Reading news articles on the will-they-won't-they post-Brexit trade negotiations with the EU sees days of optimism jarred by days of gloom. Do negative news articles, when one wants a positive outcome, leave a deeper impression?

I wondered if I could get a more objective view from quantitative analysis of textual data. To do this, I'm going to look at hundreds of articles published in the Guardian newspaper over the course of the year to see how trade-talk sentiment has changed week-to-week.

```
library(tidyverse)
library(rebus)
library(wesanderson)
library(kableExtra)
library(lubridate)
library(GuardianR)
library(quanteda)
library(scales)
library(tictoc)
library(patchwork)
library(text2vec)
library(topicmodels)
```

```
theme_set(theme_bw())

cols <- wes_palette(name = "Chevalier1")
```

The Withdrawal Agreement between the UK and the European Union was signed on the 24th of January 2020. I'll import Brexit-related newspaper articles from that date.

The Guardian newspaper asks for requests to span no more than 1 month at a time. So I'll first create a set of monthly date ranges.

```
dates_df <- tibble(start_date = seq(ymd("2020-01-24"), today(), by = "1
month")) %>%
  mutate(end_date = start_date + months(1) - 1)

dates_df %>%
  kable()
```

start_date	end_date
2020-01-24	2020-02-23
2020-02-24	2020-03-23
2020-03-24	2020-04-23
2020-04-24	2020-05-23
2020-05-24	2020-06-23
2020-06-24	2020-07-23
2020-07-24	2020-08-23
2020-08-24	2020-09-23

I'll import the newspaper articles in monthly chunks. Note, access to the Guardian's API requires a key which may be requested here.

```
tic()

article_df <-
  dates_df %>%
  pmap_dfr(., function(start_date, end_date) {
    Sys.sleep(1)
    get_guardian(
      "brexit",
      from.date = start_date,
      to.date = end_date,
      api.key = key
    )
  })

toc()
```

The data need a little cleaning, for example, to remove multi-topic articles, html tags and non-breaking spaces.

```
trade_df <-
  article_df %>%
  filter(!str_detect(id, "/live/"), sectionId %in% c("world",
"politics", "business")) %>%
  mutate(
    body = str_remove_all(body, "<.*?>") %>% str_to_lower(),
    body = str_remove_all(body, "[^a-z0-9 .-]"),
    body = str_remove_all(body, "nbsp")
  )
```

A corpus then gives me a collection of texts whereby each document is a newspaper article.

```
trade_corp <- trade_df %>%
  corpus(docid_field = "shortUrl", text_field = "body")
```

Although I've only imported articles mentioning Brexit since the Withdrawal Agreement was signed, some of these articles will not be related to trade negotiations with the EU. For example, there are on-going negotiations with many countries around the world. So, I'm going to use word embeddings to help narrow the focus to the specific context of the UK-EU trade deal.

The chief negotiator for the EU is Michel Barnier, so I'll quantitatively identify words in close proximity to "Barnier" in the context of these Brexit news articles.

```
window <- 5

trade_fcm <-
  trade_corp %>%
  fcm(context = "window", window = window, count = "weighted", weights
= window:1)

glove <- GlobalVectors$new(rank = 60, x_max = 10)

set.seed(42)

wv_main <- glove$fit_transform(trade_fcm, n_iter = 10)
```

```
## INFO  [10:06:33.114] epoch 1, loss 0.3817
## INFO  [10:06:34.959] epoch 2, loss 0.2510
## INFO  [10:06:36.759] epoch 3, loss 0.2225
## INFO  [10:06:38.577] epoch 4, loss 0.2021
## INFO  [10:06:40.438] epoch 5, loss 0.1847
## INFO  [10:06:42.303] epoch 6, loss 0.1710
## INFO  [10:06:44.124] epoch 7, loss 0.1605
## INFO  [10:06:45.936] epoch 8, loss 0.1524
## INFO  [10:06:47.754] epoch 9, loss 0.1457
## INFO  [10:06:49.594] epoch 10, loss 0.1403
```

```
wv_context <- glove$components
word_vectors <- wv_main + t(wv_context)

search_coord <-
  word_vectors["barnier", , drop = FALSE]

word_vectors %>%
  sim2(search_coord, method = "cosine") %>%
  as_tibble(rownames = NA) %>%
  rownames_to_column("term") %>%
  rename(similarity = 2) %>%
  arrange(desc(similarity)) %>%
  slice(1:10) %>%
  kable()
```

term	similarity
barnier	1.0000000
negotiator	0.7966461
michel	0.7587372
frost	0.7093119
eus	0.6728152
chief	0.6365480
brussels	0.5856139
negotiators	0.5598537
team	0.5488111
accused	0.5301669

Word embedding is a learned modelling technique placing words into a multi-dimensional vector space such that contextually-similar words may be found close by. Not surprisingly, the closest word contextually is “Michel”. And as he is the chief negotiator for the EU, we find “eu’s”, “chief”, and “negotiator” also in the top most contextually-similar words.

