Developing Tools for Analysis of Text Data November 2016

Randall Powers, Brandon Kopp and Wendy Martinez

Office of Survey Methods Research, U.S. Bureau of Labor Statistics

Powers.Randall@bls.gov

Abstract: Many surveys at the Bureau of Labor Statistics have unstructured or semi-structured text fields. Most of these sources of text are not analyzed because users usually do not know what types of analyses can be done, and they lack easy-to-use and inexpensive tools to exploit the text data. This paper will describe an application that was developed to analyze survey text data.

Key words: Survey data; Text data; Text analytics workflow; Software tool development; Statistical learning

1. Introduction

Text analysis is the process of extracting information from written language, and it is an important activity for many Bureau of Labor Statistics (BLS) programs. For example, an analyst might read job titles to assign occupation classifications, check websites for the latest product and price information, or scan news articles to track important economic events.

Currently, there are no existing tools at BLS to look at open ended text data from survey interviews. Ultimately, we want a tool that will allow us to find themes amongst the interview data using word clouds and simple visualization, and to be able to export our results to a useful file. This kind of analysis can be done in R, but there's a steep learning curve. We wanted something simpler to use such that the analyst doesn't have to write his or her own R code. It can be done using SAS or SPSS, but those can be expensive, and don't allow the user control over the tools. We wanted something that we ourselves designed and customized.

Shiny is an R package that makes it easy to build interactive web applications straight from R. No knowledge of Javascript or HTML is necessary. All coding is done using R. Additionally, the package contains many useful functions and tools that the user would otherwise have to write from scratch. Hence, the amount of R coding that is actually necessary is greatly decreased. We determined that developing an application using the R Shiny package would best suit our needs.

This paper will describe the R Shiny application we developed. Each section will detail a separate application screen. These screens will include each of these tabbed items displayed in Figure 1.

2. The 'Welcome Screen' Tab

When the application is run, the Welcome Screen (Figure 2), the first of six tabs, appears by default. This screen gives the user information about file formats and within file formatting. The imported file must be in one of three formats: text-delimited (e.g. .csv, .tsv), Excel, or R (i.e. RDS file).

The data should be formatted so that the text information you are interested in exploring is in one column. Each row of the data should be a unique 'document.' That is, it should make sense as a unit. You might have one row (document) for each respondent to a survey.

The Welcome Screen example (see Figure 2a) uses a national parks dataset. Each text description of a national park is a text variable to be analyzed (referred to as the "document"), and

the user would have the option to group by various categories such as Region. The user would be looking for common themes among the descriptions of national parks.

Once the user has the data ready to go, they can go to the tab marked 'Step 1: Upload Your Data.

3. The 'Load Your Data' Tab

This tab enables the user to load their data file from anywhere on their computer. The file must be in either Excel, CSV, or R format. Please see Figure 3 for more details.

The user can optionally choose to use a stopwords list imported via an Excel file. This is a list of common words that are excluded from the analyses. For example, "the", "and", "but", etc. are often of little or no use in differentiating one document from another. The text analysis tool, by default, removes 175 stopwords, but a user may want to customize this list (or create their own), as some common words may be of use when classifying documents.

For demonstration purposes, a respondent burden dataset is used. The Office of Survey Methods Research at BLS conducted a survey in which respondents were asked a number of questions about expenditures and then were asked how burdensome they found the survey. To better understand their burden rating, respondents were asked to list an activity that they find "not at all burdensome", an activity that they find "somewhat burdensome", and an activity that they find "extremely burdensome". The open-ended description of activities at these different burden levels is what we will use to demonstrate the application.

Once the user has loaded his file, they can begin the analysis by clicking on the next tab, 'Exploratory Plots'.

4. The 'Exploratory Plots" Tab

On the third tab (shown in Figure 4), the user can look at wordcloud and frequency plots of their data. The user first specifies the text variable they wish to analyze. For the burden dataset example, any of the three comparison burden categories work equally well. In this example, the user does not need to choose a categorical variable for this set of data.

For other datasets, the user can choose a categorical variable to compare text. For example, with the national parks dataset that was on the welcome screen, we might choose geographic region as our categorical variable.

The NGrams slider creates longer word strings; it defaults to single word strings (unigrams), but can be increased to 2-word strings (bigrams), 3-word strings (trigrams), etc. Hence, the user can specify whether they want to analyze single words, or multiple word phrases. They can also choose to exclude certain words. When the user is done choosing their desired specifications, a word cloud and a chart with word frequency or word percentage (not pictured) is produced. When there is a categorical variable, the user can see which text it more prevalent in certain categories, and see the relative frequencies of the most common words in the frequency chart.

5. The 'Context Viewer' Tab

The user may wish to see the context in which a word or phrase was used. When the user chooses this tab (see Figure 5), the user can find a word or phrase in the text containing their search text. In the burden example, the user might wish to compare in what ways the word "watching" was used as part of strings for Not At All Burdensome activities.

