Coffee Break Experiment 1

Clare Cruz

Introduction

When consuming today's news, it is rare to find an article that is not filled with dismal facts and unfortunate breaking news. But, it is unclear if all news is depressing since very few people read all the news, especially across multiple news outlets. The goal of this study is to answer this curiosity by examining the content of today's news to see how much negative content there is and if the amount of negative content is higher compared to other types of texts. More specifically, this study aims to answer:

- Are there more negatively associated words than positive ones in sample news data?
- Is the difference in the negatively associated words between the news and other texts meaningful?

Data

The news data come from a random sample of the Corpus of Contemporary American English. The subsample is balanced for text type and year, with 400 texts total and 50 texts for each of the 8 text types. Tokens are defined as lowercased, non-numeric texts that are surrounded by spaces without punctuation. Common compound expressions are also considered tokens. There is a total of 1,000,887 tokens in the sample corpus, with 119,029 tokens in the news category (Table 1).

Table 1: Composition of the sample corpus.

Text-Type	Texts	Tokens
Academic	50	121442
Blog	50	125492
Fiction	50	128644
Magazine	50	126631
News	50	119029
Spoken	50	127156
Television/Movies	50	128191
Web	50	124302
Total	**400**	**1000887**

Additionally, the two sentiment lists were collected from a very popular sentiment corpus in the QDAP dictionaries R package. The lists contain words and common phrases that were mined from reviews, forum discussions, and blogs to capture regular and comparative opinions. There are 2,003 positive words and 4,776 negative words included in the sentiment lists (Table 2).

Table 2: Composition of the sentiment lists.

Sentiment	List Size	Sample Tokens
Positive	2003	a plus, abound, abounds, abundance, abundant, accessable
Negative	4776	abnormal, abolish, abominable, abominably, abominate, abomination

Methods

To determine if the overall composition of the news is negative, a combination of descriptive statistics and statistical inference is utilized. First, the news' corpus is examined by calculating the absolute frequency and deviation of proportions (DP) dispersion value. The absolute frequency provides a fair comparison between the count of sentiment tokens within the news texts. Then, the DP's dispersion metric will be averaged to represent the overall distribution of sentiment words.

Next, the log-likelihood or goodness of fit test is utilized to see if the difference between the negative tokens in the news corpus and the other texts is significant. The log-likelihood metric is the best metric to use in this situation since the size of the news corpus will differ from the corpus of the other texts. Specifically, the average log-likelihood value is calculated so that the difference for every negative token will be summarized.

Lastly, two existing sentiment lists will be brought into the data to capture the negative and positive words. These lists were chosen for their concise coverage and accessibility to analyze the sentiment of the texts.

Results

To start, descriptive statistics were used to compare the amount of positive and negative words within the news texts. In the news texts, the absolute frequency of the unique negative words is higher than the positive words, with 1,027 negative tokens and 669 positive tokens. Additionally, the overall frequency for the positive and negative words was low, with 89% and 83% of the negative and positive tokens occurring less than five times, respectively (Figure 1).

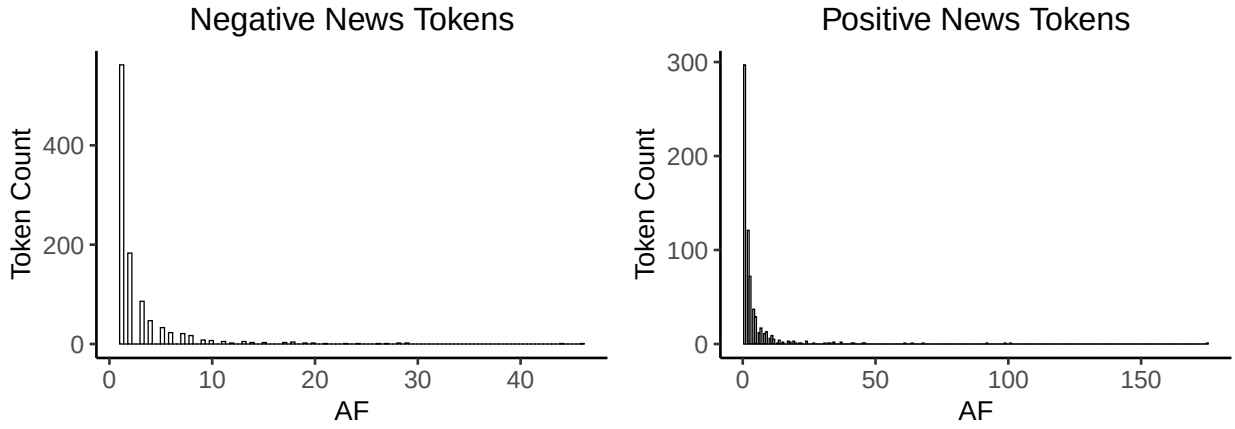

Figure 1: Histograms of the absolute frequency for the positive and negative tokens in the news data.

