



TUTORIAL

# Text and Sentiment Analysis

---

Copyright © 2020 by DecisionPro Inc.

This document is primarily intended to be used in conjunction with the Enginius® software suite. To order copies or request permission to reproduce materials, go to <http://www.enginius.biz>. No part of this publication may be reproduced, stored in a retrieval system, used in a spreadsheet, or transmitted in any form or by any means –electronic, mechanical, photocopying, recording or otherwise– without the permission of DecisionPro, Inc. v220819

## Overview

Text and Sentiment Analysis refers to a body of analytic techniques to evaluate a corpus of text to develop insights about a product, company, service, etc. The corpus could come from a wide range of sources such as user comments from review sites, tweets, or from text available at web sites, 10-K reports, and the like. When humans interpret text, we use our understanding to interpret the contents of the text, and the emotional intent behind various words to infer whether a sentence, document, or corpus has a positive or negative tone, and perhaps whether words and sentences in the corpus express various emotional states (e.g., joy, anger, surprise).

In text analysis, we attempt to understand the contents and meaning of the text by exploring issues such as, what are the underlying themes or topics within a text corpus? What does the text tell us about the author(s)? What topics are trending? What keywords (a contiguous set of words) best characterize the contents of the corpus? These questions are typically easy for humans to answer by reading the text. But most of us can potentially process perhaps 50 pages of text per hour, whereas top text analysis systems can analyze 200,000 pages per hour or more. We can use text analysis to create a summary of the contents of the text corpus, extract “important” words and phrases from the corpus and identify related words and concepts. Important outputs of text analysis are word clouds summarizing the frequency of occurrence of various words (or word combinations, such as bigrams or trigrams), and the co-occurrence patterns of the words which help us understand the overall contents and meaning of the corpus. Other types of analyses include document similarity, namely, the extent to which one or more documents within a corpus are similar to other documents.

In sentiment analyses we attempt to summarize the sentiments and emotions embedded in the text corpus, which are subjective aspects of textual content, and not just their objective contents -- we attempt to detect, extract, and assess value judgments, subjective opinions, and the emotional contents in text data. Data useful for text and sentiment analysis are widely available today and they can provide companies valuable insights for making business decisions. The typical sources of data used for text and sentiment analysis in marketing come from sources such as Facebook and Instagram postings, Tweets, reviews from sites such as Yelp and TripAdvisor, as well as from product/service specific reviews collected by the supplier of the product or service.

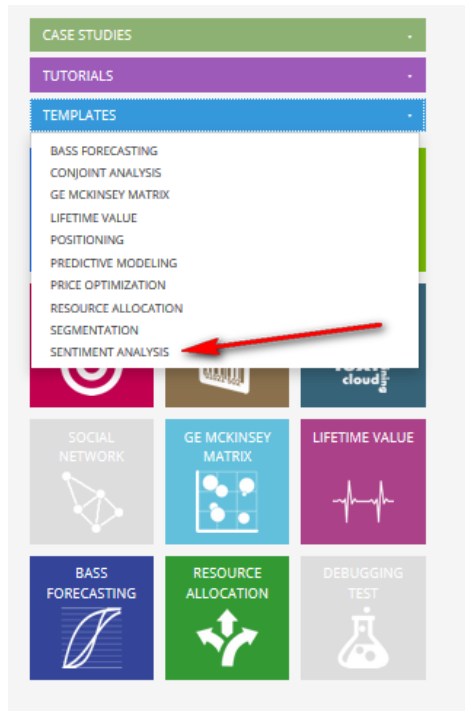
In sum, the summaries generated from text and sentiment analyses provide at least a high-level overview of the meaning and emotions embedded within a text corpus.

## Getting Started

You can use your own data or use a template preformatted by the Enginius software. Because the analytical models underlying Sentiment Analysis require a specific data format, users with their own data should review the appropriate preformatted template to become familiar with the data structure. The next section explains how to create an easy-to-use template to enter your own data.

## Creating a Template

From the Enginius Dashboard, click the Templates dropdown and select Sentiment Analysis to open the dialog box to create a template.



The following dialog box will appear:

**Sentiment Analysis**
⊗

This will generate a sentiment analysis template, with appropriate placeholders.