6. The 'Clustering' Tab

The main feature of the application is the clustering tab (Figure 6). Document clustering involves the use of descriptors and descriptor extraction. The user must specify a few parameters

before results are produced. This includes choosing the text variable to analyze, the dimension reduction method, the N-gram size, as well as elimination of stop words and the use of stem words (i.e., truncating words so that base words can be combined). The user can choose to see the results as a frequency count, as binary (word present/not present), as the proportion in the document, or as the inverse document frequency. After the input parameters are specified, a number of results are produced.

These results (see Figures 6a and 6b) include a Document Clustering Plot. Based on the number of clusters specified, we see n different clusters. Cluster groupings are created using K-nearest neighbors. Here we're looking for tightly defined clusters so that we can look at their contents and see what makes them unique. One weakness of the current clustering system in the current text analysis application is that it compresses hundreds or thousands of terms into just two difficult to interpret dimensions. This is done for visualization purposes. Two dimensions may be too few to adequately capture the variation between documents. More dimensions will be allowed in future versions of the application.

Another result is a Word Dimension Plot, which shows the distribution of words along our two compressed dimensions. It can help us interpret the meaning of the two dimensions in the graph. Comparative Word Clouds are also produced. They show the dominant terms in each cluster and where terms were more strongly related to a particular cluster. A Top Five Terms per cluster chart is also produced. This lists the five most used terms in each cluster. A Documents Per Cluster table is also produced. This shows us the number of documents that contain terms in each particular cluster, thus giving the user an idea of how exclusive a cluster is. The more documents that appear in a cluster, the less exclusive. A Latent Semantic Variables Matrix, which s pairwise scatterplots of multiple dimension. As mentioned earlier, the application currently reduces the dataset to two dimensions for visualization purposes. This plot is an attempt to explore greater dimensionality in the data and perhaps find a pair of dimensions that creates more well-defined clusters. In a future version of the application, users will be able to select which pair of dimensions they want to be used for the primary analysis.

7. The 'Output Data" Tab

The Output Data Tab allows the user to output a term document matrix (or document term matrix) into an Excel or CSV file. Again, the user chooses which text and categorical variables to analyze. The user can choose to collapse the results for all documents into one row, or have the results by document. There are also have a number of options that were previously seen on the cluster tab.

8. Final Comments

The application is still currently in the development phase. The authors plan to make the application publicly available upon completion.

9. References

Bouchet-Valat, Milan (2014). SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library. R package version 0.5.1. https://CRAN.R-project.org/package=SnowballC

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie and JonathanMcPherson (2016). shiny: Web Application Framework for R. R package version 0.13.1. https://CRAN.R-project.org/package=shiny

Dahl, David B. (2016). xtable: Export Tables to LaTeX or HTML. Rpackage version 1.8-2. https://CRAN.R-project.org/package=xtable

Dragulescu, Adrian A.(2014). xlsx: Read, write, format Excel 2007 and Excel 97/2000/XP/2003 files. R package version 0.5.7. https://CRAN.R-project.org/package=xlsx

Feinerer, Ingo and Kurt Hornik (2015). tm: Text Mining Package. R package version 0.6-2. https://CRAN.R-project.org/package=tm

Fellows, Ian (2014). wordcloud: Word Clouds. R package version 2.5. https://CRAN.R-project.org/package=wordcloud

Martinez, Wendy and Alex Measure (2013). Statistical Analysis of Text in Survey Records. *Presented at Federal Committee on Statistical Methodology Research Conference*. https://fcsm.sites.usa.gov/files/2014/05/C3_Martinez_2013FCSM.pdf

Measure, Alex (2016). Bureau of Labor Statistics Text Analysis Team internal document.

Musialek, Chris. Philip Resnik and S.Andrew Stavisky (2016) Using Text Analytic Techniques to Create Efficiencies in Analyzing Qualitative Data: A Comparison between Traditional Content Analysis and a Topic Modeling Approach. *Presented at American Association for Public Opinion Research Conference*.

R Core Team (2015). foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ... R package version 0.8-66. https://CRAN.R-project.org/package=foreign

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.URL http: www.R-project.org/

Solka, Jeffrey L. (2007). *Text Data Mining: Theory and Methods. Statistics Surveys*, **2**, 94–112. https://projecteuclid.org/euclid.ssu/1216238228

Wickham. Hadley (2016). scales: Scale Functions for Visualization. Rpackage version 0.4.0. https://CRAN.R-project.org/package=scales

Wickham, Hadley (2007). Reshaping Data with the reshape Package.Journal of Statistical Software, 21(12), 1-20. URL http://www.jstatsoft.org/v21/i12/.

