09 Case study: analyzing usenet text

H. David Shea

05 Aug 2021

Contents

The data set used in the folowing analyses is a set of 20,000 messages sent to 20 Usenet bulletin boards in 1993. This data set is publicly available at http://qwone.com/~jason/20Newsgroups/.

Pre-processing

```
# processing these data files takes a long time - saved to .rds file to speed up processing
if (file.exists("data/raw_text.rda")) {
    load("data/raw_text.rda")
} else {
    training_folder <- "data/20news-bydate/20news-bydate-train/"

    # Define a function to read all files from a folder into a data frame
    read_folder <- function(infolder) {
        tibble(file = dir(infolder, full.names = TRUE)) %>%
            mutate(text = map(file, read_lines)) %>%
            transmute(id = basename(file), text) %>%
            unnest(text)
    }

    # Use unnest() and map() to apply read_folder to each subfolder
    raw_text <-
        tibble(folder = dir(training_folder, full.names = TRUE)) %>%
        mutate(folder_out = map(folder, read_folder)) %>%
        unnest(cols = c(folder_out)) %>%
        transmute(newsgroup = basename(folder), id, text)
}

raw_text %>%
    slice_head(n = 10) %>%
    kable(caption = "Example text from `20news-bydate` data set.")
```

Table 1: Example text from `20news-bydate` data set.

newsgroup	id	text
alt.atheism	49960	From: mathew mathew@mantis.co.uk
alt.atheism	49960	Subject: Alt.Atheism FAQ: Atheist Resources
alt.atheism	49960	Summary: Books, addresses, music – anything related to atheism
alt.atheism	49960	Keywords: FAQ, atheism, books, music, fiction, addresses, contacts
alt.atheism	49960	Expires: Thu, 29 Apr 1993 11:57:19 GMT
alt.atheism	49960	Distribution: world
alt.atheism	49960	Organization: Mantis Consultants, Cambridge. UK.
alt.atheism	49960	Supersedes: 19930301143317@mantis.co.uk
alt.atheism	49960	Lines: 290
alt.atheism	49960	

```
raw_text %>%
    group_by(newsgroup) %>%
    summarize(messages = n_distinct(id)) %>%
    ggplot(aes(messages, newsgroup)) +
    geom_col() +
    labs(y = NULL)
```

