apple's_10k

sirius_ife

2024-03-09

Case Study: Analyzing Apple's 10-K Reports Using Text Analysis

Introduction

In this case study, we'll analyze Apple's 10-K reports from the years 2020 to 2023 using text analysis techniques. We aim to gain insights into Apple's financial sentiment, track changes over time, and visualize key trends in the reports.

Setup and Data Preparation

We begin by loading necessary packages for text analysis and reading the 10-K report files for each year.

```
# Load necessary packages
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2

## Warning: package 'tidyr' was built under R version 4.3.2

## Warning: package 'readr' was built under R version 4.3.2

## Warning: package 'dplyr' was built under R version 4.3.2

## Warning: package 'stringr' was built under R version 4.3.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.3.2
```

```
library(SnowballC)
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.3.3
```

```
## Loading required package: RColorBrewer
```

```
library(Rcpp)
```

```
## Warning: package 'Rcpp' was built under R version 4.3.2
```

```
# Read 10-K report files for each year
file_2020 <- "2020.txt"
apple_2020 <- readChar(file_2020, file.info(file_2020)$size)

file_2021 <- "2021.txt"
apple_2021 <- readChar(file_2021, file.info(file_2021)$size)

file_2022 <- "2022.txt"
apple_2022 <- readChar(file_2022, file.info(file_2022)$size)

file_2023 <- "2023.txt"
apple_2023 <- readChar(file_2023, file.info(file_2023)$size)

# Load custom stopwords
custom_stop_words <- read_csv("stop_words_list.csv", col_names = FALSE)
```

```
## Rows: 1087 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): X1
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Load finance sentiment list
lm_dict <- tidytext::get_sentiments('loughran')
```

Data Processing

Next, we preprocess the text data, tokenize it into words and sentences, remove stop words, and add sentiment labels.

```r
# Preprocess and tokenize the text data
apple2020 <- tibble(apple_2020) %>%
  unnest_tokens(sentence, apple_2020, token = 'sentences') %>%
  mutate(sentence_num = row_number(), call = 'apple_2020') %>%
  unnest_tokens(word, sentence, token = 'words') %>%
  mutate(word_num = row_number()) %>%
  anti_join(custom_stop_words, by = c('word' = 'X1')) %>%
  inner_join(lm_dict, by = 'word')
```

```
## Warning in inner_join(., lm_dict, by = "word"): Detected an unexpected many-to-many relationship betw
## i Row 188 of `x` matches multiple rows in `y`.
## i Row 73 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
# Repeat the same process for other years (2021, 2022, 2023)
apple2021 <- tibble(apple_2021) %>%
  unnest_tokens(sentence, apple_2021, token='sentences') %>%
  mutate(sentence_num = row_number(), call = 'apple_2021') %>%
  unnest_tokens(word, sentence) %>%
  mutate(word_num = row_number()) %>%
  anti_join(custom_stop_words, by=c('word' = 'X1')) %>%
  inner_join(lm_dict)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in inner_join(., lm_dict): Detected an unexpected many-to-many relationship between `x` and
## i Row 204 of `x` matches multiple rows in `y`.
## i Row 873 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
apple2022 <- tibble(apple_2022) %>%
  unnest_tokens(sentence, apple_2022, token='sentences') %>%
  mutate(sentence_num = row_number(), call = 'apple_2022') %>%
  unnest_tokens(word, sentence) %>%
  mutate(word_num = row_number()) %>%
  anti_join(custom_stop_words, by=c('word' = 'X1')) %>%
  inner_join(lm_dict)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in inner_join(., lm_dict): Detected an unexpected many-to-many relationship between `x` and
## i Row 1730 of `x` matches multiple rows in `y`.
## i Row 873 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
apple2023 <- tibble(apple_2023) %>%
  unnest_tokens(sentence, apple_2023, token='sentences') %>%
```

```
  mutate(sentence_num = row_number(), call = 'apple_2023') %>%
  unnest_tokens(word, sentence) %>%
  mutate(word_num = row_number()) %>%
  anti_join(custom_stop_words, by=c('word' = 'X1')) %>%
  inner_join(lm_dict)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in inner_join(., lm_dict): Detected an unexpected many-to-many relationship between `x` and
## i Row 1536 of `x` matches multiple rows in `y`.
## i Row 873 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
# Combine dataframes for all years
all_firms <- bind_rows(apple2020, apple2021, apple2022, apple2023)
```