Wickham. Hadley (2015). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.0.0. https://CRAN.R-project.org/package=stringr

Wickham, Hadley and Romain Francois (2015). dplyr: A Grammar of Data Manipulation. R package version 0.4.3. https://CRAN.R-project.org/package=dplyr

Wickham, H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

Wild, Fridolin (2015). lsa: Latent Semantic Analysis. R package version 0.73.1. https://CRAN.R-project.org/package=lsa

Xie, Yihui(2015). DT: A Wrapper of the JavaScript Library 'DataTables'. R package version 0.1. https://CRAN.R-project.org/package=DT

Figure 1: Application Tabs

Figure 2: The 'Welcome Screen' Tab

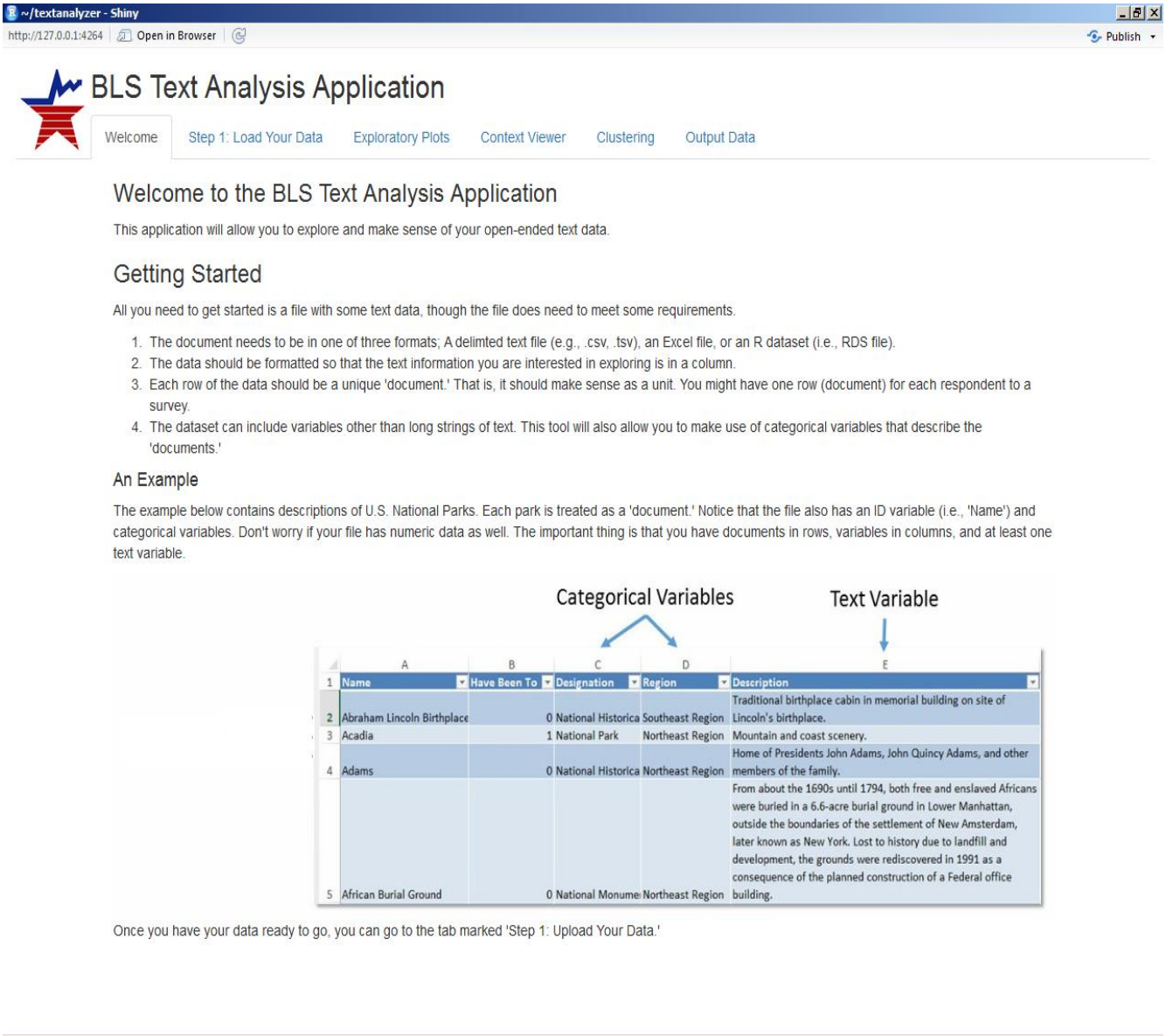
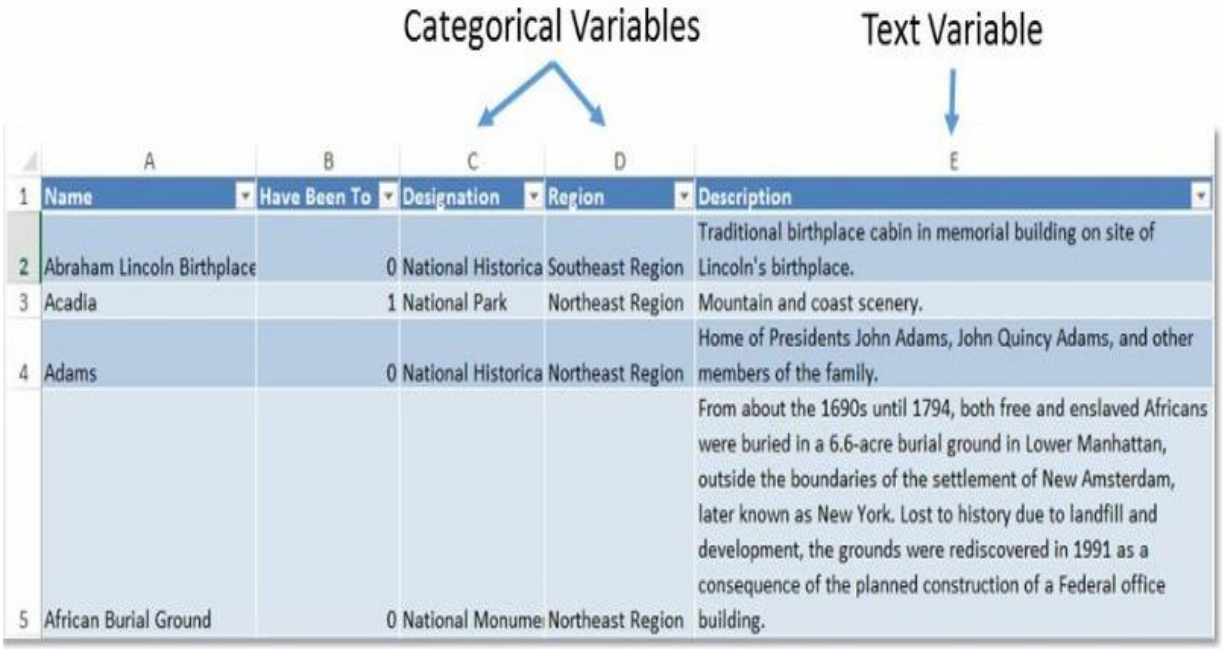


