Lab 4: Introduction to Text Analytics

Social Media and Web Analytics @ TiSEM

Last updated: 16 May, 2021

Motivation

The last two weeks of Social Media and Web Analytics focused on regression and the notion of causal inference to understand and draw conclusions from data. Now, we'll turn to another important aspect of social media and web data, *text*.

Over the last two decades there has been a shift from using text as something we "read" to digest and summarise to treating it as an input to a more automated processes for analysis. This new methodology treats the text as data to be processed and analysed, and maybe we don't even "read" the raw text itself. Given the sheer amount of text that is now available on the web and across social media - these methods which treat "text as data" have become widespread and an essential part of a marketing analyst's toolkit. This tutorial aims to work through some introductory tools & methods available to us.

Learning Goals

By the end of this tutorial you will be able to:

1. Transform text data to obey "tidy" data principles.
2. Summarize the length a set of texts.
3. Clean up text data by removing stop words, excess numbers, and lemmatize words.
4. Visualize differences in text across different pre-classified classes of text.
5. Use the insights from text analytics to answer questions that are managerially relevant and/or have marketing implications.

Instructions to Students

These tutorials are **not graded**, but we encourage you to invest time and effort into working through them from start to finish. Add your solutions to the `lab-04_answer.Rmd` file as you work through the exercises so that you have a record of the work you have done.

Obtain a copy of both the question and answer files using Git. To clone a copy of this repository to your own PC, use the following command:

```
$ git clone https://github.com/tisem-digital-marketing/smwa-lab-04.git
```

Once you have your copy, open the answer document in RStudio as an RStudio project and work through the questions.

The goal of the tutorials is to explore how to "do" the technical side of social media analytics. Use this as an opportunity to push your limits and develop new skills. When you are uncertain or do not know what to do next - ask questions of your peers and the instructors on the class Slack channel `#lab-04-discussion`.

Exercise 1: Text Analytics with Fake Reviews

This exercise works with data on hotel reviews. The reviews are a collection of truthful and deceptive (i.e. fake) reviews of 20 hotels in the Chicago area, known in the computational linguistics community as the Deceptive Opinion Spam dataset.[1] Deceptive reviews are reviews that have been written by someone who has not stayed at the hotel they are reviewing. The data contains 1600 reviews:

- 400 truthful, positive reviews from TripAdvisor
- 400 deceptive positive reviews from Mechanical Turk
- 400 truthful, negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor, and Yelp
- 400 deceptive negative reviews from Mechanical Turk

To gain access to the data, run the following code to download it and save it in the file `data/reviews.csv`:

```r
library(googledrive)

data_id <- "1WJTEJ3zjDh8Bi7v1JqGIc62sN9LKPVAw"
out_file <- "data/reviews.csv"

drive_download(
  as_id(data_id),
  path = out_file,
  overwrite = TRUE)
```

You might need to use the following `R` libraries throughout this exercise:[2]

```r
library(readr)
library(dplyr)
library(tibble)
library(tidyr)
library(stringr)
library(tidytext)
library(ggplot2)
library(magrittr)
library(textstem)
library(reshape2)
library(wordcloud)
```

1. Assuming online reviews are all truthful, explain whether they should benefit or harm consumers and the hotels themselves.

2. If fake reviews are mixed in with truthful reviews on a website (for example, TripAdvisor) identify two reasons why their presence may harm consumers, hotels and the review website. Explain the mechanisms through which the harm is done.

3. Load the data located in `data/reviews` into a data frame called `hotel_reviews`.

solution

```r
hotel_reviews <-
  read_csv('data/reviews.csv')
```

4. In what follows we will need each hotel review (i.e. each row of the data) to have a unique identifier.

[1]The data originally was published in the paper "Finding Deceptive Opinion Spam by Any Stretch of the Imagination" by M. Ott, Y. Choi, C. Cardie, and J.T. Hancock in 2011 in the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies

[2]If you haven't installed one or more of these packages, do so by entering `install.packages("PKG_NAME")` into the R console and pressing ENTER.