Sentiment Analysis

We analyze the sentiment expressed in the reports and visualize the results.

```
# Calculate sentiment percentages for each year
apple20 <- all_firms %>%
  filter(call=='apple_2020') %>% #Just Amazon
  group_by(call, sentiment) %>%
  summarize(count = n(), #Count
            percent = count/(all_firms %>% filter(call=='apple_2020') %>% nrow())) #Percent for just Am
```

```
## `summarise()` has grouped output by 'call'. You can override using the
## `.groups` argument.
```

```
apple21 <- all_firms %>%
  filter(call=='apple_2021') %>%
  group_by(call, sentiment) %>%
  summarize(count = n(), percent = count/(all_firms %>% filter(call=='apple_2021') %>% nrow()))
```

```
## `summarise()` has grouped output by 'call'. You can override using the
## `.groups` argument.
```

```
apple22 <- all_firms %>%
  filter(call=='apple_2022') %>%
  group_by(call, sentiment) %>%
  summarize(count = n(), percent = count/(all_firms %>% filter(call=='apple_2022') %>% nrow()))
```

```
## `summarise()` has grouped output by 'call'. You can override using the
## `.groups` argument.
```

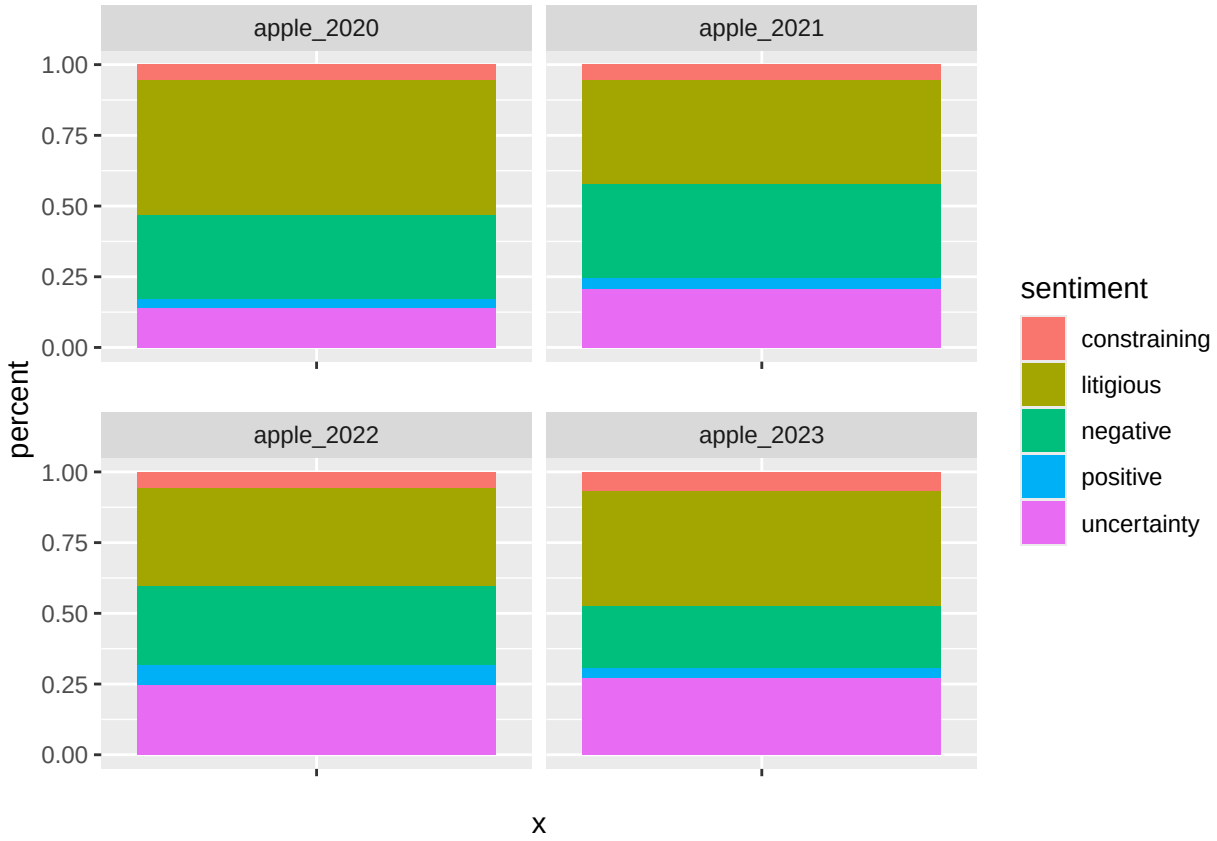
```
apple23 <- all_firms %>%
  filter(call=='apple_2023') %>%
  group_by(call, sentiment) %>%
  summarize(count = n(), percent = count/(all_firms %>% filter(call=='apple_2023') %>% nrow()))
```

```
## `summarise()` has grouped output by 'call'. You can override using the
## `.groups` argument.
```

```
# Plot sentiment percentages over the years
percentages <- bind_rows(apple20, apple21, apple22, apple23)
print(percentages)
```

```
## # A tibble: 20 x 4
## # Groups:   call [4]
##    call       sentiment    count percent
##    <chr>      <chr>        <int>   <dbl>
##  1 apple_2020 constraining     7  0.0547
##  2 apple_2020 litigious       61  0.477
##  3 apple_2020 negative        38  0.297
##  4 apple_2020 positive         4  0.0312
##  5 apple_2020 uncertainty     18  0.141
##  6 apple_2021 constraining     6  0.0541
##  7 apple_2021 litigious       41  0.369
##  8 apple_2021 negative        37  0.333
##  9 apple_2021 positive         4  0.0360
## 10 apple_2021 uncertainty     23  0.207
## 11 apple_2022 constraining     5  0.0562
## 12 apple_2022 litigious       31  0.348
## 13 apple_2022 negative        25  0.281
## 14 apple_2022 positive         6  0.0674
## 15 apple_2022 uncertainty     22  0.247
## 16 apple_2023 constraining     4  0.0678
## 17 apple_2023 litigious       24  0.407
## 18 apple_2023 negative        13  0.220
## 19 apple_2023 positive         2  0.0339
## 20 apple_2023 uncertainty     16  0.271
```

```
percentages %>%
  ggplot(aes(x='', y=percent, fill=sentiment)) +
  geom_bar(width=1, stat='identity') +
  facet_wrap(~call, ncol = 2, scales = "free_x")
```

