Text as Data: Homework 3

In this homework assignment we're going to compare the press releases of two senators—Richard Shelby and Jeff Sessions, Republican senators from Alabama. To make this comparison, we're going to download a bigger collection of Senate press releases and then focus on the releases from Shelby and Sessions.

We encourage you to spend some time processing these texts this week, because we will use this collection for the next homework assignment as well.

Downloading the Data

The press release collection are stored here:
`https://github.com/lintool/GrimmerSenatePressReleases`

Download the collection as a `.zip` file, unzip the file on your computer.

Creating a Document-Term Matrix

We're going to use the files from Richard Shelby and Jeff Sessions to make two different kinds of Document-Term Matrices. The first will consider only the 1000 most used unigrams, while the second (separate) DTM will use the 500 most common trigrams. To create the document-term matrices, use the following recipe.

1) Create two nested dictionaries for both the Shelby and Sessions press releases. The nested dictionary should contain, for each press release:

   - Month of release
   - Year of release
   - Day of release
   - Author (either Shelby or Sessions)
   - The text of the press release

   To create the nested dictionary:

   i) Use `os.listdir` to create lists of both the Sessions and Shelby press releases

   ii) The file names are formatted as `DayMonthYearAuthorNumber.txt`. Devise a parsing rule to extract the month, year, day, of the releases

iii) Store all the information in a nested dictionary

2) Next, we will find the 1000 most used unigrams and the 500 most used trigrams, after removing/simplifying a set of words

   i) discard punctuation, capitalization, and use `word_tokenize` to split the text on white space

   ii) Apply the Porter Stemmer to the tokenized documents.

   iii) Use the stop words from
   `'http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop'`
   Append to the list:

   * shelby
   * sessions
   * richard
   * jeff
   * email
   * press
   * room
   * member
   * senate

   Apply the Porter Stemmer to this list of stop words and discard all stemmed stop words from the press releases.

   iv) Form the list of trigrams using the `trigrams` function from `NLTK`

   v) Use a python dictionary to count the number of times each unigram is used and a second dictionary to count the number of times each trigram is used. These should be counts over the *whole corpus* (that is, both senators' press releases).

3) Identify the 1000 unigrams used most often and the 500 most often used trigrams. If you're writing trigrams to a csv to analyze somewhere else, be sure to represent each `tuple` without commas.

4) Write a document-term matrix, where each row contains
`Speaker, Count`$_1$`, Count`$_2$`, ..., Count`$_{1000}$
for unigrams, and
`Speaker, Count`$_1$`, Count`$_2$`, ..., Count`$_{500}$
for trigrams.

   Remember, if `foo` is a list, you can count the number of times `x` occurs with
   `foo.count(x)`

5) Write the document term matrix for the unigrams and trigrams to separate .csv files. Remember that you'll need to reformat the trigram `tuples` so that you don't end up with extra commas in your column names. We recommend defining a function in python that takes a `tuple`, like
`'wabash', 'college', 'best'`
and converts it to
`wabash.college.best`

Applying Word Separating Algorithms

1) Using the document-term matrix, for both unigrams and trigrams create the following three measures of word separation

   i) Independent linear discriminant$\rightsquigarrow$ measure used in Mosteller and Wallace (1963)

   ii) Standardized mean difference$\rightsquigarrow$ For each word J calculate:

$$\text{std diff} \;=\; \frac{\text{Difference in author means}}{\text{Standard error, diff. in means}}$$

   iii) Standardized Log Odds$\rightsquigarrow$ as described in Monroe, Colaresi, and Quinn (2009). To create the scores, set $\alpha_j = 1$

2) Create a plot for each of the measures that shows the most discriminating words. Some helpful functions are
`plot`, but set `pch = ''`
`text` allows the placement of texts on plots.
Can we learn anything about how Jeff Sessions and Richard Shelby present their work to their constituents?

3) Compare the discriminating measures in 3 plots. What are the primary differences across the measures?

Comparing Document Similarity

Using the trigram word document matrix, let's compare 100 Shelby press releases to 100 Sessions press releases.

1) Devise a method to sample 100 press releases from each senator's collection

2) Create the following six matrices:

   i) Euclidean distance between documents

   ii) Euclidean distance between documents with tf-idf weights

   iii) Cosine similarity between documents

iv) Cosine similarity between documents with tf-idf weights

v) Normalize the rows of the trigram document term matrix. For row i,

$$\boldsymbol{x}_i^* = \frac{\boldsymbol{x}_i}{\sum_{j=1}^{500} x_{ij}}$$

Then apply the `Gaussian` kernel to the normalized matrices

vi) Use the same normalization, but now with tf-idf weights. Apply the Gaussian kernel.

3) Using the matrices, identify the most similar (smallest distance) and dissimilar (greatest distance) press releases. Read the pairs of press releases—do they appear to actually be similar? Which method appears to perform best?