Create a new column called `ID` in the dataset that has a unique identifier for each review.

HINT: the function `row_to_column()` function will help you do this.

solution

```
hotel_reviews <-
    hotel_reviews %>%
    rowid_to_column("ID")
```

5. Convert the dataset to conform with tidy data principles:

- Each variable is a column
- Each observation is a row
- Every cell is a single value

Name the resulting dataset `tidy_reviews`.

solution

```
tidy_reviews <-
    hotel_reviews %>%
    unnest_tokens(word, text)
```

6. Create a dataset called `review_length` that has two columns, a column that identifies the review and a column that counts the number of words in the review. Merge the the `review_length` data into the `hotels_df` dataset.

solution

```
review_length <-
    tidy_reviews %>%
    count(ID)

hotel_reviews <-
    hotel_reviews %>%
    inner_join(review_length, by = c("ID"))
```

7. Are deceptive reviews longer than truthful reviews? If you presented this result to a marketing manager that has little analytics training what do you expect they would say about the possibility of detecting deceptive reviews? Would you agree or disagree?

solution

```
hotel_reviews %>%
  group_by(deceptive) %>%
  summarize(mean_length = mean(n),
            sd_length   = sd(n)
            )
```

```
## # A tibble: 2 x 3
##   deceptive mean_length sd_length
##   <chr>           <dbl>     <dbl>
## 1 deceptive        147.      85.1
## 2 truthful         151.      90.3
```

8. What are stop words? Explain why you would want to remove them from the review text.
9. Remove stop words from each review using a default stop word list. Name the dataset with no stop words `tidy_reviews_no_stop`. What are the 20 most common words in the dataset?

solution

```r
tidy_reviews_no_stop <-
  tidy_reviews %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```r
tidy_reviews_no_stop %>%
  count(word, sort = TRUE) %>%
  head(20)
```

```
## # A tibble: 20 x 2
##    word            n
##    <chr>       <int>
##  1 hotel        3292
##  2 chicago      1504
##  3 stay         1303
##  4 staff         836
##  5 service       817
##  6 stayed        665
##  7 night         563
##  8 time          523
##  9 desk          509
## 10 nice          490
## 11 location      483
## 12 bed           457
## 13 clean         418
## 14 front         392
## 15 check         371
## 16 day           349
## 17 bathroom      342
## 18 experience    338
## 19 friendly      321
## 20 comfortable   310
```

10. Create a custom stop word list that contains words you might want to remove in addition to the default stop word list. Remove these words from `tidy_reviews_no_stop`.

solution

```
my_stop_words <- tibble(
  word = c(
    "hotel", "stay", "stayed"
  ),
  lexicon = "hotels"
)

all_stop_words <- stop_words %>%
  bind_rows(my_stop_words)

tidy_reviews_no_stop <- tidy_reviews %>%
  anti_join(stop_words, by = "word")
```

11. Also remove any numbers from `tidy_reviews_no_stop`.

The following code can be used to get started:

```
# First, find all numbers:
nums <- YOUR_CODE %>%
  # find all numbers in the data
  filter(str_detect(word, "^[0-9]")) %>%
  YOUR_CODE

# now remove those:
tidy_reviews_no_stop <- YOUR_CODE
```

solution

```
nums <- tidy_reviews_no_stop %>%
          filter(str_detect(word, "^[0-9]")) %>%
          select(word) %>%
          unique()

tidy_reviews_no_stop <- tidy_reviews_no_stop %>%
                        anti_join(nums, by = "word")
```

The next step we want to perform is to lemmatize each word. The goal of lemmatizing is to reduce inflectional forms and some common derivationally related forms of a word to a base word. For example:

- am, are, is ⇒ be
- car, cars, car's, cars' ⇒ car

To lemmatize each word in the reviews we will use the `lemmatize_words()` from the `textstem package`:[3]