Figure 2a: The 'Welcome Screen' Tab (Example)



	A	B	C	D	E
1	Name	Have Been To	Designation	Region	Description
2	Abraham Lincoln Birthplace	0	National Historica	Southeast Region	Traditional birthplace cabin in memorial building on site of Lincoln's birthplace.
3	Acadia	1	National Park	Northeast Region	Mountain and coast scenery.
4	Adams	0	National Historica	Northeast Region	Home of Presidents John Adams, John Quincy Adams, and other members of the family.
5	African Burial Ground	0	National Monume	Northeast Region	From about the 1690s until 1794, both free and enslaved Africans were buried in a 6.6-acre burial ground in Lower Manhattan, outside the boundaries of the settlement of New Amsterdam, later known as New York. Lost to history due to landfill and development, the grounds were rediscovered in 1991 as a consequence of the planned construction of a Federal office building.

Figure 3: The Load Data Tab

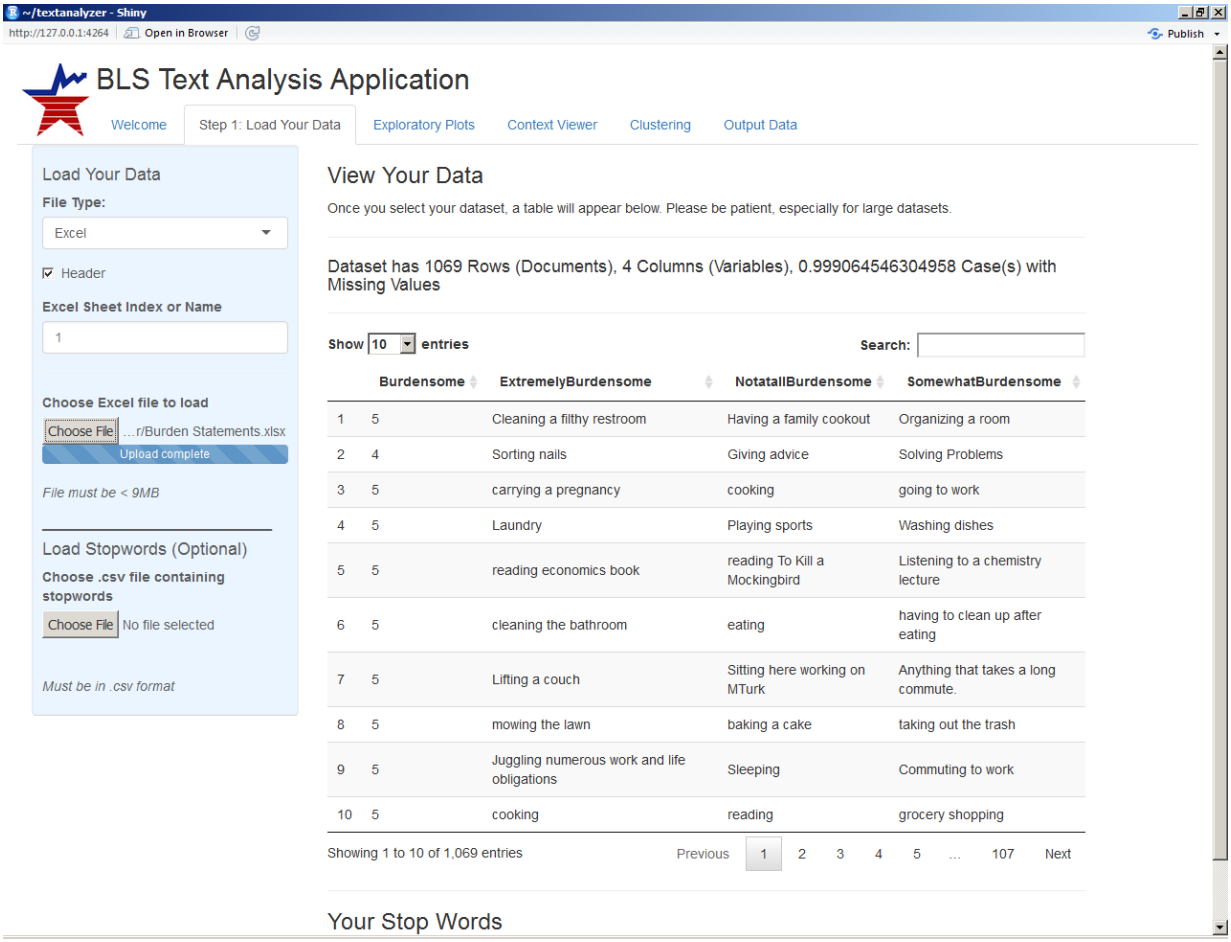


Figure 4: **The 'Exploratory Plots" Tab (with output)**

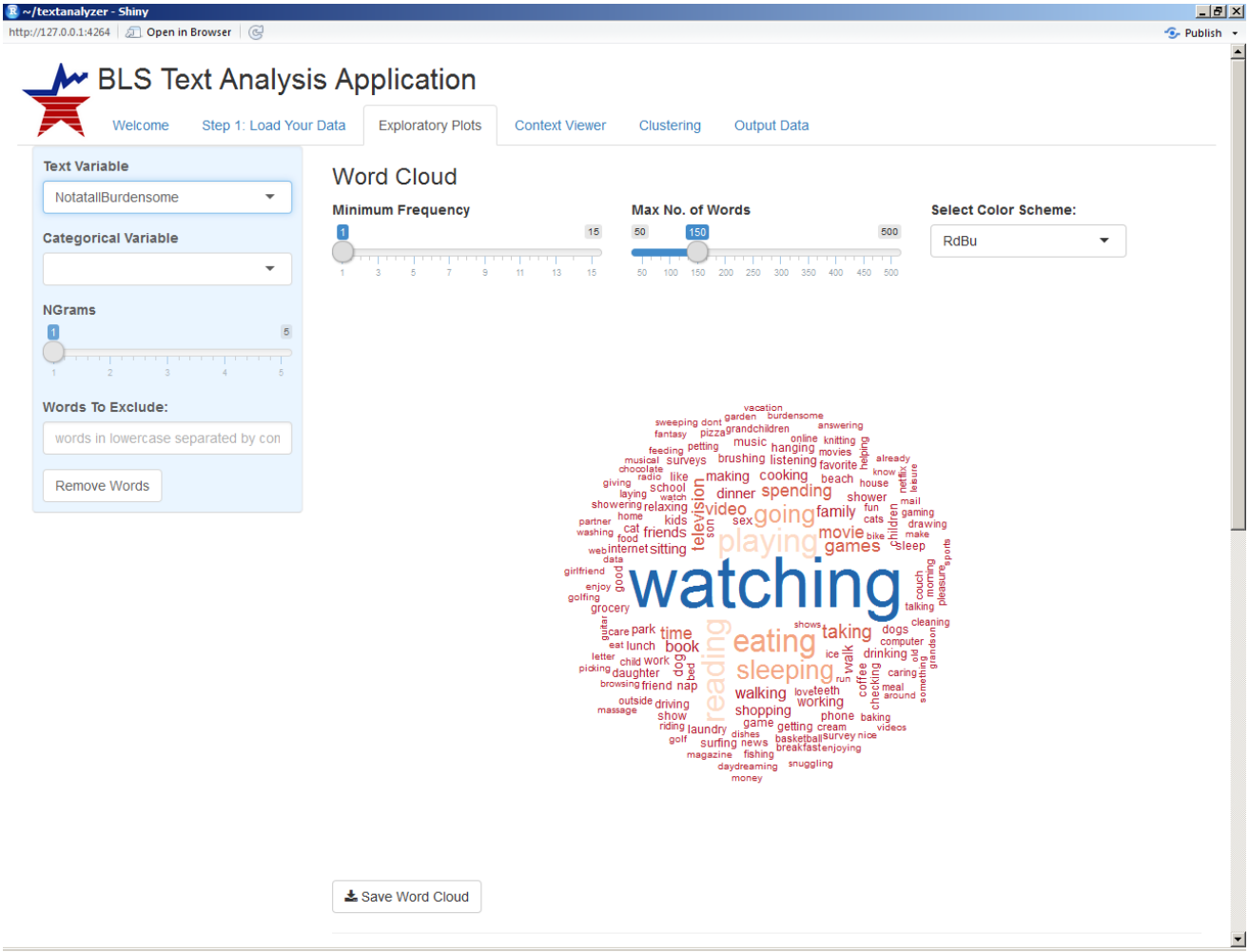


Figure 4a: The 'Exploratory Plots" Tab (with output, #2)