**Data source**

Sentiment data

Twitter data

Web pages

**Options**

Number of verbatim 20

Include date

Include rating (e.g., 1-5)

**Stop words**

Include customized stop words

**Random data**

Fill with random data (for illustration purpose only)

? Help
■ Cancel
▶ Run

Select the options desired for the sentiment analysis and click “Run” to generate the template for entering your data.

Note, the options for each of the data sources are different and users are encouraged to go through the template generation process to fully understand the model data requirements.

## Data source

- **Sentiment Data:**

This option requires a data block containing the text to be analyzed. The format of the required data is shown below, where the first column contains a unique id for each row, Verbatim data is the text to be analyzed for each respondent. The optional Date and Rating fields specify the date of the corresponding text and a rating (from a scale of 1-to-n) where the respondent has supplied a rating to accompany their text.

**SENTIMENT ANALYSIS TEMPLATE**

This template contains a placeholder for sentiment analysis with in-house data. To exclude specific stop words from the analysis, list them below.

Sentiment analysis d				
\	Verbatim	Date	Rating	
1	Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	2018-10-21	1	
2	Sunt in culpa qui officia deserunt mollit anim id est laborum.	2018-10-13	4	
3	Consectetur adipiscing elit.	2018-09-14	3	
4	Ut enim ad minim veniam.	2018-07-31	1	
5	Sunt in culpa qui officia deserunt mollit anim id est	2018-11-24	5	

Custom stop words		
\	Stop words	
1	news	
2	content	
3	feature	
4	video	
5	website	

- **Twitter Data:**

When analyzing Twitter data, no data template is needed, and you will receive an error message if you try to generate a template. To run Sentiment analysis with Twitter data, simply select the Sentiment Analysis icon to open the model and select the Twitter handle or content you would like to analyze (see **Run Analysis** section in tutorial for more details).

- **Web pages:**

The template generation dialog box allows you to specify the number of websites to be analyzed and whether to include custom stop words.

**Sentiment Analysis**
✕

This will generate a sentiment analysis template, with appropriate placeholders.

**Data source**

Sentiment data  
 Twitter data  
 Web pages

**Options**

Number of websites

**Stop words**

Include customized stop words

**Random data**

Fill with random data (for illustration purpose only)

Help
Cancel
Run

After clicking “Run”, the following template will be generated:

**SENTIMENT ANALYSIS TEMPLATE**

This template contains a placeholder for sentiment analysis by parsing the content of websites.

**Sentiment analysis**
✕
📄
📄
🔍
↓
↑

\	Websites	
1	http://news.yahoo.com	
2	http://news.google.com	
3	http://www.huffingtonpost.com	
4	http://www.cnn.com	
5	http://www.nytimes.com	

## Entering Your Data

\*If using Twitter data as your data source, you may skip directly to the Run Analysis section of the tutorial as there is not additional data that needs to be entered.

For Sentiment or Web pages data, it is recommended to use the Template feature or at least review the template format to ensure that your data is the correct format. You may enter your data in Enginius in one of three ways:

1. Enter your data directly into the Enginius online template.
2. Copy and paste your data from Excel (or other data source) to Enginius.
3. Download the Enginius template to Excel (using the Save function in Enginius), fill out the data in Excel (make sure to adhere to the template format), and then upload the Excel file back to Enginius (using the Load function in Enginius).



For the remainder of this tutorial, we will use the “OfficeStar: Sentiment Analysis” data set that loads when you open the Enginius Sentiment tutorial.

The screenshot shows the Enginius Marketing Engineering Online interface. The top navigation bar includes icons for SETTINGS, RESET, SAVE, LOAD, REPORT HISTORY, PROTECT ACCOUNT, and QUIT. Below the navigation bar, there are QUICK LINKS for 'RUN SENTIMENT ANALYSIS' and 'TUTORIAL IN PDF FORMAT'. The main content area is titled 'OFFICESTAR: SENTIMENT ANALYSIS' and contains a table of sentiment analysis data. The table has columns for Verbatim, Date, and Rating. The data is as follows:

	Verbatim	Date	Rating
1	Checkout was too slow	10/29/2013	3
2	Great selection of office products	10/26/2013	4
3	Very helpful sales people	10/26/2013	4
4	Did not have a good selection of Macs	10/25/2013	2
5	Prices were the best in this area	10/23/2013	4
6	Limited selection of office furniture	10/23/2013	3

## Run Analysis

When you click on the Run Sentiment Analysis button (or Sentiment Analysis icon), the analysis setup window will display. The set-up window will vary depending on your Data source.