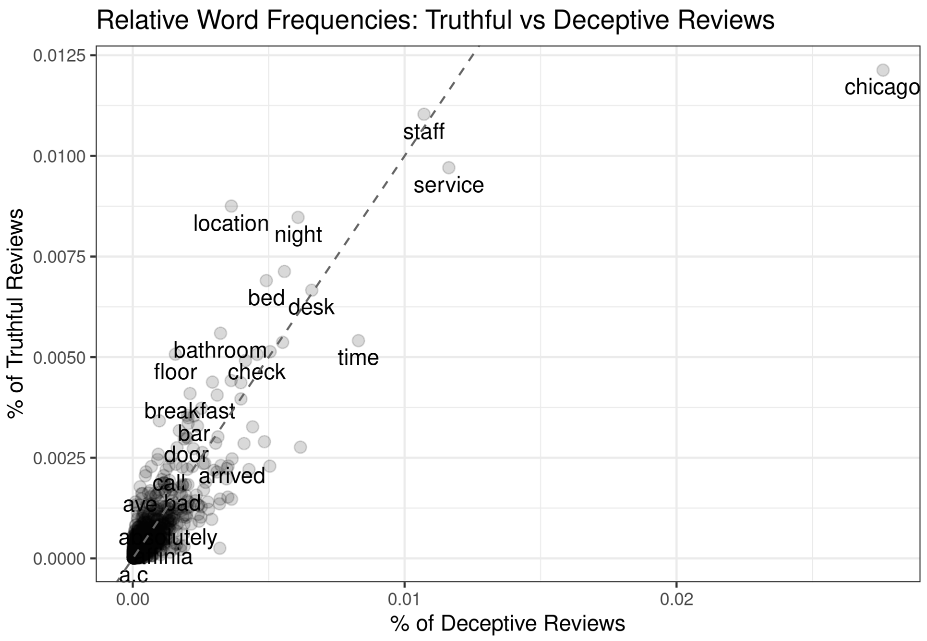
solution

```
tidy_reviews_lemma <-
    tidy_reviews_no_stop %>%
    mutate(word_lemma = lemmatize_words(word)) %>%
  # redo the stop word removal based on lemmatizing
  # so that things like what was 'hotels' now get removed
    anti_join(all_stop_words)
```

[3]An alternative would be to "stem" each word, which essentially chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

```
## Joining, by = "word"
```

12. Create a plot that graphs the frequency a word is used in deceptive reviews on the x-axis against the frequency the same word is used on the y-axis. To help you in coding up this plot, your final result should look like the following:[4]



Perform the following steps:

(a) Create a dataset called `wrd_frequency` that has three columns: a column `word` that contains all the unique words, a column `deceptive` that has the percentage of deceptive reviews that each word is included in and a column `truthful` that has the percentage of truthful reviews that each word is included in.
(b) Create the plot described above. Include a 45 degree line for reference.

To assist you in creating the plot, use the following starter code:

```r
# part (a)
wrd_frequency <- tidy_reviews_lemma %>%
  count(YOUR_CODE) %>%
  group_by(YOUR_CODE) %>%
  mutate(YOUR_CODE) %>%
  spread(YOUR_CODE)

# part (b)
wrd_frequency %>%
  ggplot(aes(x = YOUR_CODE,
             y = YOUR_CODE,
             label = YOUR_CODE
           )) +
  geom_YOURCODE(alpha = 0.15,
                size = 2.5
              ) +
  geom_text(aes(label = YOUR_CODE),
```

[4]Depending on your custom stopword list in (10) you might see slight differences between your figure and this one.

```
          check_overlap = TRUE,
          vjust = 1.5
          ) +
  geom_abline(YOUR_CODE,
            lty = 2,
            color = "grey40"
            ) +
  YOUR_CODE
```

