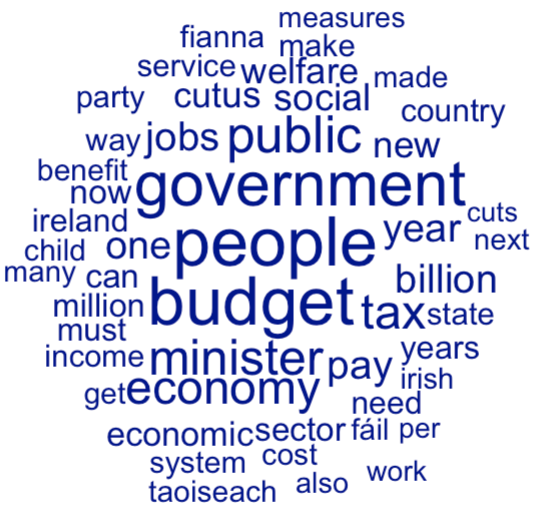
a. Modification of taxation policy (tax)

b. Adjustments to the ireland's flagship child benefit programme. (child)

c. Improved focus on Jobs and welfare schemes etc.

```
#pre-processing data
irish_tokens <- tokens(data_corpus, remove_punct = TRUE, remove_url=TRUE, remove_numb
ers=TRUE, remove_symbols=TRUE) # remove punctuation, urls, numbers, symbols
irish_dfm <- tokens_wordstem(irish_tokens)
irish_dfm <- dfm_remove(dfm(irish_tokens, tolower=TRUE), c(stopwords("english"), "htt
ps", "http")) # consider lower case only, remove stopwords and custom stopwords

#plotting word-cloud to see how data cleaning worked and to make modifications and in
ferences.
textplot_wordcloud(irish_dfm, rotation=0, min_size=.75, max_size=3, max_words=50)
```



3) Tf-idf scores and Dictionary

As we know Tf-idf scores are very crucial in identifying the most important and unique words. We can clearly see that the scores for ECONOMY is very high which means it is the most important as well as rare accross the corpus. Similarly, The Groups and rural, urban featurres have very low scores so, they are not so important.

```
# Calculating tf-idf scores
irish_dfm_tfidf <- dfm_tfidf(irish_dfm)


#applying dictionary
irish_dict_lg <- dfm_lookup(irish_dfm, dictionary = dict_lg, levels = 1)
print(irish_dict_lg)
## Document-feature matrix of: 14 documents, 9 features (19.84% sparse) and 6 docvar
s.
##                       features
## docs                   CULTURE ECONOMY ENVIRONMENT GROUPS INSTITUTIONS
##   Lenihan, Brian (FF)        9     582          21      0           93
##   Bruton, Richard (FG)      35     199           5      0           95
##   Burton, Joan (LAB)        33     399           6      3           84
##   Morgan, Arthur (SF)       56     425          10      0           63
##   Cowen, Brian (FF)         16     415          24      0           63
##   Kenny, Enda (FG)          26     210           8      1           53
##                       features
## docs                   LAW_AND_ORDER RURAL URBAN VALUES
##   Lenihan, Brian (FF)             11     9     0     19
##   Bruton, Richard (FG)            14     0     0     14
##   Burton, Joan (LAB)               6     2     3      6
##   Morgan, Arthur (SF)             22     2     1     18
##   Cowen, Brian (FF)                4     8     1     13
##   Kenny, Enda (FG)                18     0     2      8
## [ reached max_ndoc ... 8 more documents ]
```

It is very crucial to choose an important dictionary depending on the domain of research. So, The Laver-Garry dictionary was picked, which is known for its relioability backed by research. We can further able to answer to the question 2 of “To what extent do different political parties use populist rhetoric in their discourse, as identified by the Laver-Garry sentiment dictionary?”. So,

Frequency of Populist Rhetoric Features:

- Populist rhetoric features such as “CULTURE.CULTURE-POPULAR” have non-zero values for some parties, indicating the presence of these themes in their discourse.
- For example, the “CULTURE.CULTURE-POPULAR” feature has low frequencies across all parties, suggesting that discussions around popular culture are not dominant in their discourse.