The 'Sentiment Analysis' setup window is displayed. It contains the following sections and options:

- Data source:** Radio buttons for Sentiment data (selected), Twitter data, and Web pages.
- Data options:** Fields for Sentiment data (Sentiment analysis data), Verbatim (Verbatim), Date (Date), and Rating (Rating). Checkboxes for Include date and Include rating (e.g., 1-5) are checked.
- Stop words:** Default stop words (Long list (851)) and Custom stop words (Sentiment analysis data).
- Advanced options:** A checkbox for Advanced is unchecked.

At the bottom, there are buttons for Help, Cancel, and Run.

## Data source

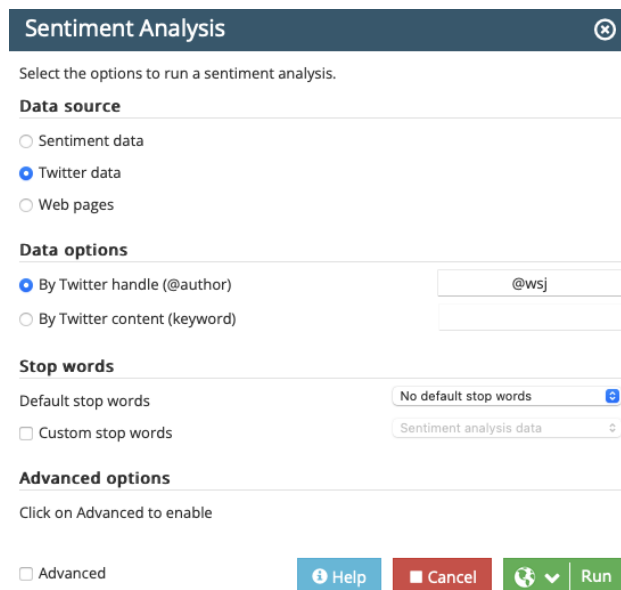
You need to first select the appropriate **Data source** for the data you are analyzing. Once selected, the remainder of the setup window will adjust to obtain the required data appropriate for analyzing that type of data.

### Sentiment data

When choosing Sentiment data as your data source, the Sentiment data analysis has the Data options as shown above. You will need to select the Enginius data block that contains the Sentiment data and then identify the Verbatim (text to be analyzed), Date (if included), and Rating (if included) columns in that data block.

### Twitter data

When the Twitter data option is selected, the data options are either a Twitter handle (e.g. @wsj) or Twitter keyword (e.g. stock market). The software retrieves up to 1,000 corresponding tweets directly from Twitter over the past week to analyze. Note: when selecting Twitter handles or keywords, multiple selections may be made by separating each with commas (e.g., @wsj, @penn\_state).



The screenshot shows a configuration window titled "Sentiment Analysis" with a close button in the top right corner. Below the title is the instruction "Select the options to run a sentiment analysis." The window is divided into several sections:

- Data source:** Three radio button options are listed: "Sentiment data", "Twitter data" (which is selected), and "Web pages".
- Data options:** Two radio button options are listed: "By Twitter handle (@author)" (selected) and "By Twitter content (keyword)". A text input field next to the selected option contains "@wsj".
- Stop words:** Two options are listed: "Default stop words" (selected) and "Custom stop words". The "Default stop words" option has a dropdown menu showing "No default stop words". The "Custom stop words" option has a dropdown menu showing "Sentiment analysis data".
- Advanced options:** A checkbox labeled "Advanced" is present, which is currently unchecked. Below this checkbox is the text "Click on Advanced to enable".

At the bottom of the window, there are three buttons: "Help" (blue), "Cancel" (red), and "Run" (green). The "Run" button includes a refresh icon and a dropdown arrow.

### Web pages

When choosing Web pages as your data source, the Sentiment data analysis has the data options as shown below. You will need to select the Enginius data block that contains the list of Web pages that you want to analyze.

**Sentiment Analysis**
⊗

Select the options to run a sentiment analysis.