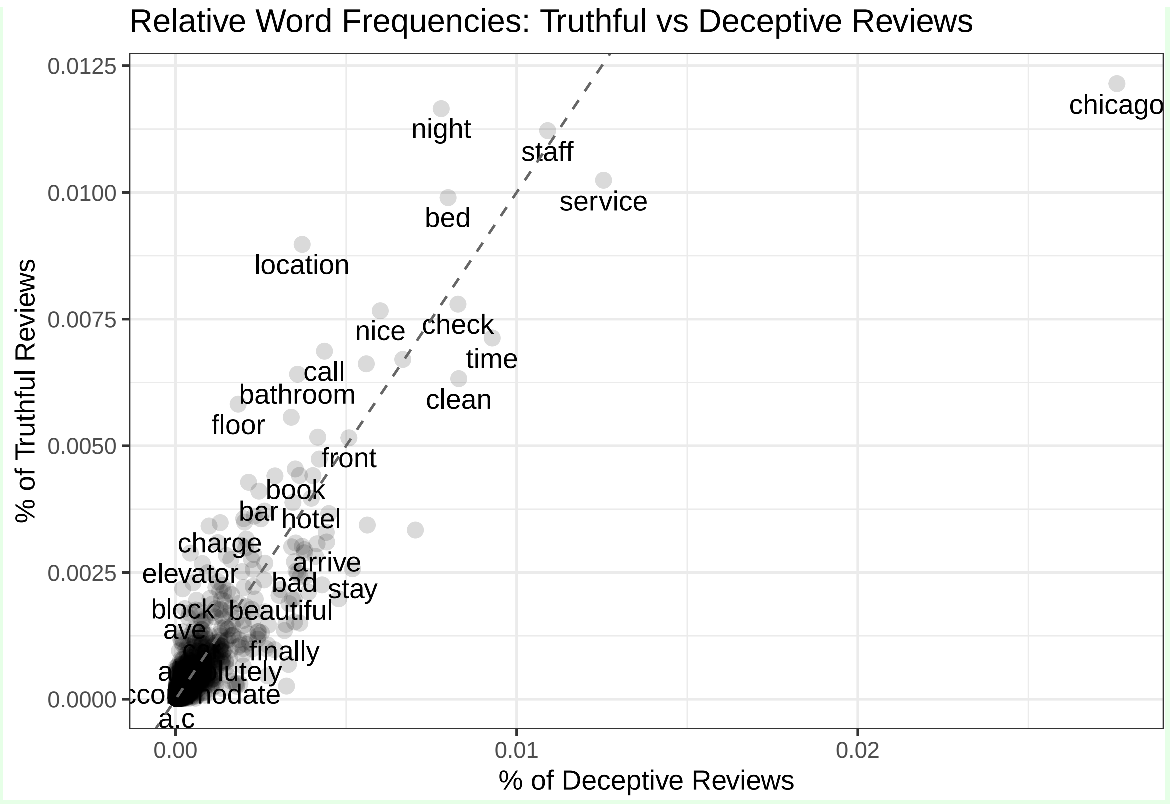
solution

```
frequency <- tidy_reviews_lemma %>%
  count(deceptive, word_lemma) %>%
  group_by(deceptive) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  tidyr::spread(deceptive, proportion)

ggplot(frequency,
       aes(x = deceptive,
           y = truthful,
           label = word_lemma
           )
       ) +
  geom_jitter(alpha = 0.15,
              size = 2.5
              ) +
  geom_text(aes(label = word_lemma),
            check_overlap = TRUE,
            vjust = 1.5
            ) +
  geom_abline(intercept = 0,
              slope = 1,
              lty = 2,
              color = "grey40"
              ) +
  theme_bw() +
  ggtitle("Relative Word Frequencies: Truthful vs Deceptive Reviews") +
  xlab("% of Deceptive Reviews") +
  ylab("% of Truthful Reviews")
```

```
## Warning: Removed 4133 rows containing missing values (geom_point).
```

```
## Warning: Removed 4133 rows containing missing values (geom_text).
```



