Homework 9

November 14, 2019

Group Members

Melinda Ding (mjd4), Henry Creamer (hmc5), Caleigh Page (cap9), Aaryan Jadhav (aj52), Cole Morgan (cmm16)

Project Update

For homework 9 of our project, we decided to focus on the airport reviews and sentiment analysis. We got the top 2,000 words in the reviews and scraped synonyms of each word. We then calculated the frequency of the word and its synonyms. Using these counts, we can then simplify the reviews we have which will make sentiment analysis easier.

```
library(RSQLite)
library(stringr)
library(XML)

dcon <- dbConnect(SQLite(), dbname = "group10.db")
dbListTables(dcon)
```

```
## [1] "flights" "reviews" "top2kdf" "zillow"
```

Query all of the flight review data into a data frame.

```
res <- dbSendQuery(conn = dcon, "
SELECT *
FROM reviews;")
reviews <- dbFetch(res, -1)
dbClearResult(res)
```

Gather all of words from all of the reviews. We eliminate some words such as "the" because these words provide no business value.

```
# Parse the review into a list of words
content = reviews$content
content_split = str_split(reviews$content, " ")
content_split_lower = sapply(content_split, tolower)
content_split_grep = sapply(content_split_lower, gsub, pattern="[[:punct:]]", replacement="")
words = unlist(content_split_grep, recursive = FALSE)

# Ignore non-meaningful words and words that do not have synonyms
words_to_ignore <- c("the", "this", "to", "a", "of", "in", "he", "she", "an", "and", "is", "with", "but
actual_words <- words[!words %in% words_to_ignore]
allContent <- as.vector(actual_words)

# Count the number of words
counts_of_words = table(allContent)

# Sort the top words
top_words = sort(counts_of_words, decreasing = TRUE)[1:2000]
word_vector = rep(names(top_words)[1:5])
```

Next we scrape thesaurus.com for each top 2,000 word. For example, for the word "airport", the URL we are

scraping is “https://www.thesaurus.com/browse/airport?s=t”. We scraped once and saved the output of the scraping into the top2kdf table in our database.

```
top2k = word_vector
top2kdf = data.frame(row.names=top2k)
# Iterate through the top 2,000 words and scrape the thesaurus website for each word.
for (word in top2k)
{
  # Create the url
  url = paste("https://www.thesaurus.com/browse/", word, "?s=t", sep="")
  # Try to scrape the url, if it exists
  result <- try(download.file(url, destfile = "advfn.html", quiet = TRUE))
  # Parse the information from the scrapped url
  if (result == 0) {
    doc <- htmlParse("advfn.html")
    tmp <- getNodeSet(doc, "//a[@data-linkid='nn1ov4']")
    new_line <- append(word, as.character(xmlToDataFrame(tmp)$text[1:10]))
    top2kdf <- rbind(top2kdf, data.frame(matrix(new_line, nrow=1)))
  }
}
top2kdf
```

Below is a snippet of the created SQL table. Here we query all of the scraped top2kdf table into a data frame. Column 1 is the word from the review (one of the top 20,000), and columns 2-11 are the 10 most closely related columns.

```
res <- dbSendQuery(conn = dcon, "
SELECT *
FROM top2kdf;")
top2kdf <- dbFetch(res, -1)
dbClearResult(res)
head(top2kdf)
```

```
##         X1       X2          X3           X4        X5        X6
## 1  airport airfield    airstrip installation    runway  airdrome
## 2      not  no more  not at all   not either   neither   no more
## 3 security     bond        care      freedom guarantee insurance
## 4 terminal    fatal   incurable       lethal   closing   extreme
## 5     very   actual appropriate    authentic      bare bona fide
## 6       no      nay         nix        never       not  negation
##             X7           X8         X9          X10      X11
## 1       hangar     heliport      strip     aerodome  helipad
## 2          not   not at all not either           no      nay
## 3 preservation surveillance      aegis    agreement armament
## 4      killing          lag       last       latest   latter
## 5      correct     especial    express      genuine    ideal
## 6   antithesis      antonym      blank cancellation contrary
```

The number of rows of data that have been produced is:

```
print(dim(top2kdf)[1])
```

```
## [1] 1749
```

The plot shows that there are multiple occurrences of a word, whether synonym or not. For example, the word “move” occurs 21 times, so we can expect that ~200 words in the top2kdf are synonyms of “move”. We can then condense all the other synonyms of “move” to “move” itself, creating a more simplified and condensed

dictionary of review words. This will help standardize the way we compare reviews, and make sentiment analysis more reliable from a simplified dictionary. We will then use sentiment analysis to determine whether an airport is liked or not, which will help us compare airports.

```
all_words = as.vector(t(top2kdf))
word_table = table(all_words)
word_table = sort(word_table, decreasing=TRUE)
word_df = data.frame(word_table[2:26])
```

We decided to plot the frequency of the top 25 words, to help visualize the relative frequency of a word compared to others.

```
library(ggplot2)

ggplot(data = word_df, aes(x=all_words, y=Freq)) +
  geom_bar(stat="identity", fill="orange") +
  labs(x="Word", y="Count", title="Top 25 word counts") +
  theme(axis.text.x=element_text(angle=90, hjust=1, size=12))
```