The word embeddings algorithm, through word co-occurrence, has identified the name of Michel Barnier’s UK counterpart David Frost. So filtering articles for “Barnier”, “Frost” and “UK-EU” should help narrow the focus.

```
context_df <-
  trade_df %>%
  filter(str_detect(body, "barnier|frost|uk-eu"))
```

```
context_corp <-
  context_df %>%
  corpus(docid_field = "shortUrl", text_field = "body")
```

I can then use quanteda's `kwic` function to review the key phrases in context to ensure I'm homing in on the texts I want. Short URLs are included below so I can click on any to read the actual article as presented by The Guardian.

```
set.seed(123)

context_corp %>%
  tokens(
    remove_punct = TRUE,
    remove_symbols = TRUE,
    remove_numbers = TRUE
  ) %>%
  kwic(pattern = phrase(c("trade negotiation", "trade deal", "trade
talks")),
       valuetype = "regex", window = 7) %>%
  as_tibble() %>%
  left_join(article_df, by = c("docname" = "shortUrl")) %>%
  slice_sample(n = 10) %>%
  select(docname, pre, keyword, post, headline) %>%
  kable()
```

docname	pre	keyword	post	headline
https://gu.com/p/ee3qc	ecj unless we have such a thin	trade deal	that its not worth the paper its	Brexit: Boris Johnson faces Eurotunnel test
https://gu.com/p/end82	london a separate process to the troubled	trade talks	that got under way in london on	Irish MEP in line for EU finance role vacated due to lockdown scandal
https://gu.com/p/ezjdz	said the downsides with the eu free	trade deal	the us free trade deal and our	Brexit bill hugely damaging to UK's reputation, says ex-ambassador
https://gu.com/p/d7d9t	people we have who have been negotiating	trade deals	forever she said while people criticise the	Brexit trade talks: EU to back Spain over Gibraltar claims
https://gu.com/p/eyzhq	played down the prospect of reaching a	trade deal	with the eu in time for december	No 10 blames EU and plays down prospects of Brexit trade deal
https://gu.com/p/ez2v6	it will make it harder to strike	trade deals	going forward he told channel news after	Brexit: UK negotiators 'believe brinkmanship will reboot trade talks'
https://gu.com/p/d7n4t	alignment with eu rules in any brexit	trade deal	while brussels threatened to put tariffs on	Pound falls as Boris Johnson takes tough line on EU trade deal
https://gu.com/p/dnvbj	personal rapport when communicating remotely related post-brexit	trade talks	with eu on course to fail johnson	Fears Brexit talks could collapse in June but UK still optimistic

docname	pre	keyword	post	headline
https://gu.com/p/d94j9	this situation and we work on a	trade deal	with them of course the united kingdom	Ursula von der Leyen mocks Boris Johnson's stance on EU trade deal
https://gu.com/p/ezkxc	it threatens to damage british prospects of	trade deals	with the us and eu it puts	Tuesday briefing: Rancour as law-breaking bill goes forward

Quanteda provides a sentiment dictionary which, in addition to identifying positive and negative words, also finds negative-negatives and negative-positives such as, for example, "not effective". For each week's worth of articles, I'll calculate the proportion of positive sentiments.

```
tic()

sent_df <-
  context_corp %>%
  dfm(dictionary = data_dictionary_LSD2015) %>%
  as_tibble() %>%
  left_join(context_df, by = c("doc_id" = "shortUrl")) %>%
  mutate(
    date = ceiling_date(as_date(webPublicationDate), "week"),
    pct_pos = (positive + neg_negative) / (positive + neg_negative +
negative + neg_positive)
  )

sent_df %>%
  select(doc_id, starts_with("pos"), starts_with("neg")) %>%
  slice(1:10) %>%
  kable()
```

doc_id	positive	negative	neg_positive	neg_negative
https://gu.com/p/d6qhb	40	22	0	0
https://gu.com/p/d9e9j	27	15	0	0
https://gu.com/p/d6kzd	51	27	0	1
https://gu.com/p/d6bt2	37	7	0	0
https://gu.com/p/d9vjq	13	23	0	0
https://gu.com/p/d7n8b	57	34	1	0
https://gu.com/p/d79cn	56	48	3	1
https://gu.com/p/d6t3c	28	26	0	0
https://gu.com/p/d9xtf	33	13	1	0
https://gu.com/p/d696t	15	21	1	0

```
summary_df <- sent_df %>%
  group_by(date) %>%
  summarise(pct_pos = mean(pct_pos), n = n())

toc()

## 0.708 sec elapsed
```