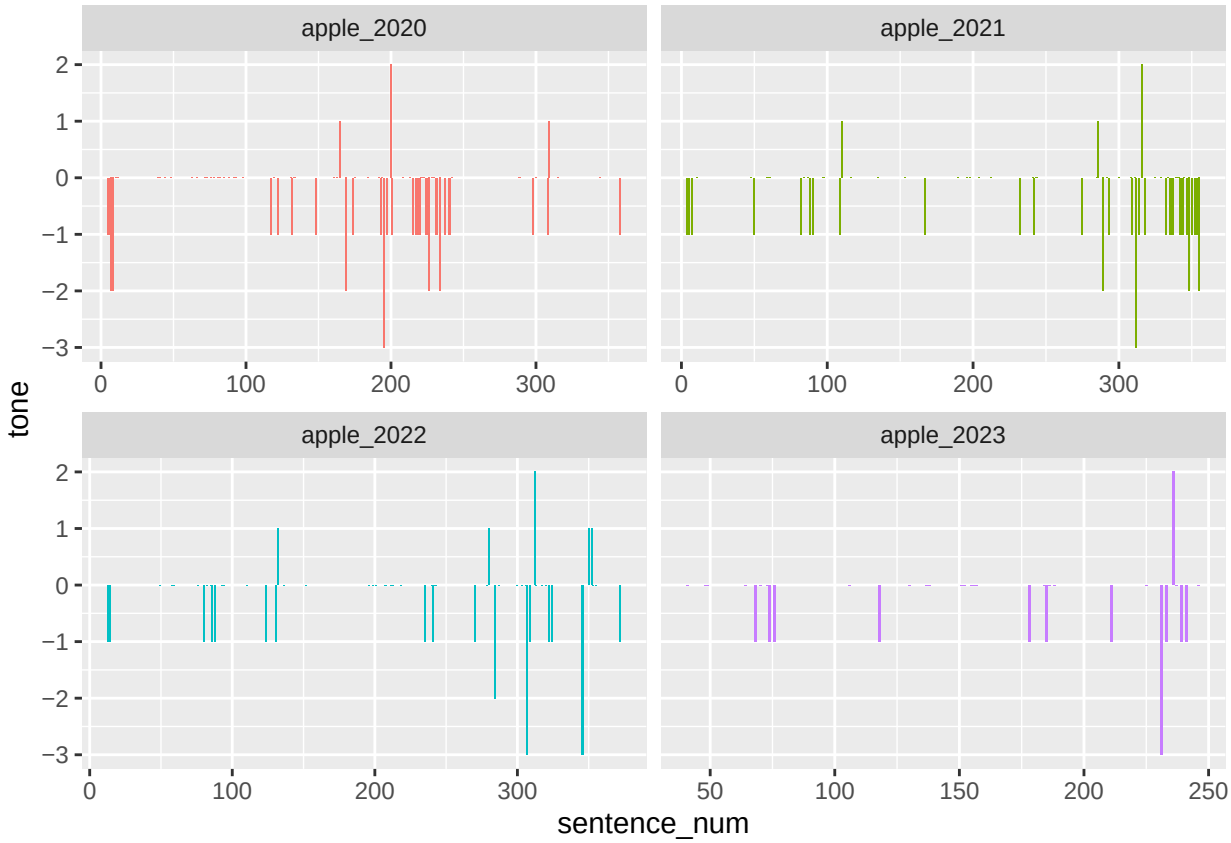


Sentiment Trends Over Time

We explore how sentiment varies over time within each year.

```
# Plot sentiment trends over time for each year
all_firms %>%
  group_by(call, sentence_num, sentiment) %>%
  summarize(n=n()) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>% #Transpose the data for the
  mutate(tone = positive - negative) %>% #Create "tone"
  ggplot(aes(x=sentence_num, y=tone, fill=call)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~call, ncol = 2, scales = "free_x")
```

```
## `summarise()` has grouped output by 'call', 'sentence_num'. You can override
## using the `.groups` argument.
```



Word Clouds

Finally, we generate word clouds to visualize the most frequent words in the reports for each year.

```
# Generate word clouds for each year
set.seed(77) #Seed for random number
cloud <- tibble(apple_2020) %>% #Create dataframe
  unnest_tokens(word, apple_2020) %>% #Word tokens
  anti_join(custom_stop_words, by=c('word' = 'X1')) %>% #Remove stop words
  group_by(word) %>%
  summarize(n = n()) %>%
  with(wordcloud(words=word, freq=n, min.freq=10, max.words=500, random.order=F, rot.per=0.30, colors=bl
```

```
## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## investment could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## management could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## statement could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## carrying could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## china could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## compensation could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## control could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## corporate could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## denominated could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## effective could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## general could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## iphone could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## maturities could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## payments could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## provision could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## significant could not be fit on page. It will not be plotted.