**Data source**

Sentiment data  
 Twitter data  
 Web pages

**Data options**

Web pages Sentiment analysis data ⊕  
 Enter URLs of Web pages you would like to parse (one per line).

**Stop words**

Default stop words Long list (851) ⊕  
 Custom stop words Sentiment analysis data ⊕

**Advanced options**

Click on Advanced to enable

Advanced

Help
Cancel

↻ ⌵ Run

## Stop words

Typically, text contains many words, called “stop words” (e.g., “and,” “the,” “of,” etc.), that do not perform a significant lexical function. It is best to remove these words to facilitate interpretation of the text. Enginius offers the option of removing a long list of stop words (851), a short list (174), or retaining all the words for analysis (see Appendix B for a list of stop words included in the Enginius lists).

Enginius also offers the option of including additional custom stop words that can facilitate interpreting documents in a specific domain. You can incorporate these custom words by generating a custom stop word list for a specific application. Often, an initial word cloud generated from the data could provide hints about possible custom stop words, e.g., product names such as “Pepsi” or popular words that do not add any meaning in a given context (e.g., app, game, iPhone). Here is the format for the custom stop word data block:

Custom stop words		
\	Stop words	
1	news	
2	content	
3	feature	
4	video	
5	website	



Be aware that removing stop words can affect the interpretation of your text significantly. For example, the Short and Long List of stop words included with Enginius includes words such as “didn’t” and “not” which can radically change the meaning of a phrase:

Before stop words	After stop words
The product is really very good <b>(Positive)</b>	product really good <b>(Positive)</b>
The product seems to be good <b>(Positive)</b>	product seems good <b>(Positive)</b>
I didn't like the product <b>(Negative)</b>	like product <b>(Positive)</b>
The product is not good <b>(Negative)</b>	product good <b>(Positive)</b>

It is recommended to test your analysis both with and without stop words to see how your analysis performs.

## Advanced options

As shown in the dialog box below, Enginius also provides advanced analysis options which you can access by clicking on the “Advanced” checkbox in the lower left corner. Checking “Advanced” will result in two additional options: “Word co-occurrence analysis and RAKE” and “Topic Model”. Instead of single word analysis as in a word cloud, we can analyze relationships between words. Specifically, we can explore co-occurrence of pairs of words within individual documents or within the entire corpus. The word co-occurrence analysis is then supplemented with RAKE (Rapid Automatic Keyword Extraction) that allows for exploration of the occurrence of more than two contiguous words, which allows extraction of important longer sequences of words that help to characterize the contents of the documents. Further details about these methods are provided in the Appendix.

### Sentiment Analysis

Select the options to run a sentiment analysis.

**Data source**

Sentiment data  
 Twitter data  
 Web pages

**Data options**

Web pages

Enter URLs of Web pages you would like to parse (one per line).

**Stop words**


Default stop words   
 Custom stop words

**Advanced options**

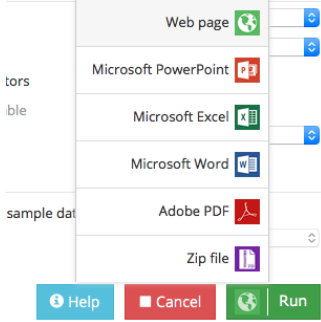
Word co-occurrence analysis and RAKE  
 Topic Model

Advanced


After selecting all the options, click the Run button found at the bottom of the Sentiment analysis setup window to begin the analysis. By default, the report will output as a web page.



Reminder: Clicking the globe icon beside the “Run” option will allow you to choose a different output format for the report.



You will see a pop-up indicating the progress of the sentiment analysis. Your report will output in the format chosen (Microsoft, PDF, or Zip format may automatically download to your hard drive).



Sentiment analysis is a time-consuming process and may take several minutes for a report to be generated.

## Interpreting the Results

The report generated by sentiment analysis contains several sections, depending on the options chosen. The results described below were generated with these model settings:

**Sentiment Analysis**
✕

Select the options to run a sentiment analysis.

**Data source**

Sentiment data  
 Twitter data  
 Web pages

**Data options**

Sentiment data	Sentiment analysis data
Verbatim	Verbatim
<input checked="" type="checkbox"/> Include date	Date
<input checked="" type="checkbox"/> Include rating (e.g., 1-5)	Rating

The leftmost column (in grey) should contain a unique id. The date should be in an unambiguous format, such as yyyy-mm-dd or yyyy/mm/dd.