Variation Across Parties:

- There is variation in the frequency of populist rhetoric features among different parties. For instance, “CULTURE.CULTURE-POPULAR” has low frequencies for all parties, but there are slight differences in the exact values.
- Parties may have different priorities or strategies in utilising populist rhetoric, leading to this variation.

```
# Assuming dfmat_irish_lg is your document-feature matrix after applying Laver-Garry
dictionary
# Let's convert it to a data frame for easier plotting
lg_df <- as.data.frame(as.matrix(irish_dict_lg))

# Adding document names as a column
lg_df$Document <- rownames(lg_df)

# Group the DFM by screen name
grouped_dfm <- dfm_group(irish_dfm, groups = data_corpus_irishbudget2010$party)
# Weight the DFM appropriately
weighted_dfm <- dfm_weight(grouped_dfm, scheme = "prop")

#identifying the extent to which each candidate uses populist rhetoric on Twitter
dfm_lookup(weighted_dfm,dictionary = dict_lg )
## Document-feature matrix of: 5 documents, 20 features (26.00% sparse) and 3 docvar
s.
##        features
## docs    CULTURE.CULTURE-HIGH CULTURE.CULTURE-POPULAR CULTURE.SPORT     CULTURE
##   FF            0.0002796812           0                       0 0.003216333
##   FG            0.0002173913           0                       0 0.016956522
##   Green         0                      0                       0 0.008797654
##   LAB           0.0001728608           0.0001728608            0 0.011927398
##   SF            0.0002018571           0.0008074283            0 0.016350424
##        features
## docs    ECONOMY.+STATE+ ECONOMY.=STATE= ECONOMY.-STATE-
##   FF         0.02894700      0.08376451      0.02684939
##   FG         0.02413043      0.07695652      0.01413043
##   Green      0.02521994      0.06803519      0.01114370
##   LAB        0.04062230      0.07882455      0.01780467
##   SF         0.04178442      0.08841340      0.01413000
##        features
## docs    ENVIRONMENT.CON ENVIRONMENT ENVIRONMENT.PRO ENVIRONMENT GROUPS.ETHNIC
##   FF                  0.0015382464                0.004754580            0
##   FG                  0.0013043478                0.001739130            0
##   Green               0                           0.012903226            0
##   LAB                 0.0001728608                0.001901469            0
##   SF                  0.0004037142                0.002825999            0
## [ reached max_nfeat ... 10 more features ]
```