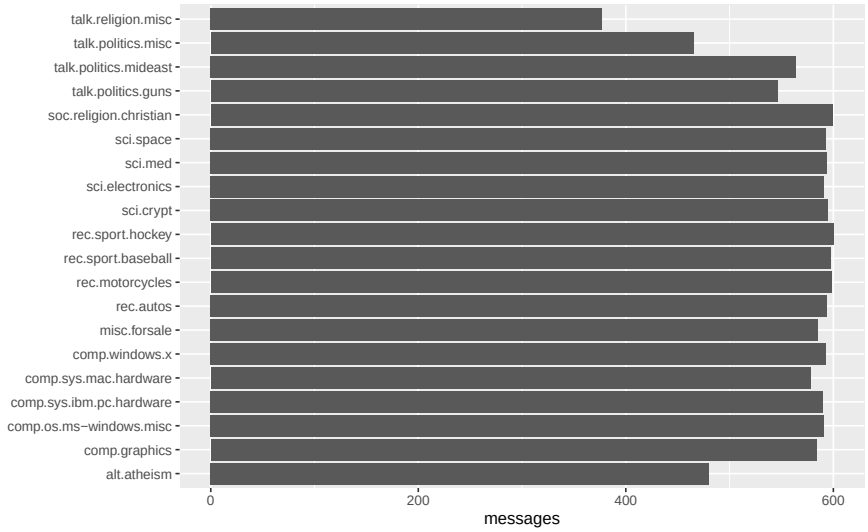

Figure 1: Number of messages from each newsgroup

Pre-processing text

Clean up the text with the following prep chunks ending in unnesting the tokens.

```
# must occur after the first occurrence of an empty line,
# and before the first occurrence of a line starting with --
cleaned_text <- raw_text %>%
    group_by(newsgroup, id) %>%
    filter(cumsum(text == "") > 0,
           cumsum(str_detect(text, "^--")) == 0) %>%
```

```
    ungroup()

# remove nested text of quotes from messages and two messages with a lot of non-text content
cleaned_text <- cleaned_text %>%
    filter(
        str_detect(text, "^[^>]+[A-Za-z\\d]") | text == "",
        !str_detect(text, "writes(:|\\.\\.\\.)$"),
        !str_detect(text, "^In article <"),
        !id %in% c(9704, 9985)
    )

usenet_words <- cleaned_text %>%
    unnest_tokens(word, text) %>%
    filter(str_detect(word, "[a-z']$"),
           !word %in% stop_words$word)
```

Words in newsgroups

```
usenet_words %>%
    count(word, sort = TRUE) %>%
    slice_max(n, n = 10, with_ties = FALSE) %>%
    kable(caption = "Highest frequency words in `20news-bydate` data set.")
```

Table 2: Highest frequency words in `20news-bydate` data set.

word	n
people	3655
time	2705
god	1626
system	1595
program	1103
bit	1097
information	1094
windows	1088
government	1084
space	1072

```
words_by_newsgroup <- usenet_words %>%
    count(newsgroup, word, sort = TRUE) %>%
    ungroup()

words_by_newsgroup %>%
    slice_max(n, n = 10, with_ties = FALSE) %>%
    kable(caption = "Highest frequency words by newsgroup in `20news-bydate` data set.")
```

Table 3: Highest frequency words by newsgroup in `20news-bydate` data set.

newsgroup	word	n
soc.religion.christian	god	917
sci.space	space	840
talk.politics.mideast	people	728
sci.crypt	key	704
comp.os.ms-windows.misc	windows	625
talk.politics.mideast	armenian	582
sci.crypt	db	549
talk.politics.mideast	turkish	514
rec.autos	car	509
talk.politics.mideast	armenians	509

Finding tf-idf within newsgroups

Newsgroups should differ in terms of topic and content. As such, the frequency of words should differ between them as well.

```
tf_idf <- words_by_newsgroup %>%
    bind_tf_idf(word, newsgroup, n) %>%
    arrange(desc(tf_idf))

tf_idf %>%
    slice_max(tf_idf, n = 10, with_ties = FALSE) %>%
    kable(caption = "Highest tf-idf values words by newsgroup in `20news-bydate` data set.")
```

Table 4: Highest tf-idf values words by newsgroup in `20news-bydate` data set.

newsgroup	word	n	tf	idf	tf_idf
comp.sys.ibm.pc.hardware	scsi	483	0.0176168	1.203973	0.0212102
talk.politics.mideast	armenian	582	0.0080489	2.302585	0.0185333
rec.motorcycles	bike	324	0.0138984	1.203973	0.0167333
talk.politics.mideast	armenians	509	0.0070393	2.302585	0.0162087
sci.crypt	encryption	410	0.0081610	1.897120	0.0154824
rec.sport.hockey	nhl	157	0.0043967	2.995732	0.0131712
talk.politics.misc	stephanopoulos	158	0.0041623	2.995732	0.0124691
rec.motorcycles	bikes	97	0.0041609	2.995732	0.0124651
rec.sport.hockey	hockey	270	0.0075611	1.609438	0.0121692
comp.windows.x	oname	136	0.0035355	2.995732	0.0105914

Looking just at the `sci.` newsgroups.

```
tf_idf %>%
    filter(str_detect(newsgroup, "^sci\\.")) %>%
    group_by(newsgroup) %>%
    slice_max(tf_idf, n = 12) %>%
    ungroup() %>%
```

```
    mutate(word = reorder(word, tf_idf)) %>%
    ggplot(aes(tf_idf, word, fill = newsgroup)) +
    geom_col(show.legend = FALSE) +
    facet_wrap( ~ newsgroup, scales = "free") +
    labs(x = "tf-idf", y = NULL)
```

