

Explorations with Textual Mining

Tidewater Big Data Enthusiasts

Chuck Cartledge

Developer

August 25, 2016 at 12:12am

Contents

List of Figures	i
1 Introduction	1
2 Analysis	1
3 Conclusion	6
4 References	6
A Misc. files	6

List of Figures

1	Histogram of all words from the party platforms.	3
2	Effect of processing all words from the part platforms.	4
3	Histogram of “important” words from the party platforms.	5

1 Introduction

Textual mining is a fun topic and gets a lot press when used for things like sentiment analysis, correlation, and finding connections between various Big Data data sets. One of the general areas that gets special attention is reducing the number of words being analyzed to only the “important” ones. The definition of “important” is domain specific, and in some cases possibly question specific. In this short report, we take the collective platforms for the Democratic, Libertarian, and Republican parties and parse them into “important” words.

2 Analysis

We wrote an R program (see Section A on page 6) to:

1. Read the party platforms,
2. Perform elementary “data wrangling” on the text (splitting the pdf into individual words, converting the words to lower case, removing punctuation marks, changing a word to its stem where possible, and removing numbers),
3. Display the “cleansed words” as a histogram (see Figure 1 on page 3),
4. Display how the total number of words changes during each of the above operations after the removal of “un-important” words (see Figure 2 on page 4), and
5. Display a histogram final set of words (see Figure 3 on page 5).

The text processing steps are (see Figure 2 on page 4):

- No action – raw strings are read from the platforms and not processed
- Remove 0 length – various string manipulations can result in a string have 0 characters, so those strings with 0 characters are removed
- Remove numeric – all strings that consist of just numbers are removed
- Remove punctuation – all punctuation marks are removed (this may change words when internal punctuation marks are removed)
- Remove short words – words with less than 3 characters are removed
- Remove stop words – generic “stop words” are removed from the list of words
- Split words – strings are split at each whitespace into smaller strings
- Stem words – suffixes are removed as much as possible to get to the “stem” word

- To lower case – all letters in all strings are folded from upper case to lower case
- Trim whitespace – leading and trailing whitespaces are removed from the strings

Top 50 words. Representing 28,188 of 66,835 (or %42) of all.

