Lab #3: Dictionaries and Sentiment Analysis

Due in Sakai February 15, 2019 (by 10:00a)

Introduction

This lab will give you hands-on practice with applying dictionaries to a corpus and conducting some dictionary-based sentiment analysis.

Getting Started

First, set your working directory. Open up a fresh RStudio session (if you open RStudio and a previous work session is loaded, be sure you save any R and/or RData files before you close them out). Then set your working directory:

```
setwd("working_directory_here")
```

Then open up a fresh R script. Save it using your first initial, full last name, and then "_lab3." So my script would be titled **MTaylor_lab3.R**.

You should always strive to keep your scripts tidy. At the top of your R script, type this (substituting in your first initial and last name):

```
##############################
##  MTaylor_lab3.R
##  Note: Code for lab assignment #3
##  Author: Marshall A. Taylor
##############################

###BEGIN###
```

You are ready to begin your code. Be sure to include all the code necessary for me to check your work. Document your code thoroughly (using "#").

Remember that saving your R work is a two-step process. You save your R script using Cmd+Enter (Mac) or Cntrl+Enter (Windows). You save your R data objects like this:

```
save.image("MTaylor_lab3.RData")
```

Lastly, prep a Word, Pages, or LaTeX document that has the same title structure: e.g., **MTaylor_lab3.docx**. This is where you put your write-ups and visualizations.

You should turn in three documents to Sakai: your **R script** that shows the code you used, **RData file** that provides the data your scraped, and **Word document** (or whatever text processor you choose to use) with your write-ups.

Assignment

There are three components to this lab. First, you will decide on the corpus you want to use. Second, you will create your own custom dictionary and apply it to the corpus. Finally, you will conduct some basic sentiment analyses.

Section #1: Select a Corpus

First, select a corpus from the `quanteda.corpora` package that interests you. After loading the package, pick one of the following:

- **Amicus Curiae Briefs:** A collection of 102 advocacy briefs submitted to the U.S. Supreme Court during the *Regents of the University of California v. Bakke* and *Grutter/Gratz v. Bollinger* cases in 1978 and 2003, respectively. These two cases dealt with affirmative action admissions policies: the former at the University of California, Davis School of Medicine, and the latter at the University of Michigan Law School. The data come from Evans et al. (2007). To load these data, use `your.corpus.name <- data_corpus_amicus$documents`.
- **Movie Reviews:** A collection of 2,000 movie reviews from the IMDb archive. The data come from Pang, Lee, and Vaithyanathan (2002). To load these data, use `your.corpus.name <- data_corpus_movies$documents`.
- **Party Manifestos:** A collection of 101 manifestos from U.K. political parties, written between 1945 and 2005. The data come from Laver (1998). To load these data, use `your.corpus.name <- data_corpus_ukmanifestos$documents`.
- ***The Guardian* Articles:** A collection of 6,000 articles from *The Guardian*, published between 2012 to 2016, in the politics, economy, society, and international sections. Data come from Benoit et al. (2019). To load these data, use `your.corpus.name <- download("data_corpus_guardian")`, followed by `your.corpus.name <- your.corpus.name$documents`.

Once you have chosen and loaded the desired corpus, answer the following questions.

1. Provide 3-5 sentences telling me why you chose the corpus that you did. In your response, tell me (1) what custom-made dictionary you want to apply to it and why, and (2) why a sentiment analysis of this corpus might be insightful.
2. Take a look at the first document in the corpus using the `inspect()` function. Copy and paste the text into your Word document.
3. Convert your corpus into a `tm` package corpus. Create two different versions of the corpus. For example, create a `corpus.tm1` object and a `corpus.tm2` object. At this stage they should be identical to one another.
4. Using one of the copies of your corpus, use any preprocessors you think are necessary to clean the text **for an analysis with a custom dictionary** (which you will create momentarily). Tell me the preprocessors you applied, why you applied them in the order that you did, and copy and paste the cleaned text for the first document into your Word document.
5. Using the other version of your corpus, use any preprocessors you think are necessary to clean the text **for a sentence-level sentiment analysis**. Tell me the preprocessors you applied, why you applied them in the order that you did, and copy and paste the cleaned text for the first document into your Word document.

Section #2: Applying a Custom Dictionary

Now let's do some basic dictionary-based analyses.

1. Create your dictionary. It should contain at least three words. Keep each entry a unigram (single word) for now. Why did you choose the terms that you did?
2. Using the `str_detect()` function, tell me how many documents in your corpus used at least one of the terms in your dictionary. Copy and paste 1 or 2 of the texts into your Word document. (Be sure to use the version of the corpus you cleaned for the dictionary analysis.)

3. Convert your corpus to a DTM. Create a new DTM that arrays only the dictionary terms (as columns) across your documents (rows). Copy and paste the first few rows and columns of this new DTM into your Word document. What is this new DTM telling you?
4. Sum the dictionary term counts by document, and normalize this sum by the total document word count. What does this summation tell us, and why is it important to normalize by the total word count?

Section #3: Sentiment Analysis

Let's finish up with some sentiment analyses.

1. Using the `sentimentr` package, parse your corpus documents into sentences. (Be sure to use the version of the corpus you cleaned for the sentiment analysis.)
2. Calculate the sentiment polarity of each sentence in each document.
3. Pick one of the documents in your corpus. Visualize the sentence-level sentiment in that document using a histogram. Make sure the sentences are in sequential order. Provide 3-5 sentences explaining what the histogram is telling us.
4. Using the `extract_sentiment_terms()` function, what are some of the positive and negative words in the first few sentences of this document? Are there any terms identified as positive or negative that you think are problematic to be identified as such? Why?
5. Re-calculate the sentiment, where the sentiment score is aggregated by document (you can do this with the `sentiment_by()` function and setting `by = NULL` wihthin it). Now use the `highlight()` function to pull up an HTML page that color codes the documents. Take a screenshot of one of the documents (or part of the document, if it is really long) and put it in your Word document. What are the color codes telling us? Are there any sentences that seem to be color-coded incorrectly? If so, why do you think this might be the case?
6. Load up the `syuzhet` package. Get the raw counts for each of the emotion word dictionaries and normalize by total word count.
7. Create a box plot that visualizes the prevalence of the following emotion categories across the documents: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust (leave out negative and positive). What does the plot tell us?

Hints

The R scripts for the dictionary and sentiment analysis slides (pres41.R and pres42.R) might be very helpful. Like, *really really helpful*.

Be sure you don't load the `syuzhet` package until **after** you are done with the `sentimentr` package. These packages share some function names, so it can be problematic if you try to use both of them at the same time.

References

Benoit, Kenneth, Kohei Watanabe, Akitaka Matsuo, Gokhan Ciflikli, and Stefan M"uller. 2019. *Quanteda Initiative*. London, UK: Quanteda Initiative CIC. (link)

Evans, Michael, Wayne McIntosh, Jimmy Lin, and Cynthia Cates. 2007. "Recounting the Courts? Applying Automated Content Analysis to Enhance Empirical Legal Research." *Journal of Empirical Legal Studies* 4(4):1007-39. (link)

Laver, Michael. 1998. "Party Policy in Britain 1997: Results from an Expert Survey." *Political Studies* 46(2):336-47. (link)

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. "Thumbs Up? Sentiment Classification using Machine Learning Techniques." Pp. 79-86 in *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Philadelpia, PA: Association for Computational Linguistics. (link)