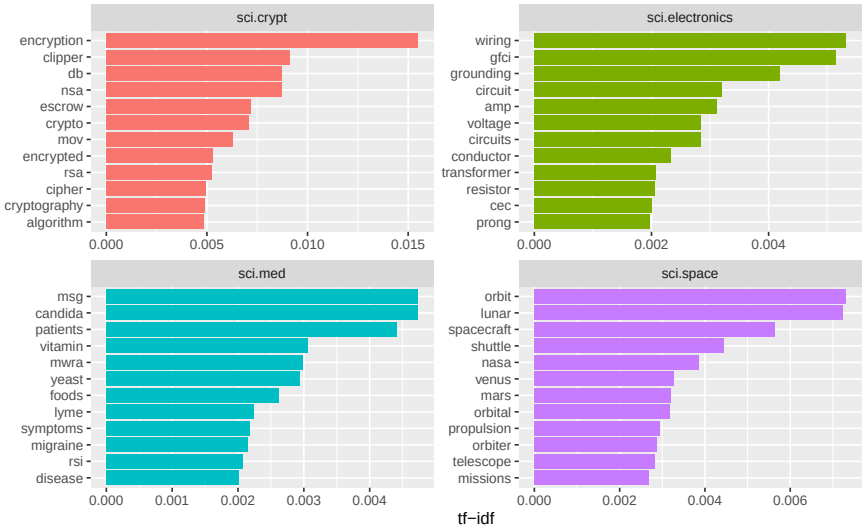

Figure 2: Terms with the highest tf-idf within each of the science-related newsgroups

```
newsgroup_cors <- words_by_newsgroup %>%
  pairwise_cor(newsgroup, word, n, sort = TRUE)
#> Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
#> Please use `tibble::as_tibble()` instead.

newsgroup_cors %>%
    slice_max(correlation, n = 10, with_ties = FALSE) %>%
    kable(caption = "Newsgroups with highest pairwise correlation of word frequencies within each newsg
```

Table 5: Newsgroups with highest pairwise correlation of word frequencies within each newsgroup

item1	item2	correlation
talk.religion.misc	soc.religion.christian	0.8347275
soc.religion.christian	talk.religion.misc	0.8347275
alt.atheism	talk.religion.misc	0.7793079
talk.religion.misc	alt.atheism	0.7793079
alt.atheism	soc.religion.christian	0.7510723
soc.religion.christian	alt.atheism	0.7510723
comp.sys.mac.hardware	comp.sys.ibm.pc.hardware	0.6799043
comp.sys.ibm.pc.hardware	comp.sys.mac.hardware	0.6799043
rec.sport.baseball	rec.sport.hockey	0.5770378
rec.sport.hockey	rec.sport.baseball	0.5770378

```
set.seed(2017)

newsgroup_cors %>%
    filter(correlation > .4) %>%
    graph_from_data_frame() %>%
    ggraph(layout = "fr") +
    geom_edge_link(aes(alpha = correlation, width = correlation)) +
    geom_node_point(size = 6, color = "lightblue") +
    geom_node_text(aes(label = name), repel = TRUE) +
    theme_void()
```