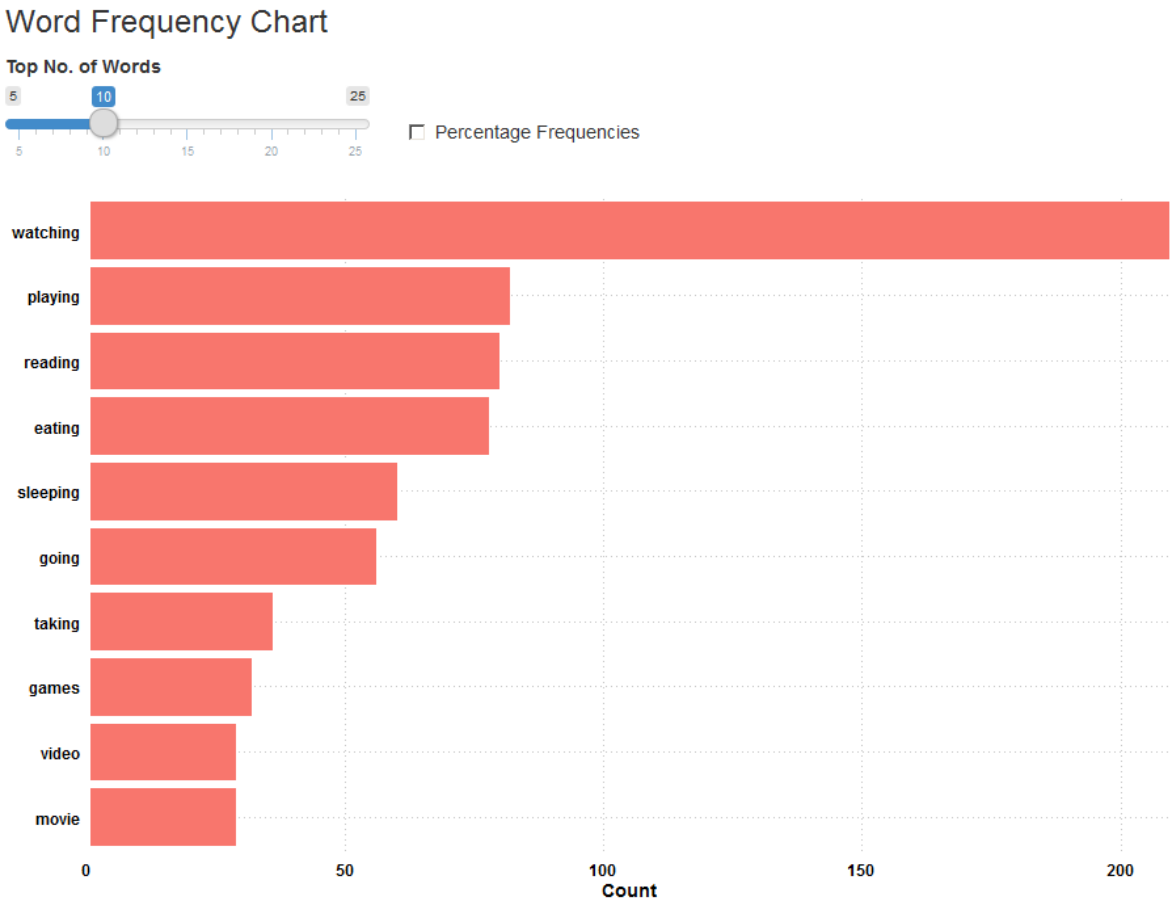


Figure 5: Context Viewer Tab (output)

Show 10 entries Search:

	Term	Sentence
15	Watching	Watching TV
22	Watching	Watching television
26	Watching	Watching TV
30	Watching	Watching a movie at the theater.
39	Watching	watching a comedy show on television
40	Watching	watching TV
48	Watching	watching television
58	Watching	Watching TV
59	Watching	watching television
68	Watching	watching TV

