Vena Solutions Sentiment Lexicon

Alex Cookson (alexander.cookson@gmail.com)

2020-08-13

First, we set up our R environment with the necessary package and set a default theme for graphs.

```r
library(tidyverse) # Data manipulation
library(rvest) # Scraping
library(tidytext) # Text analysis
library(textstem) # Lemmatization
library(glmnet) # LASSO model
library(extrafont) # Additional fonts for graphs

# set default theme for graphs
theme_set(theme_minimal())
```

Next, we write a function to scrape the reviews from TrustRadius.com. This function scrapes only the "Pros and Cons" list, leaving other parts of the webpage alone.

```r
scrape_pros_cons <- function(url) {
  # Read website
  webpage <- read_html(url)

  # Plus/Minus
  icons <- webpage %>%
    html_nodes(".ugc .sprite") %>%
    html_attrs() %>%
    unlist() %>%
    as_tibble_col(column_name = "pro_con") %>%
    mutate(pro_con = str_remove(pro_con, "sprite sprite-proCon"))

  # Text
  text <- webpage %>%
    html_nodes(".ugc div:nth-child(2)") %>%
    html_text() %>%
    as_tibble_col(column_name = "text")

  # Merge
  return(bind_cols(icons, text))
}
```

Now, we execute the function and clean the output so that we have something tidy to work with. We also convert "Pro" or "Con" to a score (1 for "Pro", 0 for "Con"). Having a numerical score lets us apply LASSO regression, an approach we use here to create a "sentiment lexicon" of positive and negative terms associated with Vena Solutions' product.

```
pros_cons_raw <- tibble(base_url = "https://www.trustradius.com/products/vena-solutions/reviews?f=",
       review_num = seq(0, 125, by = 25)) %>%
  mutate(scrape_url = paste0(base_url, review_num)) %>%
  select(scrape_url) %>%
  mutate(data = map(scrape_url, possibly(scrape_pros_cons, NULL))) %>%
  unnest(data) %>%
  select(-scrape_url)

pros_cons <- pros_cons_raw %>%
  mutate(bullet_id = row_number(),
         score = ifelse(pro_con == "Plus", 1, 0)) %>%
  select(bullet_id, text, score)
```

We take the raw text of each pro/con bullet point and extract "bigrams" (two-word terms) from the text. We also get rid of "stop words" (e.g., "a", "the", "and"), which don't give us a lot of information because they are used so frequently.

```
bigrams <- pros_cons %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2) %>%
  separate(bigram, into = c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word,
         !is.na(word1),
         !is.na(word2)) %>%
  unite("bigram", word1, word2, sep = " ") %>%
  distinct(bullet_id, bigram, .keep_all = TRUE)
```

Now we get our data in the proper format for running LASSO regression. We create a sparse document-term matrix of bigrams and extract the pro/con score (1 or 0).

```
bigrams_clean <- bigrams %>%
  mutate(bigram_id = row_number()) %>%
  select(bigram_id, bigram, score)

bigram_matrix <- bigrams_clean %>%
  select(bigram_id, bigram) %>%
  cast_sparse(bigram_id, bigram)

ids <- as.integer(rownames(bigram_matrix))

ratings <- bigrams_clean$score[ids]
```

We fit a cross-validated LASSO model with 100 folds to create a list of terms (bigrams) and their estimated effect on the score.

```
cv_lasso <- cv.glmnet(bigram_matrix,
                      ratings,
                      nfolds = 100)

cv_lasso_tidy <- cv_lasso$glmnet.fit %>%
  tidy() %>%
  filter(lambda == cv_lasso$lambda.min)
```