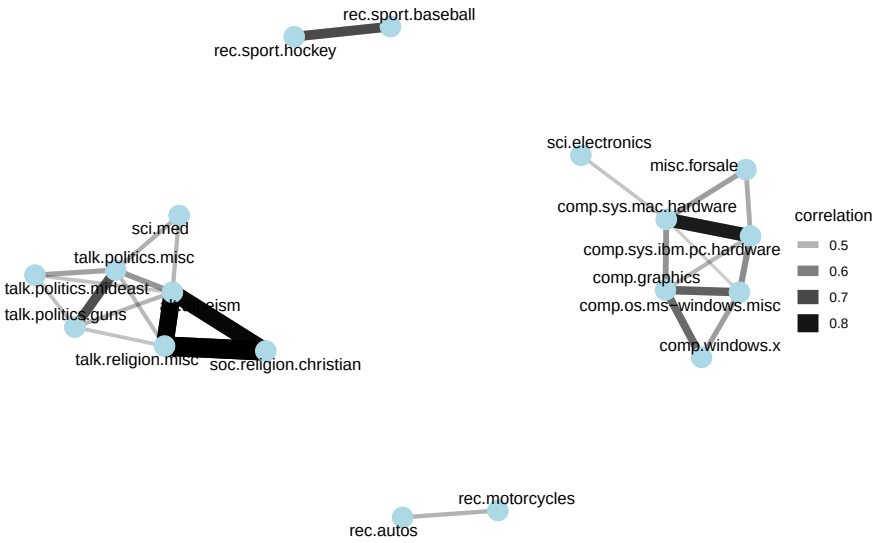

Figure 3: Network of Usenet groups based on the correlation of word counts between them (correlation > 0.4)

"It looks like there were four main clusters of newsgroups: computers/electronics, politics/religion, motor vehicles, and sports."

Topic modeling

Using the four `sci.` newsgroups, use LDA to fit a topic model.

```
# include only words that occur at least 50 times
word_sci_newsgroups <- usenet_words %>%
    filter(str_detect(newsgroup, "^sci")) %>%
    group_by(word) %>%
    mutate(word_total = n()) %>%
    ungroup() %>%
    filter(word_total > 50)

# convert into a document-term matrix
# with document names such as sci.crypt_14147
sci_dtm <- word_sci_newsgroups %>%
    unite(document, newsgroup, id) %>%
    count(document, word) %>%
```

```
    cast_dtm(document, word, n)

sci_lda <- LDA(sci_dtm, k = 4, control = list(seed = 2016))

sci_lda %>%
    tidy() %>%
    group_by(topic) %>%
    slice_max(beta, n = 8) %>%
    ungroup() %>%
    mutate(term = reorder_within(term, beta, topic)) %>%
    ggplot(aes(beta, term, fill = factor(topic))) +
    geom_col(show.legend = FALSE) +
    facet_wrap( ~ topic, scales = "free") +
    scale_y_reordered() +
    theme_light()
```

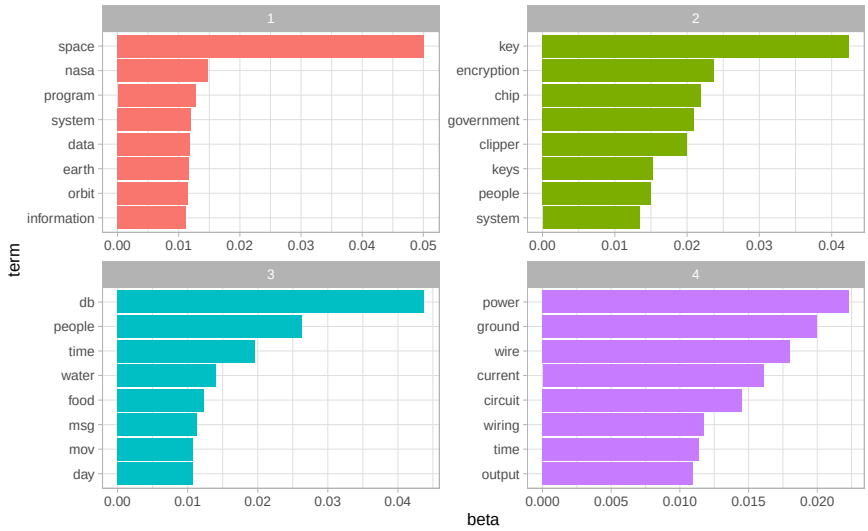

Figure 4: Top words from each topic fit by LDA on the science-related newsgroups

```
sci_lda %>%
    tidy(matrix = "gamma") %>%
    separate(document, c("newsgroup", "id"), sep = "_") %>%
    mutate(newsgroup = reorder(newsgroup, gamma * topic)) %>%
    ggplot(aes(factor(topic), gamma)) +
    geom_boxplot() +
    facet_wrap( ~ newsgroup) +
    labs(x = "Topic",
         y = "# of messages where this was the highest % topic") +
    theme_light()
```

These two graphics show that:

- Topic 1 lines up with the sci.space newsgroup
- Topic 2 lines up with the sci.crypt newsgroup
- Topic 3 lines up with the sci.med newsgroup
- Topic 4 lines up with the sci.electronics newsgroup

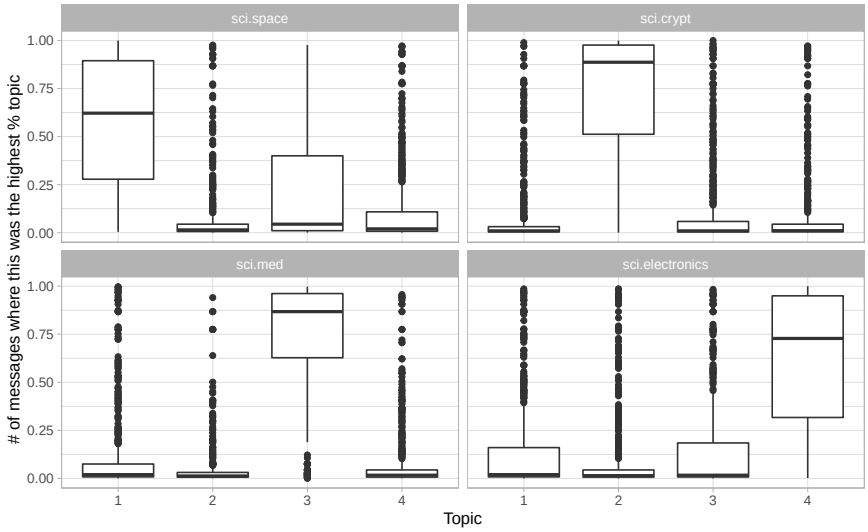

Figure 5: Distribution of gamma for each topic within each Usenet newsgroup

Sentiment analysis

Using the `AFINN` sentiment lexicon to analyze how often positive and negative words occur by newsgroup.

```
newsgroup_sentiments <- words_by_newsgroup %>%
    inner_join(get_sentiments("afinn"), by = "word") %>%
    group_by(newsgroup) %>%
    summarize(value = sum(value * n) / sum(n))

newsgroup_sentiments %>%
    mutate(newsgroup = reorder(newsgroup, value)) %>%
    ggplot(aes(value, newsgroup, fill = value > 0)) +
    geom_col(show.legend = FALSE) +
    labs(x = "Average sentiment value", y = NULL)
```

Politics - negative. People selling things - positive. Makes sense.

Sentiment analysis by word

Examining the total positive and negative contributions of each word.

```
contributions <- usenet_words %>%
    inner_join(get_sentiments("afinn"), by = "word") %>%
    group_by(word) %>%
    summarize(occurences = n(),
              contribution = sum(value))

contributions %>%
    arrange(desc(abs(contribution))) %>%
    slice_max(abs(contribution), n = 10, with_ties = FALSE) %>%
    kable(caption = "Words with the highest contribution to sentiment scoring")
```