Showing 1 to 10 of 210 entries Previous 1 2 3 4 5 ... 21 Next

Figure 6: The 'Clustering' Tab

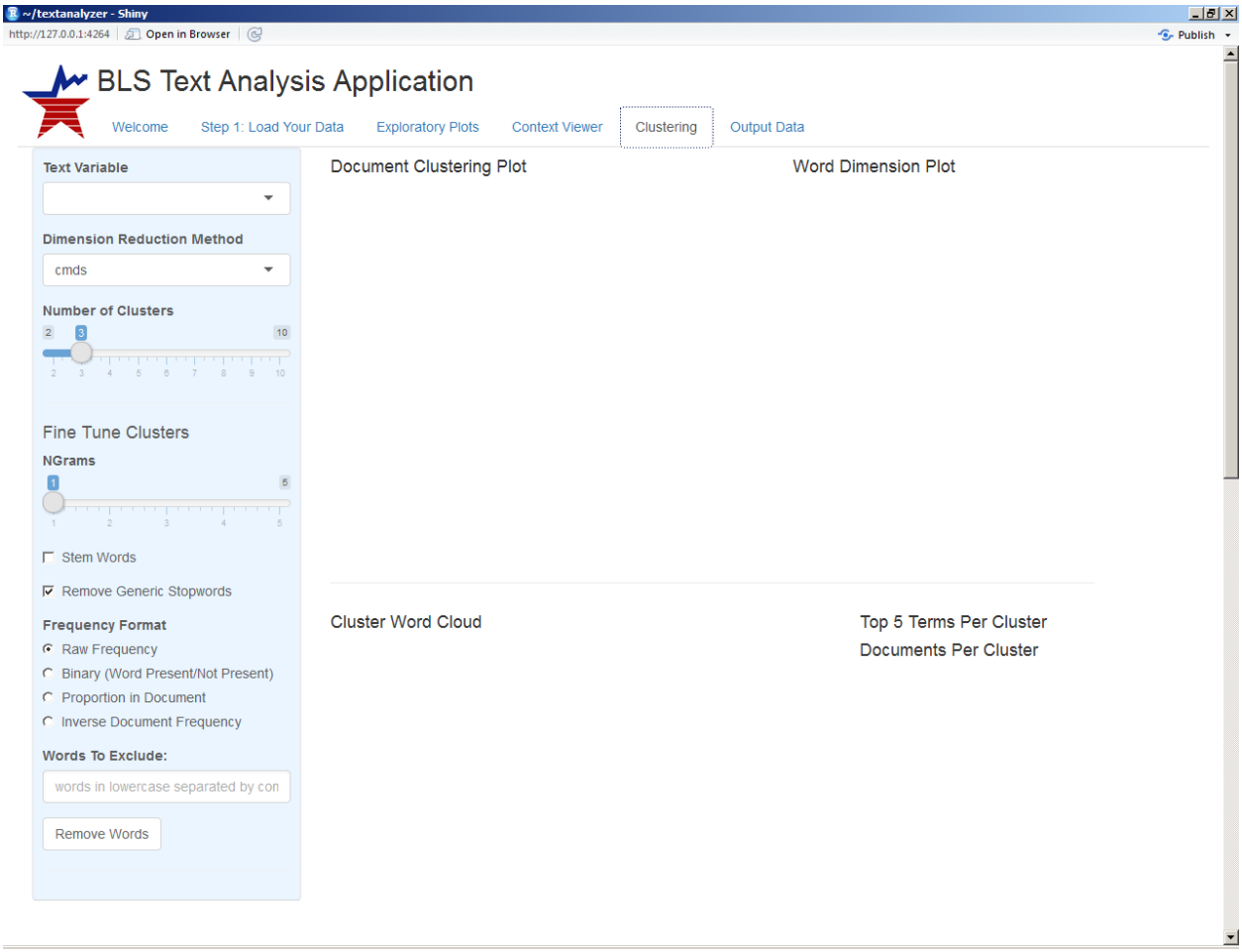


Figure 6a: The 'Clustering' Tab (output #1)

