Sentiment Analysis–Donor

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
library(tm)
```

```
## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```r
#myData <- fread('opendata_essays000.gz')
```

```r
df<-read_rds('projects_with_text.rds')
```

```r
#set.seed(1234)
#rows <- sample(nrow(df))
#df2 <- df[rows, ]
#df3<- df2[1:50000,]
```

```r
df_small<-df %>%
  select(`_projectid`, essay)
```

Put in lowercase

```r
df_small$essay<- tolower(df_small$essay)
```

CORPUS METHOD: In order to create a corpus of these essays, we need to use the Corpus function within the tm package:

```r
df_essay <- VCorpus(VectorSource(as.vector(df_small$essay)))
```

To remove punctuation marks within a Corpus object, we use this code:

```r
df_essay  <- tm_map(df_essay , removeWords, stopwords("english"))
```

```r
df_essay <- tm_map(df_essay, content_transformer(removePunctuation))
df_essay  <- tm_map(df_essay, content_transformer(removeNumbers))
```

stemming:

```r
#df_essay  <- tm_map(df_essay, content_transformer(stemDocument), language = "english")
#dictCorpus <- df_essay
```

Document term matrix to mathc with dictionary

```r
dictionary = get_sentiments("nrc")
dictionary = dictionary$word
```

```r
#veamos_term_matrix <- inspect(DocumentTermMatrix(df_essay, list(dictionary = dictionary)))
```

```r
#DocumentTermMatrix(v, control = list(dictionary = my_terms)) %>% as.matrix()
DocumentTerm = DocumentTermMatrix(df_essay, list(dictionary = dictionary))

#Doc_term_matrix <- as.matrix(DocumentTerm)
```

```r
documents_dtm_rm_sparse <- removeSparseTerms(DocumentTerm, 0.9)

#Doc_term_matrix_sin_sparse <- as.matrix(documents_dtm_rm_sparse)
```

```r
indices_matrix = df_small$`_projectid`


documents_dtm_rm_sparse$dimnames$Docs <- indices_matrix
```

```r
library(dplyr)
library(tidytext)

ap_td <- tidy(documents_dtm_rm_sparse)

ap_sentiments <- ap_td %>%
  inner_join(get_sentiments("nrc"), by = c(term = "word"))
```

```r
#library(tidyr)

#ap_sentiments %>%
#  filter(sentiment %in% c("positive","negative")) %>%
#  count(document, sentiment, wt = count) %>%
#  spread(sentiment, n, fill = 0) %>%
#  mutate(sentiment = positive - negative) %>%
#  arrange(sentiment)

sentimients_por_essay <- ap_sentiments %>%
  #filter(sentiment %in% c("positive","negative")) %>%
  count(document, sentiment, wt = count) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```r
#Calculate TF-IDF
tidyReviewsTFIDF <- ap_td %>%
  #filter(review_id %in% tidyReviewsLong$review_id) %>%
  bind_tf_idf(term,document,count) %>%
  group_by(document)
```

```r
tf_idf_values_per_essay <- tidyReviewsTFIDF %>%
  select(document,term,tf_idf) %>%
  #filter(sentiment %in% c("positive","negative")) %>%
  #count(document, sentiment, wt = count) %>%
  spread(term,tf_idf, fill = 0)
```

```r
text_features <- df_small %>%
  select(`_projectid`)



text_features <- text_features %>%
  left_join(sentimients_por_essay, by = c("_projectid" = "document"))

text_features <- text_features %>%
  left_join(tf_idf_values_per_essay, by = c("_projectid" = "document"))


write.csv(text_features, file = "text_feats.csv")
```