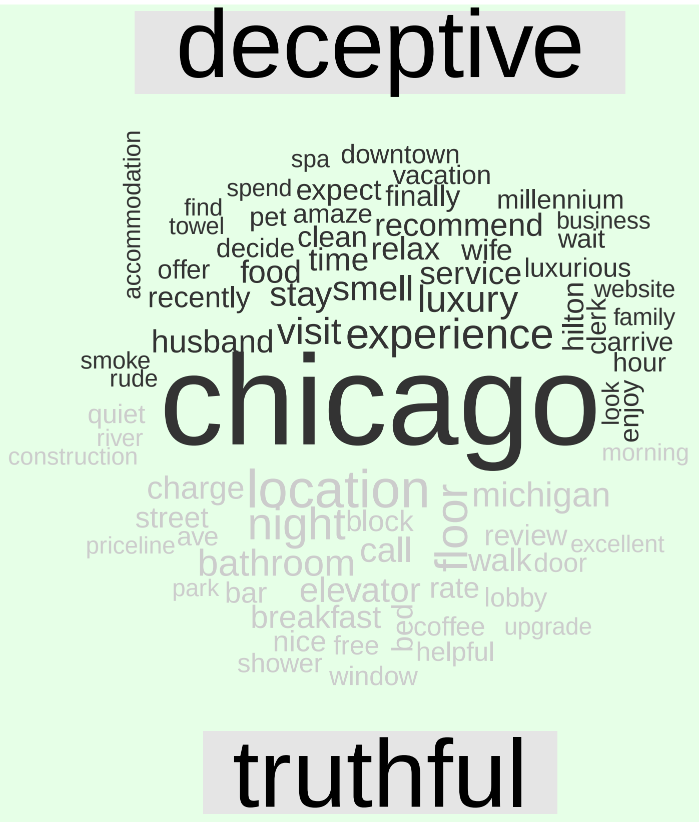
An alternative way to visualize the differences and commonalities in word use between truthful and deceptive reviews is to construct word clouds. To assist you in creating the plots, use the following starter code for each of the next two questions:

```
tidy_reviews_lemma %>%
  count(YOUR_CODE,
        YOUR_CODE,
        sort = TRUE
        ) %>%
  acast(YOUR_CODE ~ YOUR_CODE,
        value.var = "n",
        fill = 0
        ) %>%
  YOUR_CODE
```

13. Create a word cloud that visualizes the differences in word use between deceptive and truthful reviews. Use a maximum of 75 words in the word cloud.

solution

```
tidy_reviews_lemma %>%
  count(word_lemma, deceptive, sort = TRUE) %>%
  acast(word_lemma ~ deceptive, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"),
                   max.words = 75)
```