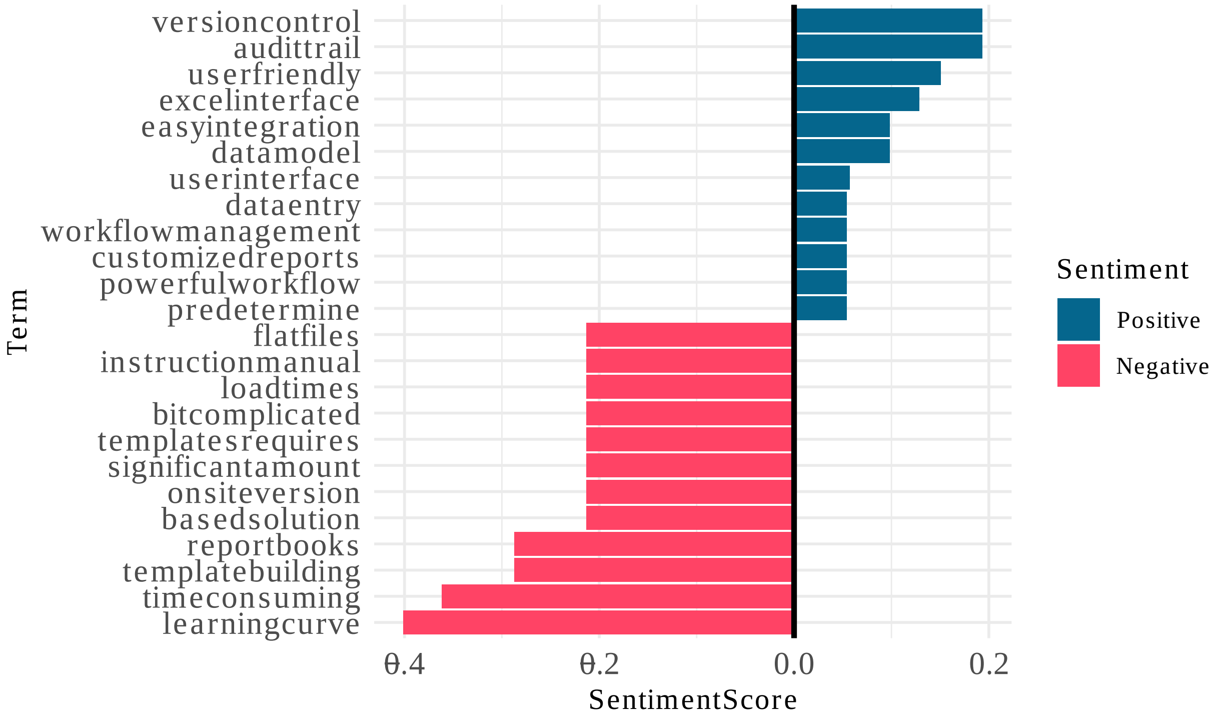
With a fitted LASSO model, we can take the top positive and negative terms and visualize their sentiment score. We have, in essence, created a “sentiment lexicon” that tells us which terms tend to be associated with positive or negative feelings about Vena Solutions. Furthermore, we have also *quantified* the degree of positive or negative sentiment.

```
sentiment_lexicon <- cv_lasso_tidy %>%
  mutate(direction = ifelse(estimate > 0, "Positive", "Negative"),
         direction = fct_relevel(direction, "Positive")) %>%
  filter(term != "(Intercept)") %>%
  select(term, estimate, direction)

sentiment_lexicon %>%
  group_by(direction) %>%
  top_n(12, wt = abs(estimate)) %>%
  ungroup() %>%
  mutate(term = fct_reorder(term, estimate)) %>%
  ggplot(aes(estimate, term, fill = direction)) +
  geom_col() +
  geom_vline(xintercept = 0, size = 1) +
  scale_fill_manual(values = c("#05668D", "#FF4365")) +
  labs(title = "Vena Solutions Sentiment Lexicon",
       subtitle = "Top 12 positive and negative terms from TrustRadius reviews",
       fill = "Sentiment",
       x = "Sentiment Score",
       y = "Term",
       caption = "Source: TrustRadius Reviews") +
  theme(text = element_text(family = "Bahnschrift"),
        axis.text = element_text(size = 12))
```

VenaSolutionsSentimentLexicon

Top12positiveandnegativetermsfromTrustRadiusreviews



Source:TrustRadiusReviews