However, further analysis showed that the comparison between the positive and negative words is biased since many positive words are likely utilized without sentiment. For example, the positive token with the

highest absolute frequency, 'like', has multiple functions and is likely to be unrelated to sentiment in most cases (Table 3).

Table 3: Positive tokens with the highest absolute frequency in the news texts.

Positive Token	AF News
like	175
free	101
work	99
best	92
right	68
good	64

Next, the average log-likelihood was used to determine if there was a meaningful difference in the frequency of negative tokens between the news texts and other texts. The average log-likelihood for the negative tokens is -0.15 with an average p-value of 0.41 (Table 4). This result indicates that while the negative tokens occur more often in the other texts than in the news texts, the difference is not significant. Additionally, the average DP dispersion metric for the news texts is close to 1 which means that the negative tokens are not dispersed or consistent in the texts.

Table 4: Aggregate statistics for the negative tokens in the news text-type when compared to the other text-types.

	Value
Log Likelihood	-0.15
Log Ratio	0.80
P-value	0.41
AF News	0.82
AF Other Texts	0.96
DP News	0.96
DP Other Texts	0.98

Discussion

The descriptive analysis showed that there are more negatively associated words than positive ones in the news texts. Then, the inferential statistics proved that there is no meaningful difference in the frequency of negative words between news texts and other text types. While the first result aligns with the initial intuition behind the research question, the second result proved to be a surprise. Perhaps the content of the news is inherently depressing but when it comes to the reporting it remains objective. This hypothesis aligns with the results since negative objects like 'death' were frequent while adjectives like 'depressing' did not occur.

Since this study is a casual experimentation, there are many limitations for sake of curiosity. First, the news data in this sample is small compared to other corpora. Also, the result of the study is heavily dependent upon the sentiment lists, so another more robust list would likely produce a different result. Next, the context of the tokens and their sentiment was not considered due to the scope of the project. This limitation caused some words to be misinterpreted, so the comparison between positive and negative words is likely inaccurate. Lastly, it is unclear if aggregating the log-likelihood and p-value is a valid method. In most situations, these metrics are evaluated on a row-by-row basis. But for the context of this study, an aggregation was necessary to summarize all negative tokens.