**Stop words**

Default stop words	Long list (851)
<input type="checkbox"/> Custom stop words	Sentiment analysis data

**Advanced options**

Word co-occurrence analysis and RAKE  
 Topic Model

5 topics

IMPORTANT: Topic model analysis may take a long time to complete and may even appear to hang. Please be patient and do not relaunch the analysis. If you leave, the results will later appear in your report history.

Advanced

Help
Cancel
Run

## Word cloud

The first section presents the word cloud developed from the data. The word cloud is a graphical representation of the words found in a corpus, where the size and location of a word in the cloud diagram indicates the prominence of the word in the text. A word cloud is not the output of a statistical model per se. Rather it is an analysis of the input text for frequency of occurrence of various words after preprocessing the text by using such techniques as removing stop words (typically common words such as the, and, how, but, etc. which are unlikely to add meaning to a sentence), and stemming to reduce the number of words in a document by getting to the stem or root (e.g., buyer, buyers, buying, etc. could be put into a stem word "buy").

Enginius outputs two versions of word clouds, the first based on the stems (roots) of words and the second based on the un-stemmed words in the text, as in the example below:

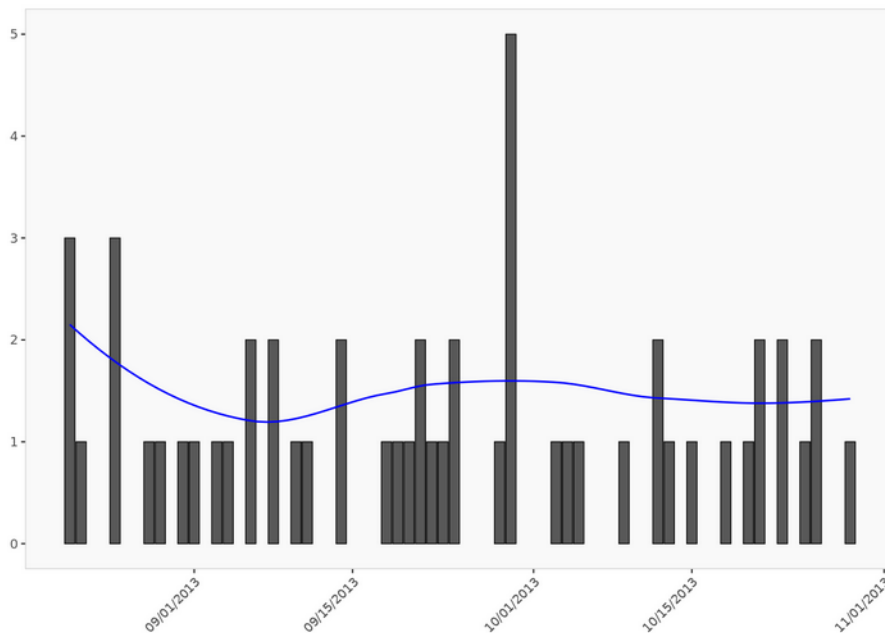


**Word cloud without stemming.** The word cloud represents the most frequently used words inside the corpus of texts provided. The bigger a word appears, the larger the number of times it occurs in the text corpus.

## Sentiment analysis

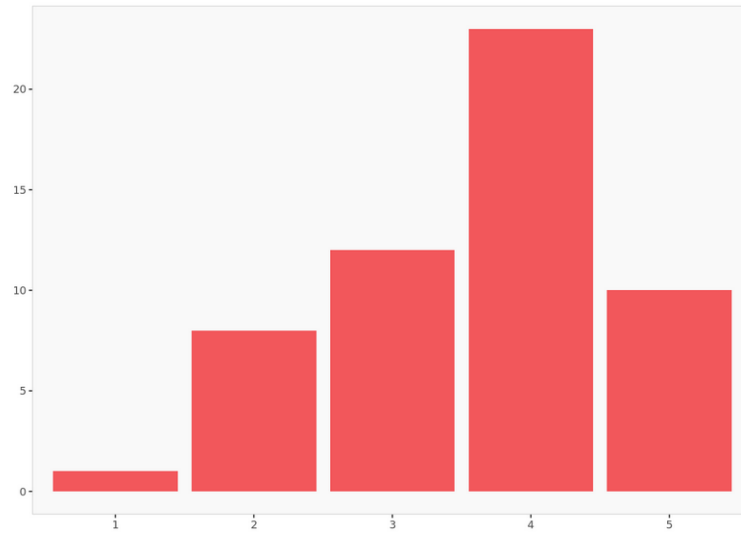
If the input text corpus contains information about the date of a specific review/document and an associated rating, then the Enginius output includes additional summaries as shown below (for data set consisting of user reviews and ratings of products posted on various dates):