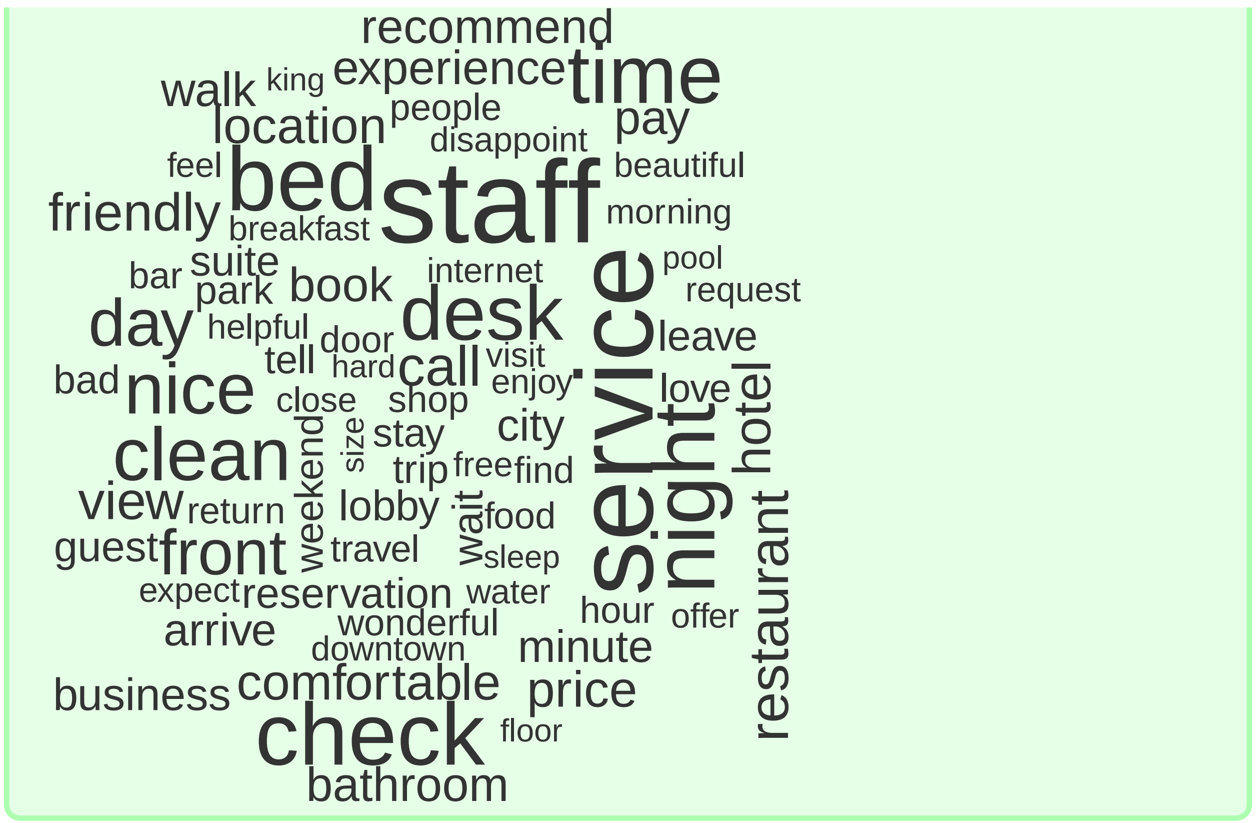


14. Create a word cloud that visualizes the commonalities in word use between deceptive and truthful reviews. Use a maximum of 75 words in the word cloud.

solution

```
tidy_reviews_lemma %>%
  count(word_lemma, deceptive, sort = TRUE) %>%
  acast(word_lemma ~ deceptive, value.var = "n", fill = 0) %>%
  commonality.cloud(colors = c("gray20"),
                    max.words = 75)
```

```
## Warning in wordcloud(rownames(term.matrix)[freq > 0], freq[freq > 0], min.freq =
## 0, : chicago could not be fit on page. It will not be plotted.
```



15. Provide a brief summary of what you learn from the three plots. (Max. 10 sentences)

16. In their 2014 AER article, Mayzlin, Dover and Chevalier argue that fake/deceptive reviews are difficult to detect.[5] Based on your analysis, do you agree with their presumption? Explain the marketing implications of your conclusions. (max. 10 sentences)

17. (Harder!) Suppose you were working in an analytics team that wanted to build a predictive model to detect fake reviews on a platform such as TripAdvisor.

(a) Can you sketch out a method/model to predict whether a review is fake? Your method should be scaleable so that it can handle thousands of new reviews per day. You can use any combination of words, figures, simple equations or pseudocode to explain your thoughts.

(b) How could you assess how successful the model is at detecting fake reviews?

(c) Suppose the model was adopted by the platform. If you detected a newly posted review was fake, would you want to remove it from the platform? Or would you leave the review online with a notification that this review might be fake? Explain.

[5] Mayzlin, Dina, Yaniv Dover, and Judith Chevalier. 2014. "Promotional Reviews: An Empirical Investigation of Online Review Manipulation." American Economic Review, 104 (8): 2421-55.