```


```
set.seed(77)
cloud <- tibble(apple_2021) %>%
  unnest_tokens(word, apple_2021) %>%
  anti_join(custom_stop_words, by=c('word' = 'X1')) %>%
  group_by(word) %>%
  summarize(n = n()) %>%
  with(wordcloud(words=word, freq=n, min.freq=10, max.words=500, random.order=F, rot.per=0.30, colors=bl
```

```
## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## payments could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## countries could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## geographic could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## manufacturing could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## material could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = word, freq = n, min.freq = 10, max.words = 500, :
## maturities could not be fit on page. It will not be plotted.
```


```
set.seed(77)
cloud <- tibble(apple_2022) %>%
  unnest_tokens(word, apple_2022) %>%
  anti_join(custom_stop_words, by=c('word' = 'X1')) %>%
  group_by(word) %>%
  summarize(n = n()) %>%
  with(wordcloud(words=word, freq=n, min.freq=15, max.words=500, random.order=F, rot.per=0.30, colors=bl
```


```
set.seed(77)
cloud <- tibble(apple_2023) %>%
  unnest_tokens(word, apple_2023) %>%
  anti_join(custom_stop_words, by=c('word' = 'X1')) %>%
  group_by(word) %>%
  summarize(n = n()) %>%
  with(wordcloud(words=word, freq=n, min.freq=26, max.words=500, random.order=F, rot.per=0.30, colors=bl
```

Conclusion

In conclusion, our analysis of Apple's 10-K reports using text analysis techniques has provided valuable insights into the company's financial sentiment, trends over time, and key themes. This information can be useful for investors, analysts, and stakeholders in understanding Apple's performance and strategic focus.