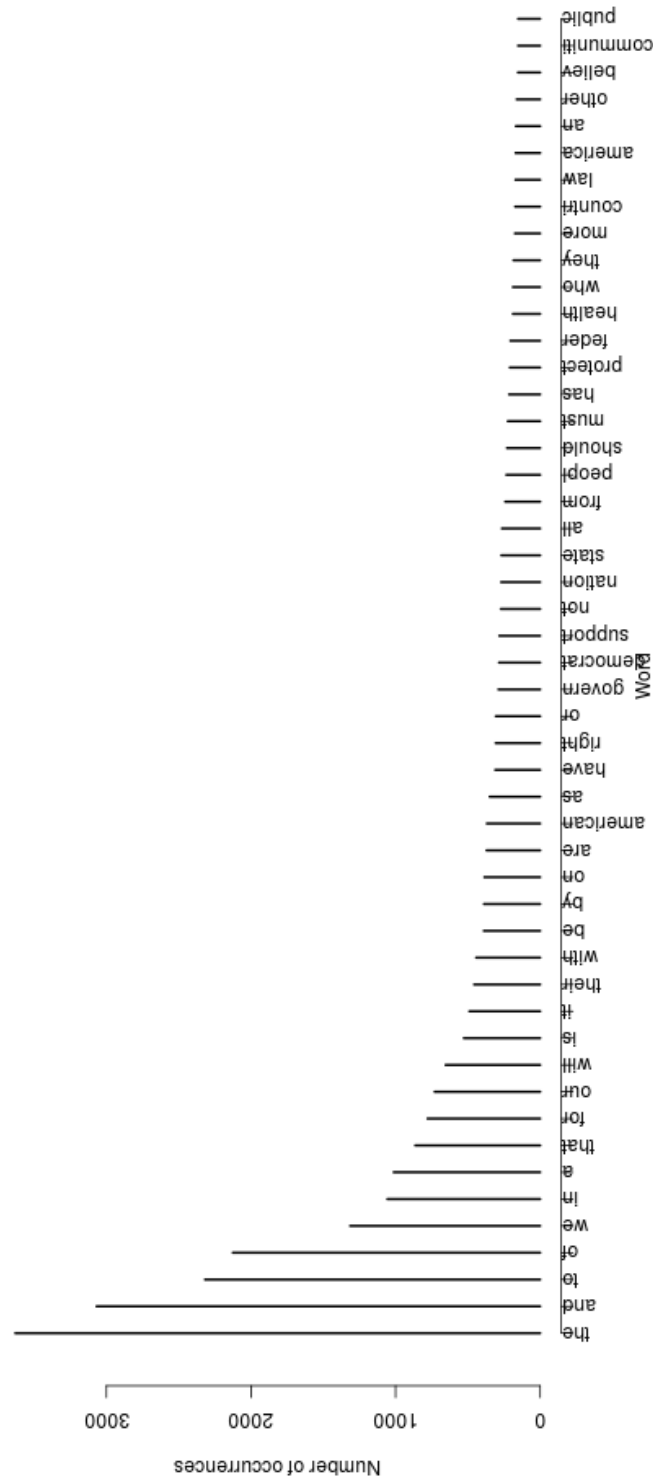


Figure 1: Histogram of all words from the party platforms. After rudimentary “data wrangling” to separate each word and remove numbers, the histogram shows how many times common English words appear in the text. These common words are known as “stop words” due to their high rate of occurrence. The high rate indicates that they add very little information to the body of the text, so the algorithms should “stop processing” them. Generic stop word lists include parts of speech like articles, adverbs, etc. It is also possible to augment the list with words that are specific to the problem domain, or the particular data set.

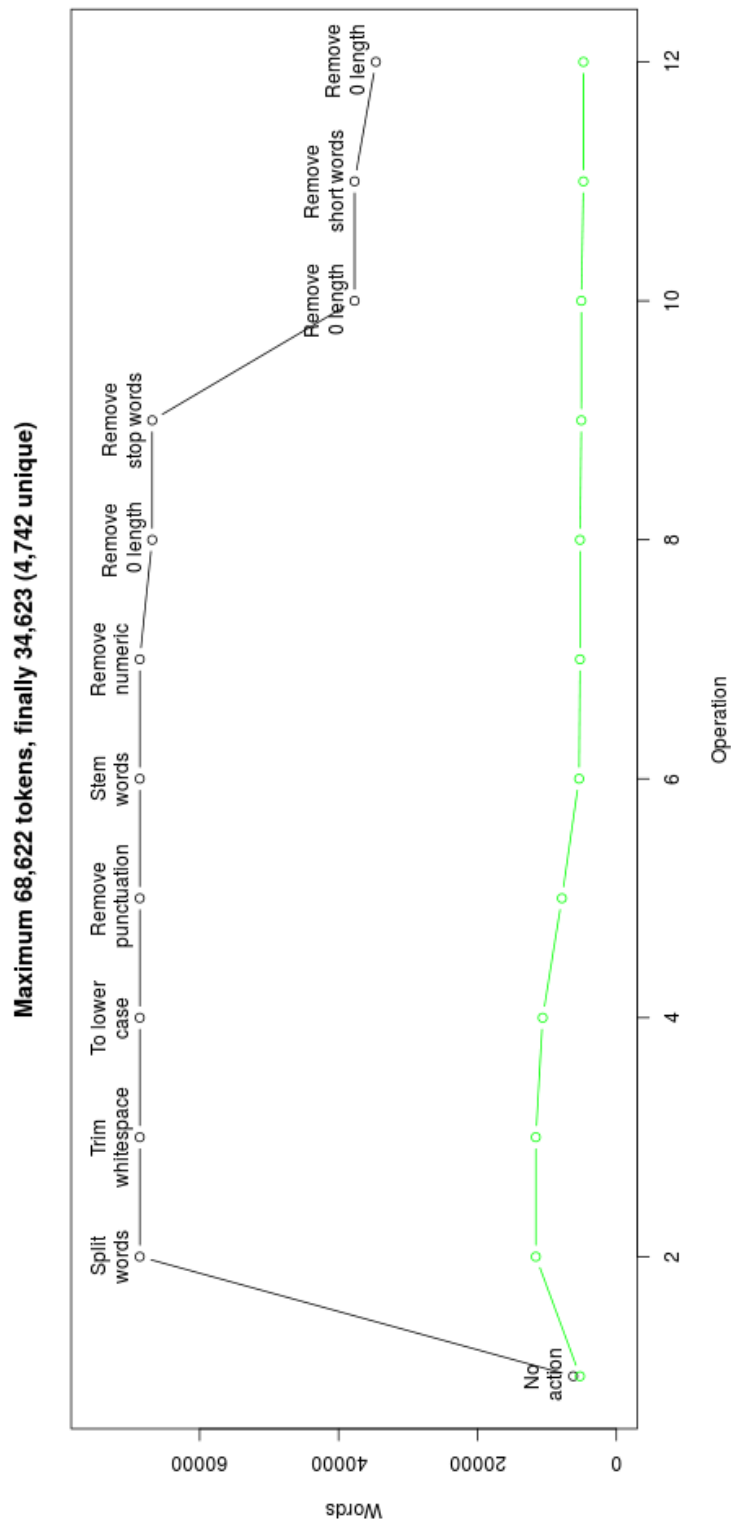


Figure 2: Effect of processing all words from the part platforms. The black annotated curve shows the processing that happened at each step. The green curve shows the number of unique words resulting from each processing step.