Code Appendix

```
library(cmu.textstat)
library(tidyverse)
library(quanteda)
library(quanteda.textstats)
library(qdapDictionaries)
library(scales)
library(ggplot2)
library("ggpubr")
## Pre-process the data & create a corpus

sc <- sample_corpus %>%
  mutate(text = preprocess_text(text)) %>%
  corpus()
## Extract meta-data from file names

doc_categories <- sample_corpus %>%
  dplyr::select(doc_id) %>%
  mutate(doc_id = str_extract(doc_id, "^[a-z]+")) %>%
  rename(text_type = doc_id)
## Assign the meta-data to the corpus

docvars(sc) <- doc_categories
#doc_categories
## Create a **dfm**

sc_dfm <- sc %>%
  tokens(what="fastestword", remove_numbers=TRUE) %>%
  tokens_compound(pattern = phrase(multiword_expressions)) %>%
  dfm()

#sc_dfm
corpus_comp <- ntoken(sc_dfm) %>%
  data.frame(Tokens = .) %>%
  rownames_to_column("Text_Type") %>%
  mutate(Text_Type = str_extract(Text_Type, "^[a-z]+")) %>%
  group_by(Text_Type) %>%
  summarize(Texts = n(),
    Tokens = sum(Tokens)) %>%
  mutate(Text_Type = c("Academic", "Blog", "Fiction", "Magazine", "News", "Spoken", "Television/Movies"
  rename("Text-Type" = Text_Type) %>%
  janitor::adorn_totals()
kableExtra::kbl(corpus_comp, caption = "Composition of the sample corpus.", booktabs = T, linesep = "")
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic() %>%
  kableExtra::row_spec(8, hline_after = TRUE) %>%
  kableExtra::row_spec(9, bold=T)
# Sentiment Lists from the qdap dictionaries package
data(positive.words)
data(negative.words)

positive <- positive.words
```

```r
negative <- negative.words

#head(positive)
#head(negative)
sentiment <- data.frame(c('Positive','Negative'),
                        c(length(positive), length(negative)),
                        c(paste(head(positive), collapse=', ' ) , paste(head(negative), collapse=', ' ))
colnames(sentiment) <- c('Sentiment', 'List Size','Sample Tokens')

kableExtra::kbl(sentiment, caption = "Composition of the sentiment lists.", booktabs = T, linesep = "")
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
# Keyness Calculations
news_dfm <- dfm_subset(sc_dfm, text_type == "news")
notnews_dfm <- dfm_subset(sc_dfm, text_type != "news")

news_kw <- keyness_table(news_dfm, notnews_dfm)
# Keyness of interest
positive_news_kw <- news_kw %>% filter(Token %in% positive)
negative_news_kw <- news_kw %>% filter(Token %in% negative)

# Since the keyness table includes words from target and reference texts, filter to just the target tok
negative_news <- negative_news_kw %>% filter(AF_Tar > 0)
positive_news <- positive_news_kw %>% filter(AF_Tar > 0)
bin_width <- function(x){
  2 * IQR(x) / length(x)^(1/3)
  }
negative_hist <- ggplot(negative_news,aes(x = AF_Tar)) +
  geom_histogram(binwidth = bin_width(negative_news$AF_Tar), colour="black", fill="white",  size=.25) +
  theme_classic() +
  theme(axis.text = element_text(size=10)) +
  ggtitle('Negative News Tokens')+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab('Token Count')+
  xlab("AF")

positive_hist <- ggplot(positive_news,aes(x = AF_Tar)) +
  geom_histogram(binwidth = bin_width(positive_news$AF_Tar), colour="black", fill="white", size=.25) +
  theme_classic() +
  theme(axis.text = element_text(size=10)) +
  ggtitle('Positive News Tokens')+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab('Token Count')+
  xlab("AF")

ggarrange(negative_hist,positive_hist,ncol = 2)
# sentiment <- data.frame(
#   c(length(positive),dim(positive_news)[1],percent(round(dim(positive_news)[1]/length(positive),1))),
#   c(length(negative), dim(negative_news)[1], percent(round(dim(negative_news)[1]/length(negative),1))),
#   row.names = c('Sentiment List Size', 'Total Sentiment Tokens in Corpus', 'Proportion of Sentiment T
#
# colnames(sentiment) <- c('Positive','Negative')
#
```

```r
# kableExtra::kbl(sentiment, caption = "Distribution of the sentiment lists in the news corpus.", bookt
#   kableExtra::kable_styling(latex_options = "HOLD_position") %>%
#   kableExtra::kable_classic()
# Calculating the average LL and DP, map the lapply output since it returns a list
agg_stats <- lapply(negative_news_kw, mean)
agg_stats_avg <- map_df(agg_stats, ~as.data.frame(t(.)))
agg_stats_trim <- agg_stats_avg[c(-1,-5,-6,-9,-10),]

# Since DP has na's the mean is calculated separately
agg_tar_dp <- mean(negative_news_kw$DP_Tar, na.rm = TRUE)
agg_ref_dp <- mean(negative_news_kw$DP_Ref, na.rm = TRUE)

agg_stats_full <- c(agg_stats_trim,agg_tar_dp, agg_ref_dp)
agg_stats_df <- data.frame(agg_stats_full, row.names = c("Log Likelihood","Log Ratio","P-value","AF News
colnames(agg_stats_df) <- c('Value')
outliers <- positive_news %>% arrange(desc(AF_Tar)) %>% select(Token,AF_Tar)
colnames(outliers) <- c('Positive Token','AF News')
kableExtra::kbl(head(outliers), caption = "Positive tokens with the highest absolute frequency in the ne
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
kableExtra::kbl(agg_stats_df, caption = "Aggregate statistics for the negative tokens in the news text-
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```