Plotting the changing proportion of positive sentiment over time did surprise me a little. The outcome was more balanced than I expected which perhaps confirms the deeper impression left on me by negative articles.

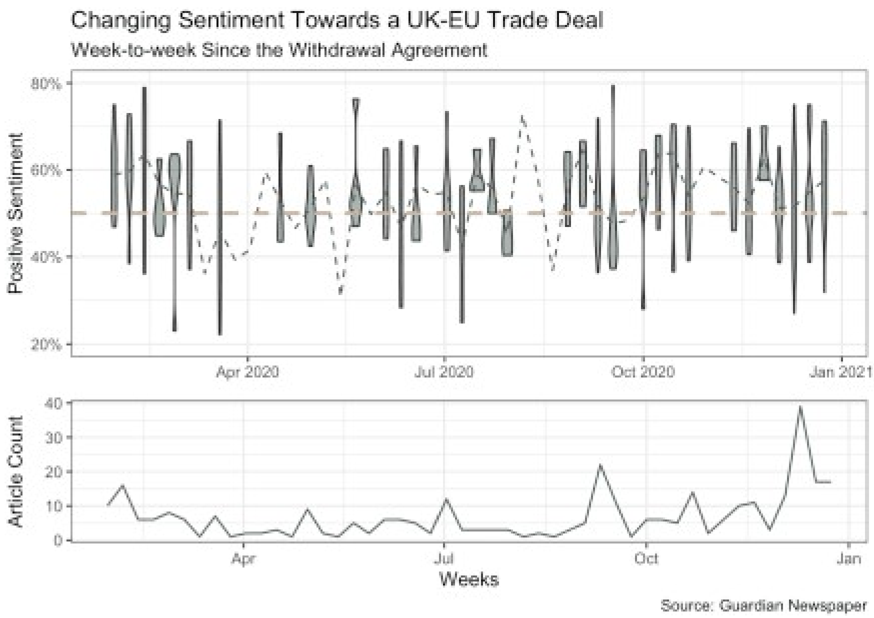
The upper violin plot shows the average weight of the sentiment across multiple articles for each week. Individually the articles range from 20% to 80% positive, with discernible periods of relatively negative and relatively positive sentiment.

The lower plot shows the volume of articles. As we draw closer to the crunch-point the volume appears to be picking up.

```
p1 <- sent_df %>%
  ggplot(aes(date, pct_pos)) +
  geom_violin(aes(group = date), alpha = 0.5, fill = cols[1]) +
  geom_line(data = summary_df, aes(date, pct_pos), colour = cols[1],
linetype = "dashed") +
  geom_hline(yintercept = 0.5, linetype = "dotted", colour = cols[4]) +
  scale_y_continuous(labels = percent_format(), limits = c(0.2, 0.8)) +
  labs(title = "Changing Sentiment Towards a UK-EU Trade Deal",
       subtitle = "Week-to-week Since the Withdrawal Agreement",
       x = NULL, y = "Positive Sentiment")

p2 <- summary_df %>%
  ggplot(aes(date, n)) +
  geom_line(colour = cols[1]) +
  labs(x = "Weeks", y = "Article Count",
       caption = "Source: Guardian Newspaper")

p1 / p2 +
  plot_layout(heights = c(2, 1))
```