Top 50 words. Representing 7,442 of 34,623 (or %21) of all.

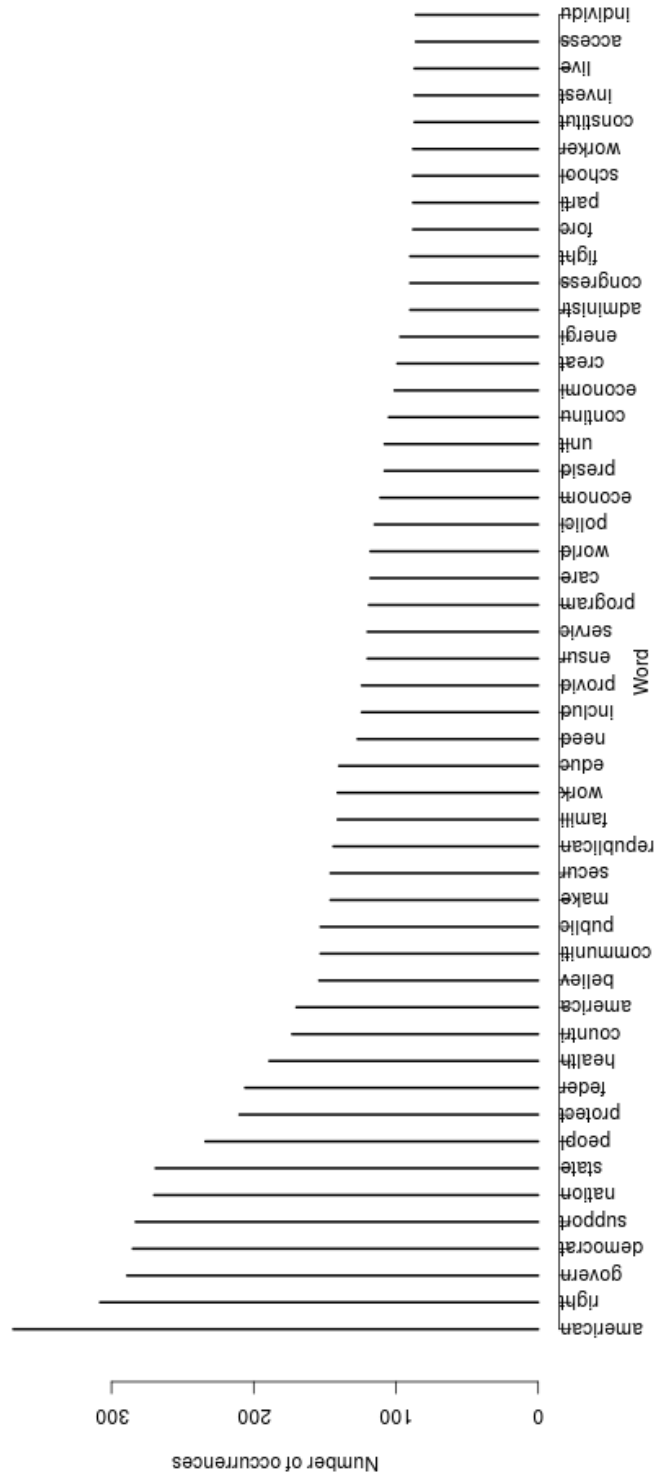


Figure 3: Histogram of “important” words from the party platforms. A histogram of words after all “un-important” words have been removed.

3 Conclusion





The order of processing probably has a small impact on the final outcome. Operations like “remove punctuation” could create incorrect words like: wont from won’t. This probably won’t happen too often (pun intended), but needs to be acknowledged. The 50% reduction in the number of words would have a significant positive impact on any detailed, or intensive later processing.

4 References

- [1] Democratic Party, *2016 democratic party platform*, <https://www.demconvention.com/wp-content/uploads/2016/07/Democratic-Party-Platform-7.21.16-no-lines.pdf>, 2016.
- [2] Libertarian Party, *Libertarian party platform*, <http://www.lp.org/files/2016LPPlatform.pdf>, 2016.
- [3] Republican Party, *Republican platform 2016*, [https://prod-staticngop-pbl.s3.amazonaws.com/media/documents/DRAFT_12_FINAL\[1\]-ben_1468872234.pdf](https://prod-staticngop-pbl.s3.amazonaws.com/media/documents/DRAFT_12_FINAL[1]-ben_1468872234.pdf), 2016.

A Misc. files

The files used to create all these figures are attached to this report. They are:

1. histogramWords.R - an R program used to analyze the party platforms and create histograms of the words. 
2. platform-Democratic.pdf - the Democratic party platform[1]. 
3. platform-Libertarian.pdf - the Libertarian party platform[2]. 
4. platform-Republican.pdf - the Republican party platform[3]. 
5. stopwords.txt - a collection of US English stop words. 