Portfolio2

Negar

4/09/2022

Summary

This project is about topic modeling, which is an interesting task in NLP. In this project,first, we are trying to identify major topics in unlabeled texts, and then see which words are essential for text that is labeled for topic.

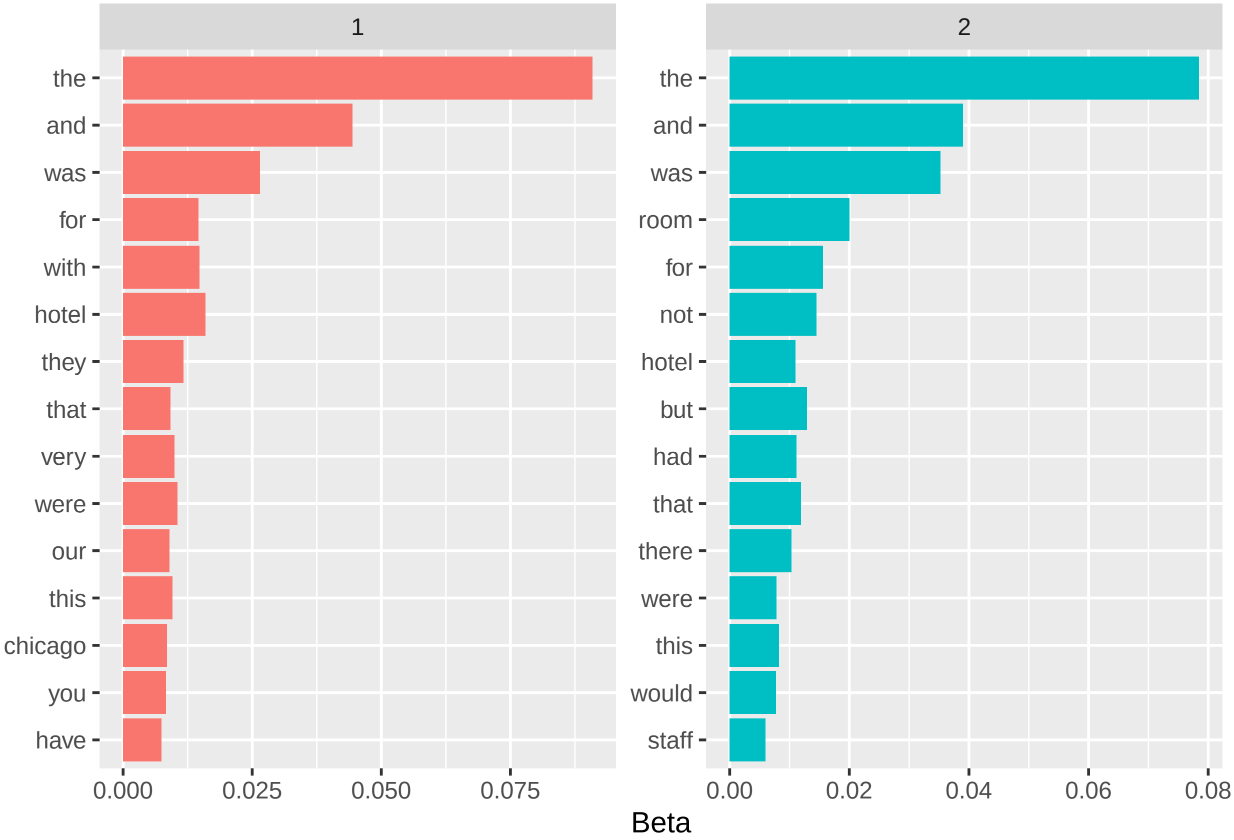
```r
data = read.csv(here::here("Data","data_raw","deceptive-opinion.csv"))
```

Function to get and plot the most informative terms by a specified number of topics, using LDA. Latent Dirichlet allocation, more commonly shortened to LDA, is one unsupervised way of doing topic modeling. Since I need to use LDA several times, the follwing function is written to do it more easily. The function takes a text column from a dataset and returns a plot of the most informative words for a given number of topics.

```r
top_terms_by_topic_LDA <- function(input_text,
                                   plot = T,
                                   number_of_topics = 4) {
    Corpus <- VCorpus(VectorSource(input_text))
    DTM <- DocumentTermMatrix(Corpus)
    unique_indexes <- unique(DTM$i)
    DTM <- DTM[unique_indexes,]

    # preform LDA & get the words/topic in a tidy text format
    lda <- LDA(DTM, k = number_of_topics, control = list(seed = 1234))
    topics <- tidy(lda, matrix = "beta")

    # get the top ten terms for each topic
    top_terms <- topics  |>
      group_by(topic) |>
      top_n(15, beta) |>
      ungroup() |>
      arrange(topic, -beta)

    if(plot == T){
        # plot the top ten terms for each topic in order
        top_terms |>
          mutate(term = reorder(term, beta)) |>
          ggplot(aes(term, beta, fill = factor(topic))) +
          geom_col(show.legend = FALSE) +
          facet_wrap(~ topic, scales = "free") +
          labs(x = NULL, y = "Beta") +
          coord_flip()
    }else{
        return(top_terms)
    }
}
```

Since we know that this dataset contains deceptive and truthful reviews, I am going to specify that I want to know about two topics. The following plot shows top fifteen terms in the hotel reviews by topic.

```
top_terms_by_topic_LDA(data$text, number_of_topics = 2)
```



As we can see, the figure is not so informative as we do not preprocess the data and remove the uninformative words. So, in the next step, we get rid off these unformative words.

