Lab #8: Word Embeddings

Due in Sakai March 29, 2019 (by 10:00a)

Introduction

For this lab, you will generate and interpret your own GloVe word embeddings (i.e., word vectors).

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 "_lab8." So my script would be titled **MTaylor_lab8.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_lab8.R
##  Note: Code for lab assignment #8
##  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_lab8.RData")
```

Lastly, prep a Word, Pages, or LaTeX document that has the same title structure: e.g., **MTaylor_lab8.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

For this lab, we are going to use articles from the 2006 Wikipedia data drump (English version). The data are made available through the Wikimedia Foundation, and this particular version with preprocessed and compressed texts comes from Matt Mahoney (for more information on Wikipedia data dumps, see here; for details on Matt Mahoney's compressed data file, see here).

To read in the data, run this code (from Selivanov 2017; see here):

```r
text8_file = "~/text8"
if (!file.exists(text8_file)) {
  download.file("http://mattmahoney.net/dc/text8.zip", "~/text8.zip")
  unzip ("~/text8.zip", files = "text8", exdir = "~/")
}
wiki = readLines(text8_file, n = 1, warn = F)
```

The data are already cleaned and in character vector form, so no need for any preprocessing.

Once you have the articles, do the following:

1. Generate your word vectors. Instead of removing terms using the `doc_proportion_min` and `doc_proportion_max` arguments, use `term_count_min = 5L` to remove terms that show up fewer than five times (Selivanov 2017). You can include bigrams in your analysis if you want, but this is not required. Note that this is a Wikipedia data dump, so the sky is the limit for what terms/names/concepts you can choose.

2. Select at least 5 words/names/concepts you find interesting (we'll call these your "main words"). (Note that this is a Wikipedia data dump, so the sky is the limit for what terms/names/concepts you can choose.) Calculate the cosine similarities between each of these main word vectors and the rest of the corpus. Array the resulting vector of cosine similarities in descending order, and copy and paste the top 10 context words associated with each of your 10 main words into your Word document.

3. Generate a word cloud for each of the 5 main words. Copy and paste the plots into your Word document.

4. What can you say about each of you main words, given their top 10 context words and the associated word clouds? That is, what do these high-similarity words tell you about your main words? In what ways do we mean when we say words are "similar" or not when we are comparing word embeddings?

5. Using some combination of addition and subtraction, "shift" the contexts for at least 3 of your main words. Re-calculate the cosine similarities between these "new" vectors and the rest of the corpus and array the similarities in descending order. Copy and paste the top 10 context words associated with each of these 3 (or 4 or 5) main words into your Word document.

6. Generate new word clouds that reflect the new contexts for your main words. Copy and paste the plots into your Word document.

7. What can you say about each of you main words, now that you have shifted their context?

Hints

The R scripts for the structural topic modeling slides (pres101.R) might be very helpful. Like, *really really helpful.*

References

Selivanov, Dmitriy. 2017. "GloVe Word Embeddings." Retrieved March 21, 2019 (link).