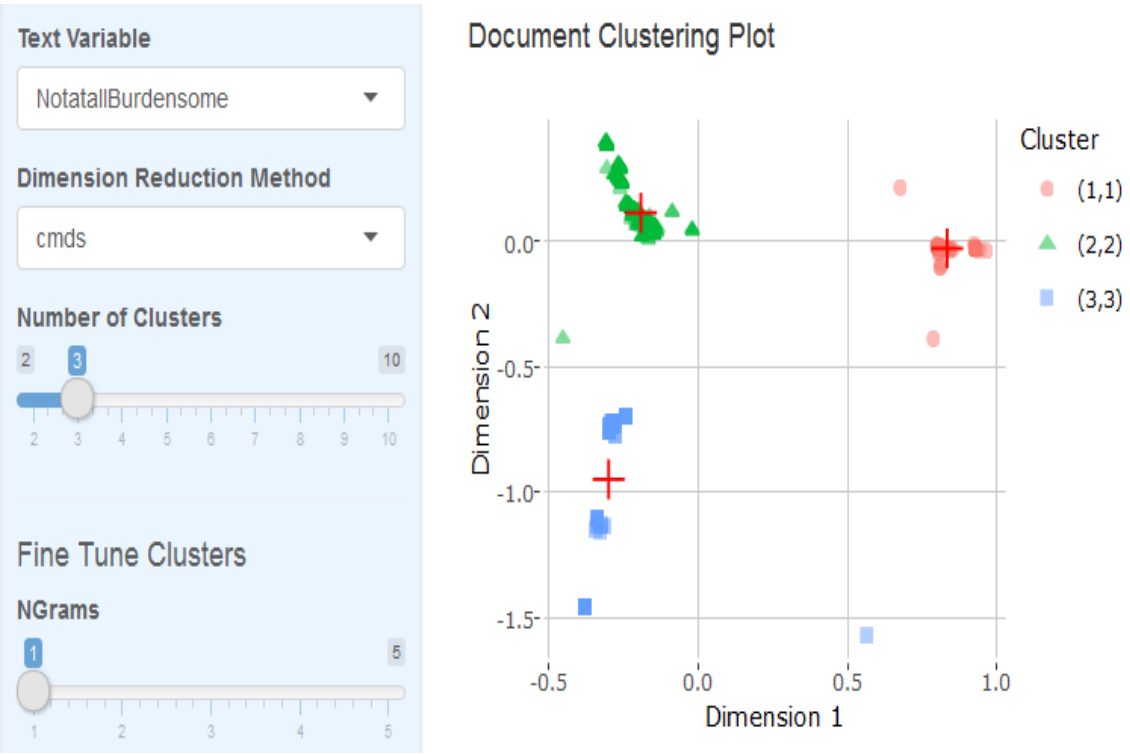


Figure 6b: The 'Clustering' Tab (output #2)

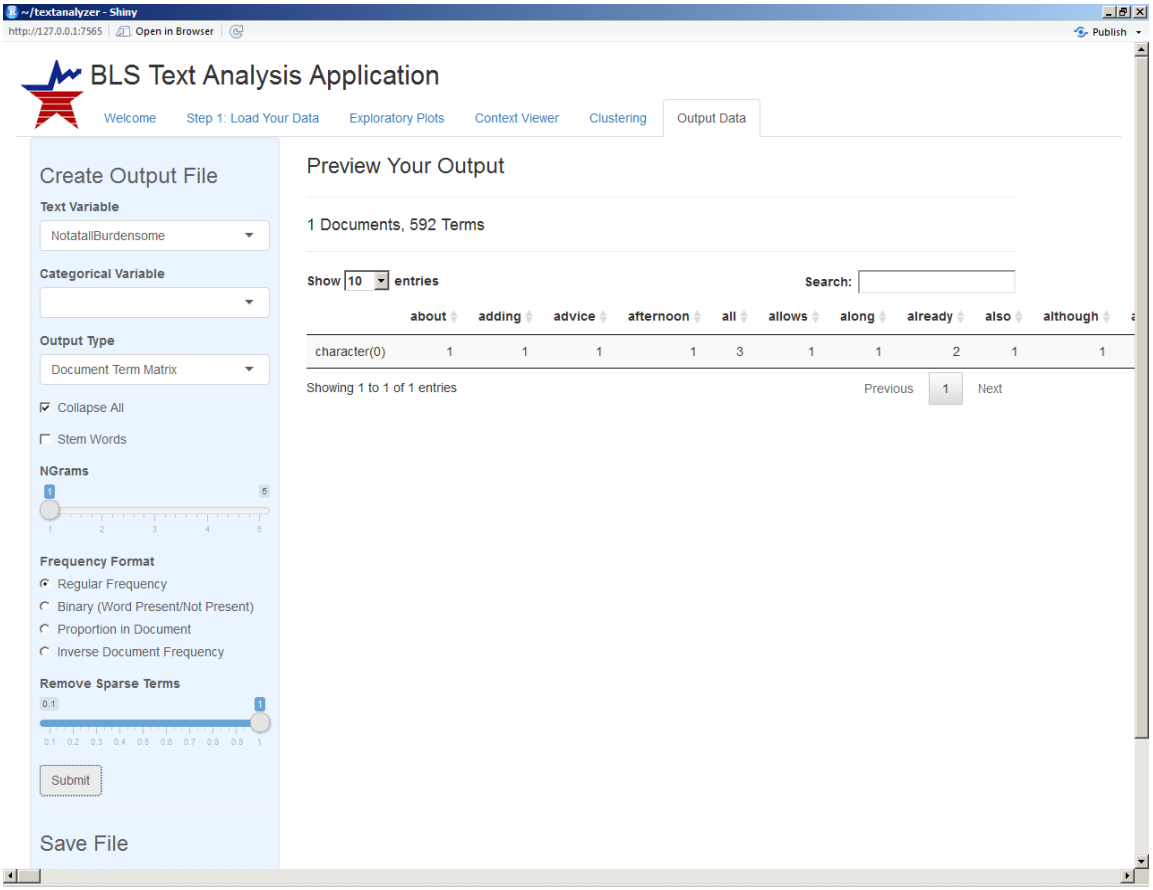



Figure 7: The 'Output' Tab