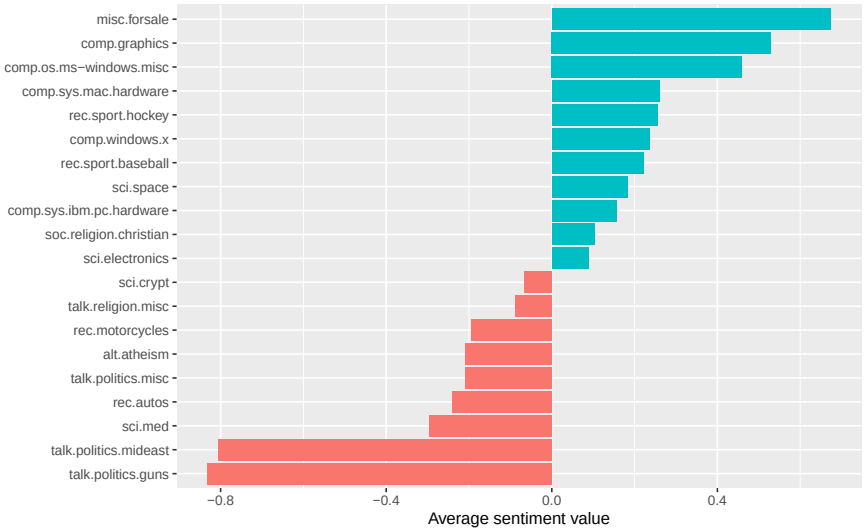

Figure 6: Average AFINN value for posts within each newsgroup

Table 6: Words with the highest contribution to sentiment scoring

word	occurences	contribution
bad	659	-1977
true	864	1728
god	1626	1626
win	395	1580
support	703	1406
hell	339	-1356
wrong	638	-1276
nice	391	1173
love	387	1161
dead	364	-1092

```
contributions %>%
    slice_max(abs(contribution), n = 25) %>%
    mutate(word = reorder(word, contribution)) %>%
    ggplot(aes(contribution, word, fill = contribution > 0)) +
    geom_col(show.legend = FALSE) +
    labs(y = NULL) +
    theme_light()
```

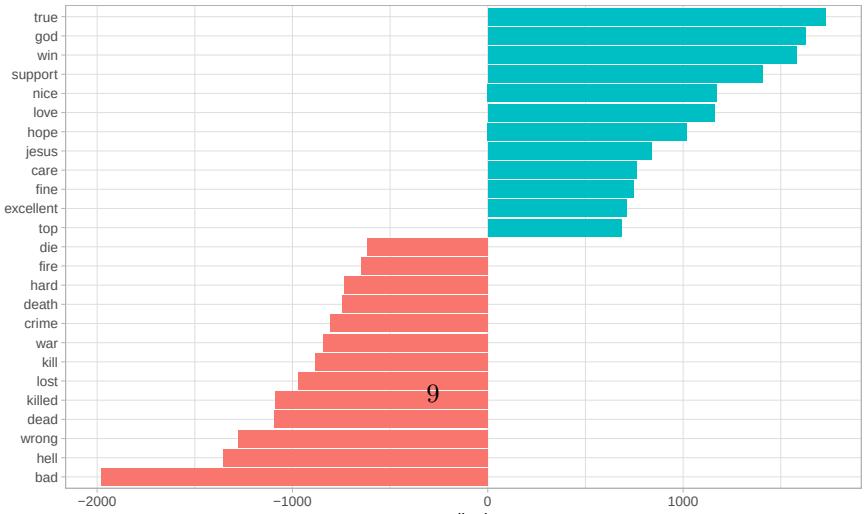

Table 7: Contribution of each word to each newsgroup's sentiment score

newsgroup	word	n	value	contribution
soc.religion.christian	god	917	1	0.0144180
soc.religion.christian	jesus	440	1	0.0069181
talk.politics.guns	gun	425	-1	-0.0066823
talk.religion.misc	god	296	1	0.0046540
alt.atheism	god	268	1	0.0042138
soc.religion.christian	faith	257	1	0.0040408
talk.religion.misc	jesus	256	1	0.0040251
talk.politics.mideast	killed	202	-3	-0.0095282
talk.politics.mideast	war	187	-2	-0.0058804
soc.religion.christian	true	179	2	0.0056288

```
top_sentiment_words %>%
    filter(str_detect(newsgroup, "^(talk|alt|misc)")) %>%
    group_by(newsgroup) %>%
    slice_max(abs(contribution), n = 12) %>%
    ungroup() %>%
    mutate(
        newsgroup = reorder(newsgroup, contribution),
        word = reorder_within(word, contribution, newsgroup)
    ) %>%
    ggplot(aes(contribution, word, fill = contribution > 0)) +
    geom_col(show.legend = FALSE) +
    scale_y_reordered() +
    facet_wrap( ~ newsgroup, scales = "free") +
    labs(x = "Sentiment value * # of occurrences", y = NULL) +
    theme_light()
```

Figure 8: Words that contributed the most to sentiment scores within each of six newsgroups

Another drawback is shown here where 'god' and 'jesus' show up as high positive contribution in alt.atheism

and talk.religion.misc but likely have different true sentiment between those groups. And similarly, 'gun' is the highest negative sentiment contributor in the talk.politics.guns newsgroup where the term is very likely used in a positive sentiment.