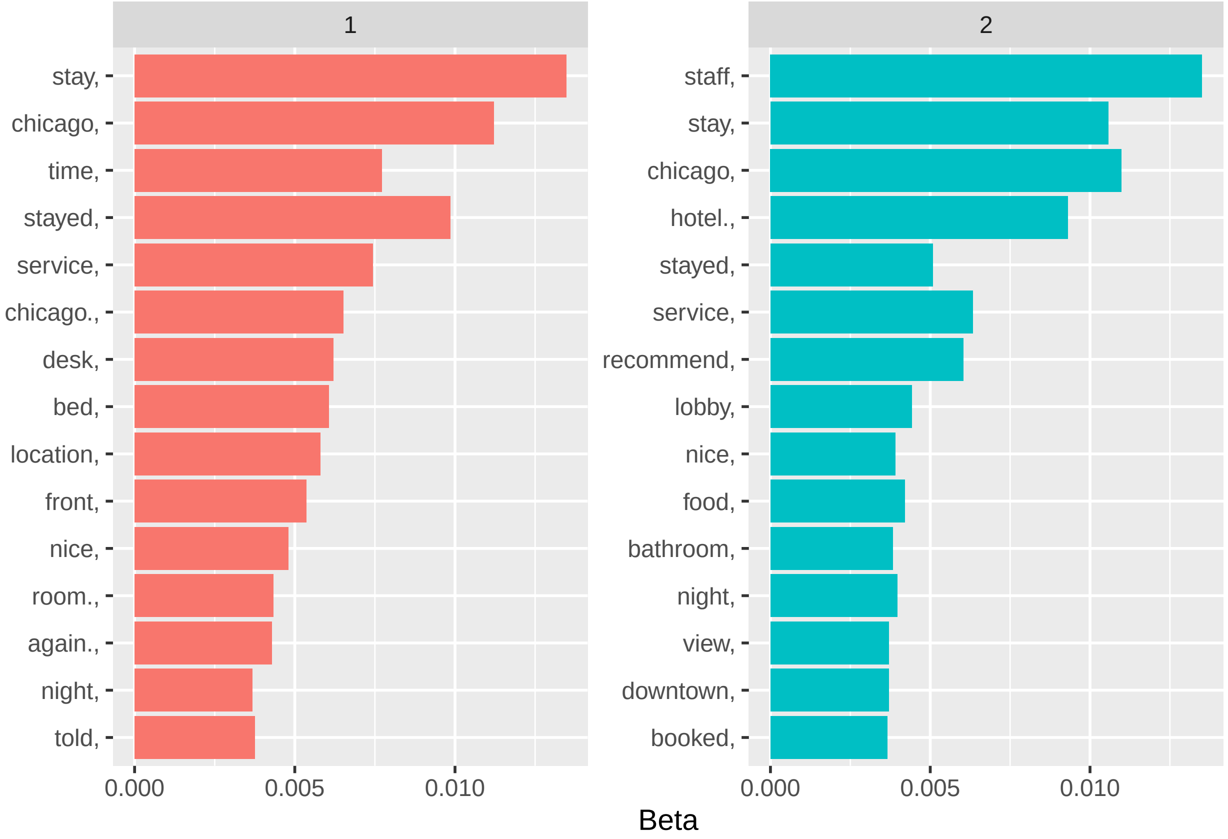
```
# create a document term matrix to clean
reviewsCorpus <- VCorpus(VectorSource(data$text))
reviewsDTM <- DocumentTermMatrix(reviewsCorpus)

# convert the document term matrix to a tidytext corpus
reviewsDTM_tidy <- tidy(reviewsDTM)

# I'm going to add my own custom stop words that I don't think will be
# very informative in hotel reviews
custom_stop_words <- tibble(word = c("hotel", "room"))

# remove stopwords
reviewsDTM_tidy_cleaned <- reviewsDTM_tidy |>
    anti_join(stop_words, by = c("term" = "word")) |>
    anti_join(custom_stop_words, by = c("term" = "word"))

# reconstruct cleaned documents (so that each word shows up the correct number of times)
cleaned_data <- reviewsDTM_tidy_cleaned |>
```

```
    group_by(document) |>
    mutate(terms = toString(rep(term, count))) |>
    select(document, terms) |>
    unique()

head(cleaned_data)
```

```
## # A tibble: 6 x 2
## # Groups:   document [6]
##   document terms
##   <chr>    <chr>
## 1 1        173, 44in, 7th, aaa, adults, bathroom(no, beat, bose, breakfast, chi~
## 2 2        $200, attractions, bed., bldg., bldg., breakfast, comfortable, dista~
## 3 3        (however,, $100, 2007, attractions., bears, bit, bother, busier, cab~
## 4 4        'standard', address, all,, avenue., cardio, center, check, chicago, ~
## 5 5        avenue, center, clean., decorated,, district., elevator, equipped, f~
## 6 6        (no, (the, 676,, 676,, absolutely, air, amazing, amenities, anything~
```

Let's plot again, and see how the top 15 words for 2 topics will change!!