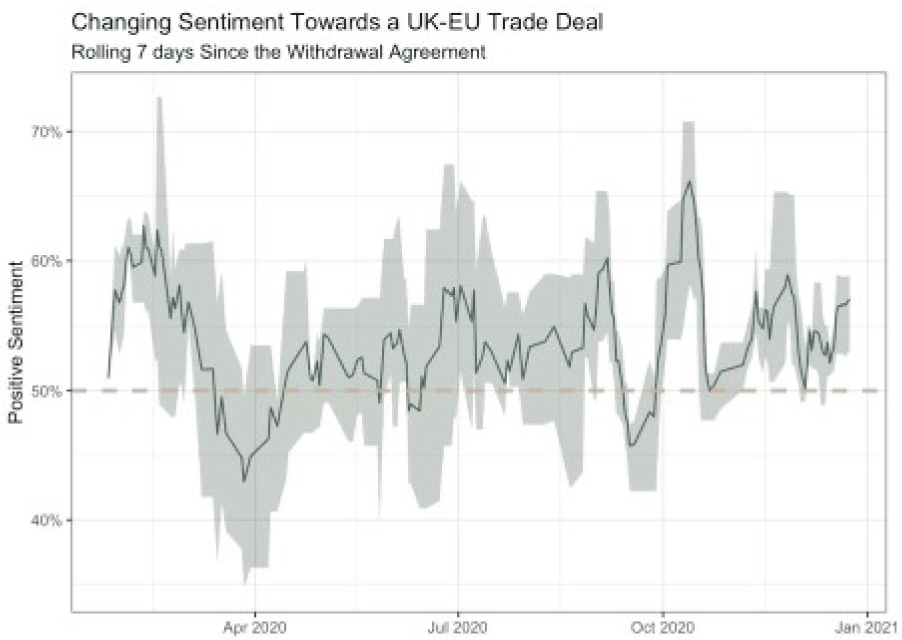
Some writers exhibit more sentiment variation than others.

```
byline_df <-
  sent_df %>%
  mutate(byline = word(byline, 1, 2) %>% str_remove_all(PUNCT)) %>%
```

```
  group_by(byline, date) %>%
  summarise(pct_pos = mean(pct_pos), n = n())

top_3 <- byline_df %>%
  count(byline, sort = TRUE) %>%
  ungroup() %>%
  filter(!is.na(byline)) %>%
  slice(c(3, 2)) %>%
  pull(byline)

byline_df %>%
  filter(byline %in% top_3) %>%
  ggplot(aes(date, pct_pos, colour = byline)) +
  geom_line() +
  geom_hline(yintercept = 0.5, linetype = "dotted", colour = cols[2]) +
  scale_y_continuous(labels = percent_format(), limits = c(0.2, 0.8)) +
  scale_colour_manual(values = cols[c(1, 4)]) +
  labs(title = "Changing Sentiment Towards a UK-EU Trade Deal",
       subtitle = "Week-to-week Since the Withdrawal Agreement",
       x = "Weeks", y = "Positive Sentiment", colour = "Byline",
       caption = "Source: Guardian Newspaper")
```



R Toolbox

Summarising below the packages and functions used in this post enables me to separately create a toolbox visualisation summarising the usage of packages and functions across all posts.

Package	Function
base	library[12]; c[8]; function[2]; mean[2]; set.seed[2]; conflicts[1]; cumsum[1]; is.na[1]; months[1]; search[1]; seq[1]; sum[1]; Sys.sleep[1]
dplyr	filter[8]; mutate[8]; as_tibble[4]; group_by[3]; if_else[3]; n[3]; select[3]; slice[3]; summarise[3]; tibble[3]; arrange[2]; desc[2]; left_join[2]; starts_with[2]; count[1]; pull[1]; rename[1]; slice_sample[1]; ungroup[1]

Package	Function
ggplot2	aes[5]; geom_line[3]; ggplot[3]; labs[3]; geom_hline[2]; scale_y_continuous[2]; geom_violin[1]; scale_colour_manual[1]; theme_bw[1]; theme_set[1]
GuardianR	get_guardian[1]
kableExtra	kable[5]
lubridate	date[3]; as_date[1]; ceiling_date[1]; today[1]; ymd[1]
patchwork	plot_layout[1]
purrr	map[1]; map2_dfr[1]; pmap_dfr[1]; possibly[1]; set_names[1]
quanteda	corpus[2]; data_dictionary_LSD2015[1]; dfm[1]; fcm[1]; kwic[1]; phrase[1]; t[1]; tokens[1]
readr	read_lines[1]
rebus	literal[4]; lookahead[3]; whole_word[2]; ALPHA[1]; lookbehind[1]; one_or_more[1]; or[1]; PUNCT[1]
scales	percent_format[2]
stringr	str_detect[5]; str_remove_all[5]; str_c[2]; str_remove[2]; str_count[1]; str_to_lower[1]; word[1]
text2vec	sim2[1]
tibble	enframe[1]; rownames_to_column[1]
tictoc	tic[2]; toc[2]
tidyr	unnest[1]
wesanderson	wes_palette[1]