Sentiment analysis by message

Examining the sentiment by individual message.

```
sentiment_messages <- usenet_words %>%
    inner_join(get_sentiments("afinn"), by = "word") %>%
    group_by(newsgroup, id) %>%
    summarize(sentiment = mean(value),
              words = n()) %>%
    ungroup() %>%
    filter(words >= 5)

sentiment_messages %>%
    arrange(desc(sentiment)) %>%
    slice_min(sentiment, n = 10, with_ties = FALSE) %>%
    kable(caption = "Most positive messages")
```

Table 8: Most positive messages

newsgroup	id	sentiment	words
rec.sport.hockey	53907	-3.000000	6
sci.electronics	53899	-3.000000	5
talk.politics.mideast	75918	-3.000000	7
rec.autos	101627	-2.833333	6
comp.graphics	37948	-2.800000	5
comp.windows.x	67204	-2.700000	10
talk.politics.guns	53362	-2.666667	6
alt.atheism	51309	-2.600000	5
comp.sys.mac.hardware	51513	-2.600000	5
rec.autos	102883	-2.600000	5

```
# function to print an individual message
print_message <- function(group, message_id) {
    result <- cleaned_text %>%
        filter(newsgroup == group, id == message_id, text != "")

    cat(result$text, sep = "\n")
}
```

Winner! Winner! Chicken dinner!

```
# Most positive message
print_message("rec.sport.hockey", 53560)
#> Everybody.  Please send me your predictions for the Stanley Cup Playoffs!
#> I want to see who people think will win.!!!!!!!
#> Please Send them in this format, or something comparable:
```

```
#> 1. Winner of Buffalo-Boston
#> 2. Winner of Montreal-Quebec
#> 3. Winner of Pittsburgh-New York
#> 4. Winner of New Jersey-Washington
#> 5. Winner of Chicago-(Minnesota/St.Louis)
#> 6. Winner of Toronto-Detroit
#> 7. Winner of Vancouver-Winnipeg
#> 8. Winner of Calgary-Los Angeles
#> 9. Winner of Adams Division (1-2 above)
#> 10. Winner of Patrick Division (3-4 above)
#> 11. Winner of Norris Division (5-6 above)
#> 12. Winner of Smythe Division (7-8 above)
#> 13. Winner of Wales Conference (9-10 above)
#> 14. Winner of Campbell Conference (11-12 above)
#> 15. Winner of Stanley Cup (13-14 above)
#> I will summarize the predictions, and see who is the biggest
#> INTERNET GURU PREDICTING GUY/GAL.
#> Send entries to Richard Madison
#> rrmadiso@napier.uwaterloo.ca
#> PS:  I will send my entries to one of you folks so you know when I say
#> I won, that I won!!!!!

sentiment_messages %>%
  arrange(sentiment) %>%
    slice_max(sentiment, n = 10, with_ties = FALSE) %>%
    kable(caption = "Most negative messages")
```

Table 9: Most negative messages

newsgroup	id	sentiment	words
rec.sport.hockey	53560	3.888889	18
rec.sport.hockey	53602	3.833333	30
rec.sport.hockey	53822	3.833333	6
rec.sport.hockey	53645	3.230769	13
rec.autos	102768	3.200000	5
misc.forsale	75965	3.000000	5
misc.forsale	76037	3.000000	5
rec.sport.baseball	104458	3.000000	11
rec.sport.hockey	53571	3.000000	5
comp.os.ms-windows.misc	9620	2.857143	7

```
# Most negative message
print_message("rec.sport.hockey", 53907)
#> Losers like us? You are the fucking moron who has never heard of the Western
#> Business School, or the University of Western Ontario for that matter. Why
#> don't you pull your head out of your asshole and smell something other than
#> shit for once so you can look on a map to see where UWO is! Back to hockey,
#> the North Stars should be moved because for the past few years they have
#> just been SHIT. A real team like Toronto would never be moved!!!
#> Andrew--
```

Negative indeed.