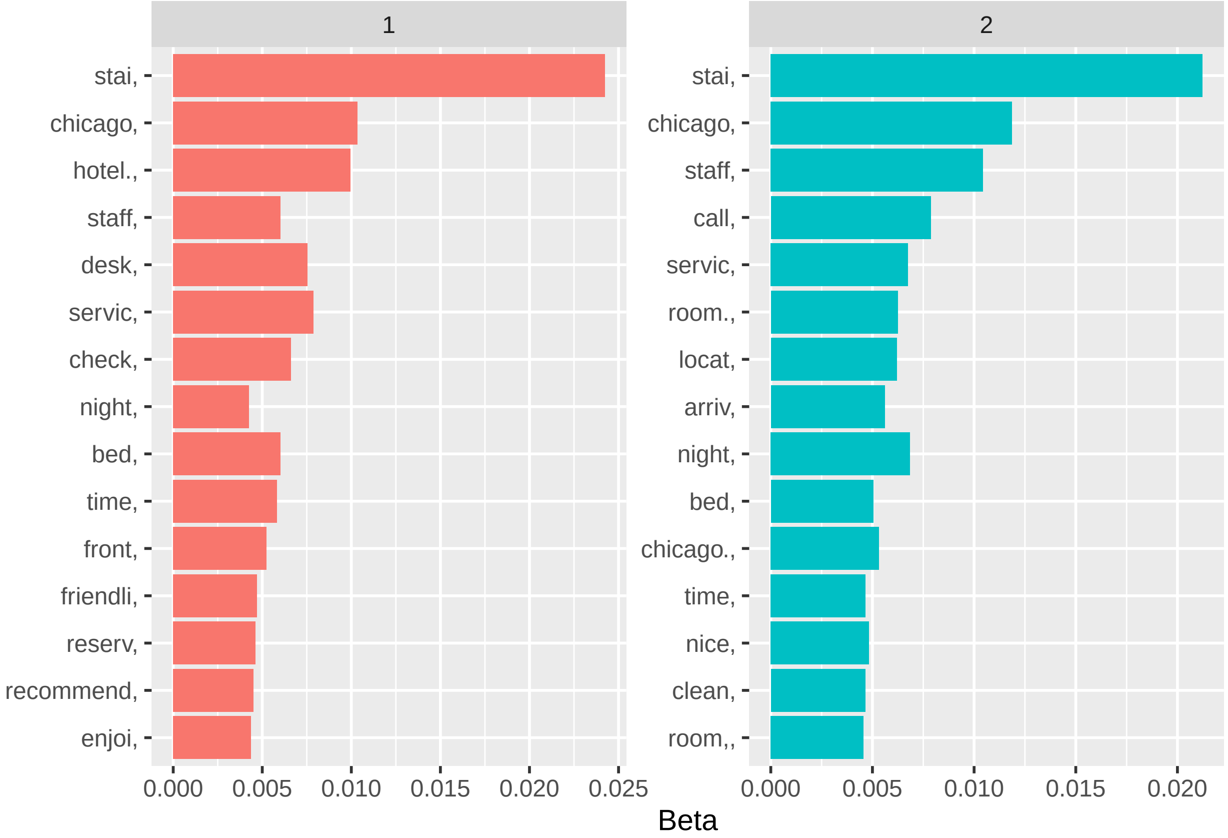
```
top_terms_by_topic_LDA(cleaned_data$terms, number_of_topics = 2)
```



This is now much better. However, as you might notice, some words like "stay" and "stayed" are similar. So, in the next step we will get rid off them by doing "stemming", which means removing all the inflection from words.

```
# stem the words
reviewsDTM_tidy_cleaned <- reviewsDTM_tidy_cleaned |>
    mutate(stem = wordStem(term))
```

```r
# reconstruct our documents
cleaned_data <- reviewsDTM_tidy_cleaned |>
    group_by(document) |>
    mutate(terms = toString(rep(stem, count))) |>
    select(document, terms) |>
    unique()

# now let's look at the new most informative terms
top_terms_by_topic_LDA(cleaned_data$terms, number_of_topics = 2)
```



In this study, it appears that stemming was not very useful in terms of producing informative subjects. We also cannot understand which (if either) of these topics are associated with deceptive reviews and which are associated with truthful ones. However, this is an intriguing issue in NLP, and I'd want to demonstrate how we can easily use topic modeling.