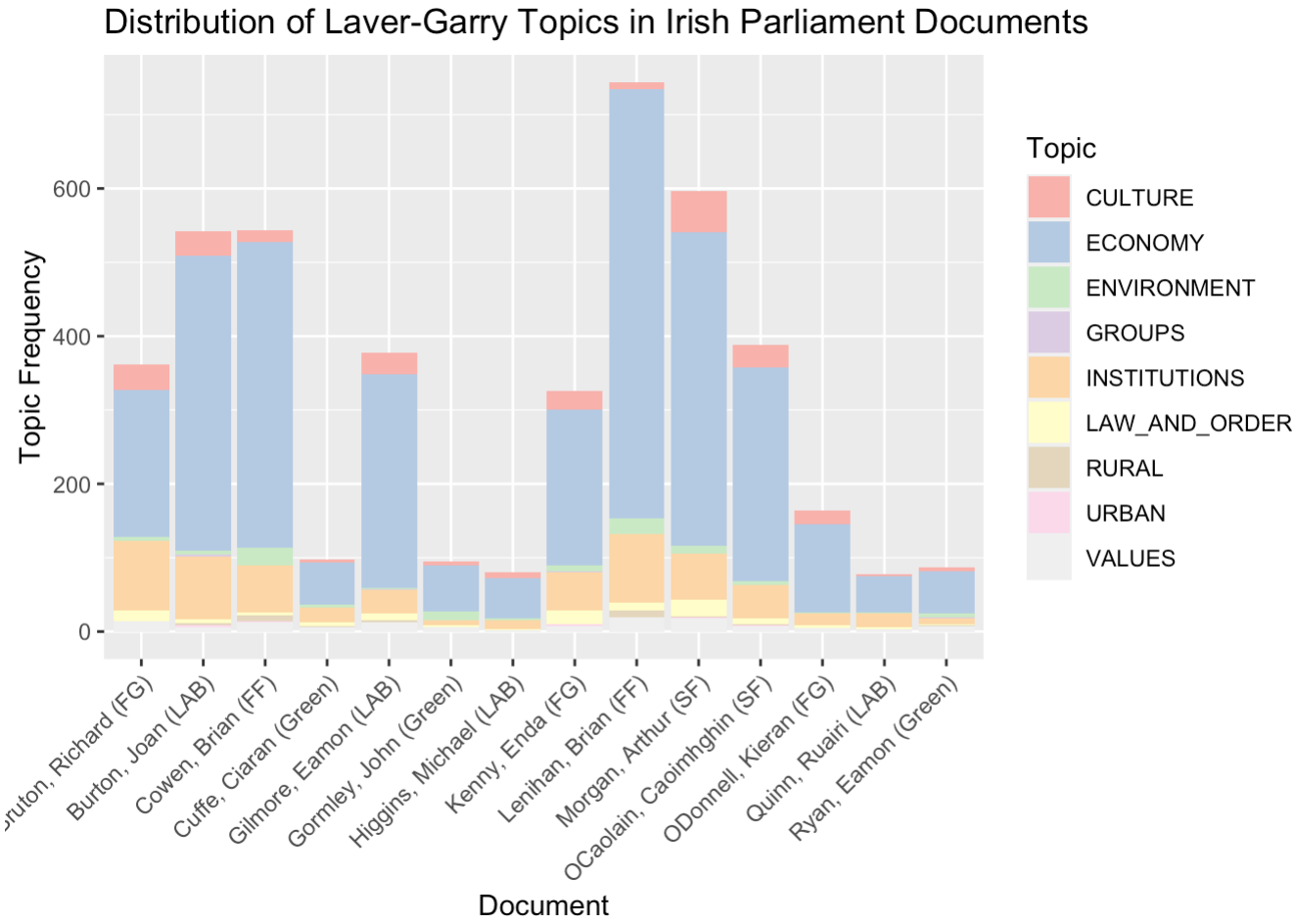
4) EDA

We can clearly understand that a majoriity of the topic s which were discussed are related to the Economy, posibly related to Economic slowdown and need to boost it. Followed by Institutions and law and order. Possibly related to the Improving the Government institutions to uphold law and order.

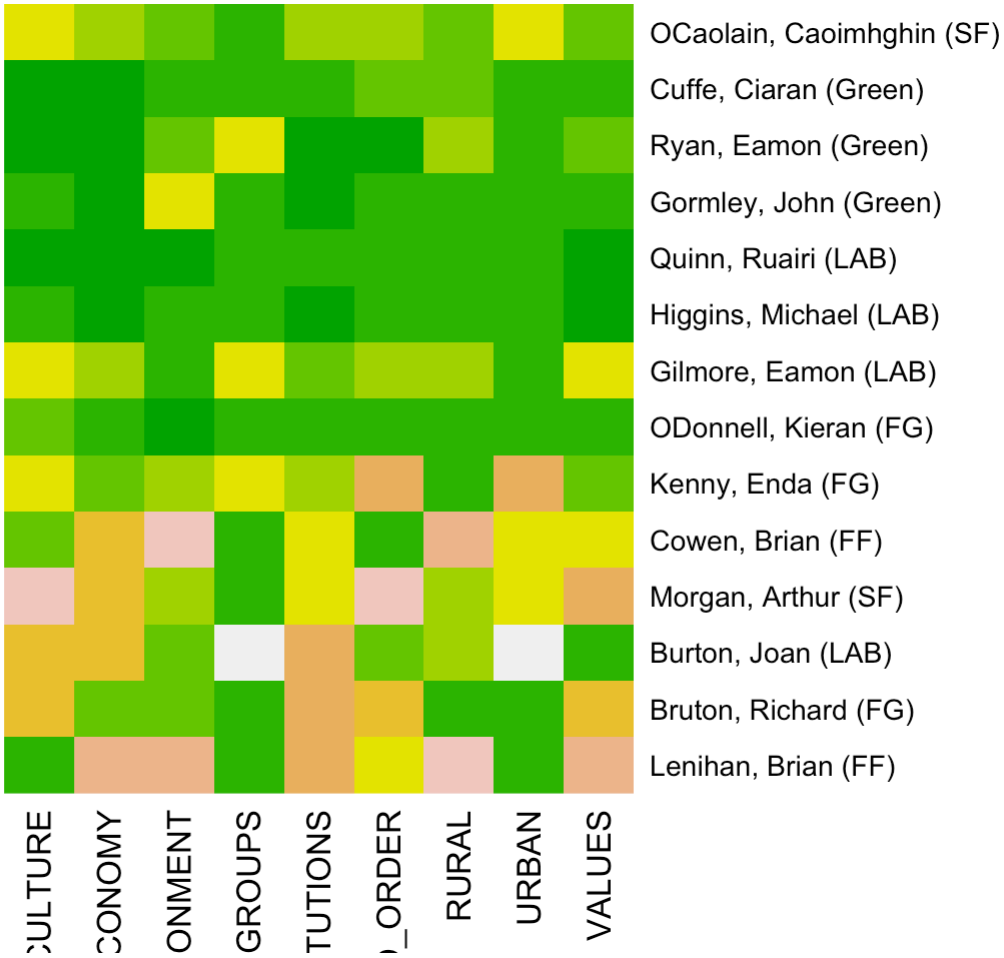
```
# Adding document names as a column
lg_df$Document <- rownames(lg_df)

###Used Chatgpt to enhance the code for stsacked plot to represent these docs.
# Melt the data frame to long format for plotting
library(reshape2)
dfmat_melted <- melt(lg_df, id.vars = "Document", variable.name = "Topic", value.name
= "Frequency")

# Plotting
ggplot(dfmat_melted, aes(Document, Frequency, fill = Topic)) +
  geom_bar(stat = "identity") +
  scale_fill_brewer(palette = "Pastel1") +
  ggtitle("Distribution of Laver-Garry Topics in Irish Parliament Documents") +
  xlab("Document") +
  ylab("Topic Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotating x-axis labels
for better readability
```




```
#ploting heatmap
heatmap(as.matrix(irish_dict_lg), Rowv=NA, Colv=NA, col = terrain.colors(10), scale
="column", margins=c(5,10))
```