N-gram analysis

Look at using bigrams to counter the effect of negations like 'don't like' and 'not true'.

```
usenet_bigrams <- cleaned_text %>%
    unnest_tokens(bigram, text, token = "ngrams", n = 2)

usenet_bigram_counts <- usenet_bigrams %>%
    count(newsgroup, bigram, sort = TRUE) %>%
    separate(bigram, c("word1", "word2"), sep = " ")

negate_words <- c("not", "without", "no", "can't", "don't", "won't")

usenet_bigram_counts %>%
    filter(word1 %in% negate_words) %>%
    count(word1, word2, wt = n, sort = TRUE) %>%
    inner_join(get_sentiments("afinn"), by = c(word2 = "word")) %>%
    mutate(contribution = value * n) %>%
    group_by(word1) %>%
    slice_max(abs(contribution), n = 10) %>%
    ungroup() %>%
    mutate(word2 = reorder_within(word2, contribution, word1)) %>%
    ggplot(aes(contribution, word2, fill = contribution > 0)) +
    geom_col(show.legend = FALSE) +
    facet_wrap( ~ word1, scales = "free", nrow = 3) +
    scale_y_reordered() +
    labs(x = "Sentiment value * # of occurrences",
         y = "Words preceded by a negation") +
    theme_light()
```

Don't want/like/care. **No** problem(s).

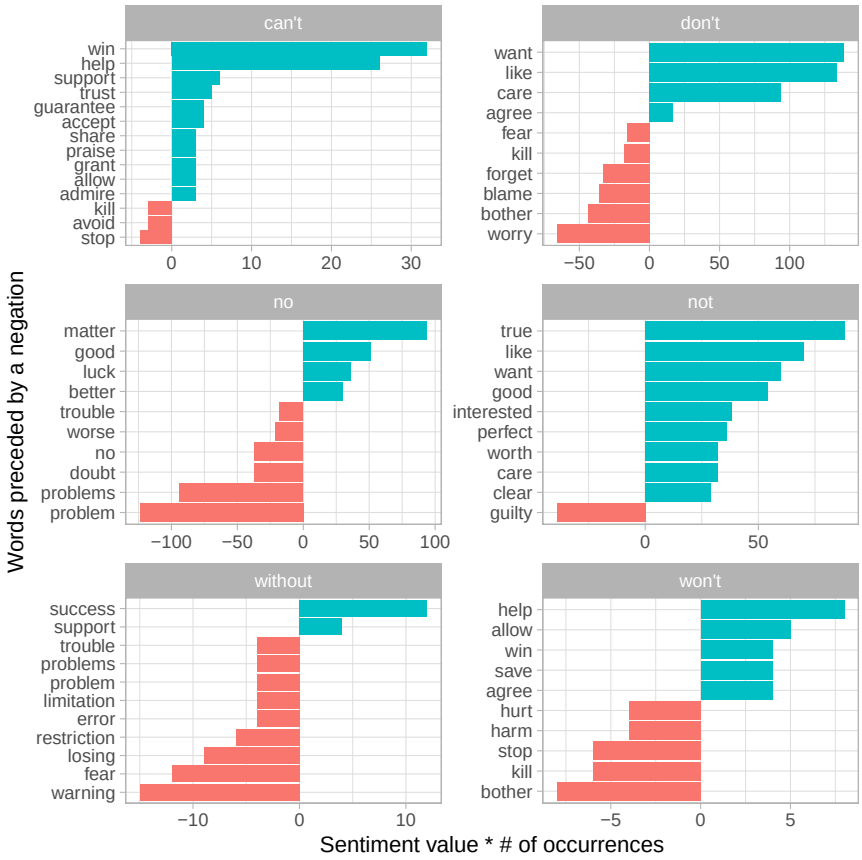

Figure 9: Words that contributed the most to sentiment when they followed a 'negating' word