5)Naive Bayes Classification:

```
#library(quanteda)
#library(quanteda.textmodels)

# loading data
#data_corpus <- data_corpus_irishbudget2010
summary(data_corpus, 5)
## Corpus consisting of 14 documents, showing 5 documents:
##
##                  Text Types Tokens Sentences year debate number   foren    name
##   Lenihan, Brian (FF)  1953   8641       374 2010 BUDGET     01   Brian Lenihan
##  Bruton, Richard (FG)  1040   4446       217 2010 BUDGET     02 Richard  Bruton
##    Burton, Joan (LAB)  1624   6393       309 2010 BUDGET     03    Joan  Burton
##   Morgan, Arthur (SF)  1595   7107       344 2010 BUDGET     04  Arthur  Morgan
##     Cowen, Brian (FF)  1629   6599       251 2010 BUDGET     05   Brian   Cowen
##  party
##     FF
##     FG
##    LAB
##     SF
##     FF
```

Data split

```
# Splitting the data into training and test sets of 70% and 30%
set.seed(123)
smp <- sample(c("train", "test"), size = ndoc(data_corpus),
              prob = c(0.70, 0.30), replace = TRUE)
train <- which(smp == "train")
test <- which(smp == "test")
```

Pre-processing

```
# Tokenizing and creating DFM
irish_tokens <- tokens(data_corpus, remove_punct = TRUE, remove_url = TRUE,
                       remove_symbols = TRUE, remove_numbers = TRUE, verbose = TRUE)
## Creating a tokens object from a corpus input...
##  ...starting tokenization
##  ...preserving hyphens
##  ...preserving social media tags (#, @)
##  ...tokenizing 1 of 1 blocks
##  ...segmenting into words
##  ...5,590 unique types
##  ...removing separators, punctuation, symbols, numbers, URLs
##  ...complete, elapsed time: 0.079 seconds.
## Finished constructing tokens from 14 documents.
irish_tokens <- tokens_remove(irish_tokens, stopwords("english"))
irish_tokens <- tokens_wordstem(irish_tokens)
irish_dfm <- dfm(irish_tokens)

# Setting minimum occurrences as 2 docs
irish_dfm <- dfm_trim(irish_dfm, min_docfreq = 2, verbose = TRUE)
## Removing features occurring:
##   - in fewer than 2 documents: 1,536
##   Total features removed: 1,536 (47.0%).
```

Model Training

```
# Training Naive Bayes model
nb <- textmodel_nb(irish_dfm[train, ], docvars(irish_dfm, "party")[train],distributio
n="multinomial")

# Predicting labels for the test set
preds <- predict(nb, newdata = irish_dfm[test, ])

# Computing the confusion matrix
cm <- table(preds, docvars(irish_dfm, "party")[test])
cm
##
## preds   FF FG Green LAB SF
##   FF     1  0     1   0  0
##   FG     0  0     0   0  0
##   Green  0  0     0   0  0
##   LAB    0  1     0   1  0
##   SF     0  0     0   0  1
```

Evaluating model performance

After training the naive bayes classifier, We can see that despite accuracy being 0.6 and occurance of some Nans and 0's for many cases of precision and recall. This is used to answer the research question 4. Which upholds the importance of having more documents. In this case we just have 12 documents. Which might have potentially lead to the overfitting of the data.

Further, It is not ideal to perform classification on this limited dataset. Alternatively we can perform it on similar speeches from past years into account.

```
calculate_metrics <- function(conf_matrix) {

  # Calculate precision
  precision <- diag(conf_matrix) / rowSums(conf_matrix)

  # Calculate accuracy
  accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

  # Calculate recall
  recall <- diag(conf_matrix) / colSums(conf_matrix)

  # Return results
  return(list(precision = precision, accuracy = accuracy, recall = recall))
}
calculate_metrics(cm)
## $precision
##    FF    FG Green   LAB    SF
##   0.5   NaN   NaN   0.5   1.0
##
## $accuracy
## [1] 0.6
##
## $recall
##    FF    FG Green   LAB    SF
##     1     0     0     1     1
```

Conclusion:

In conclusion, When the data is limited It is very much ideal to perform basic data analysis and to identify trends and patterns rather than employing algorithms for predicting and clustering which are based on document level. We can enhance this further by employing keyness, scaling like wordfish etc which might take the features into account primarily.