**Post frequency**



**Post frequency.** The post frequency histogram indicates the daily frequency of posts. The blue smoothing line helps visualize the trend.

### Rating histogram



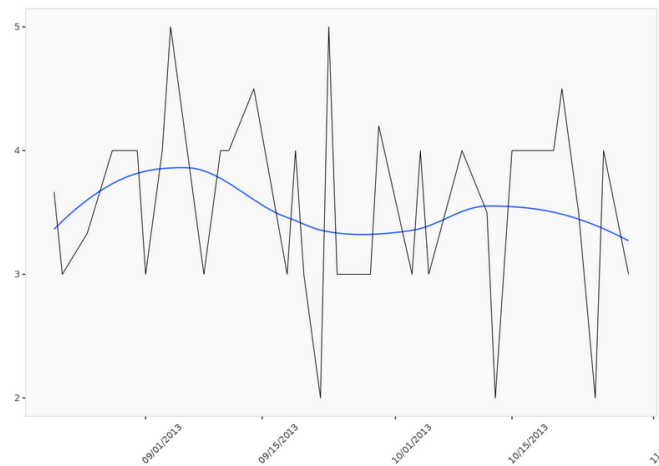
**Rating histogram.** The rating histogram indicates the number of posts by rating value.

### Rating frequency

	Frequency	Relative frequency
<b>Total</b>	54	100%
<b>1</b>	1	2%
<b>2</b>	8	15%
<b>3</b>	12	22%
<b>4</b>	23	43%
<b>5</b>	10	19%

**Rating frequency.** Row names indicates the rating value, then frequency gives the number of post associated to this rating.

### Average rating by dates



**Average rating.** The average rating graph indicates for each date the average rating of the posts. The blue smoothing line helps to visualize the trend.

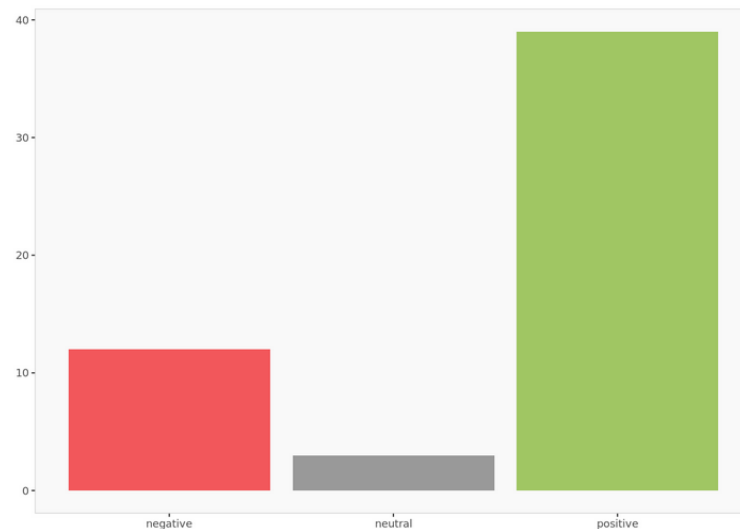
## Valence analysis

The next section shows the overall valence of the text corpus, a popular application of sentiment analysis. Here, we develop an overall valence or polarity of a text corpus. Valence is the technical term for the “subjective inclination” of a document, measured along a Positive/Neutral/Negative continuum. Several dictionaries or lexicons are publicly available (e.g., AFINN, NRC, Bing) which contain the valence weights associated with common words and phrases. This type of analysis can be applied to determine the polarity of a sentence, document, or the whole corpus. The Enginius output is for the entire corpus. (If you need the polarity for a subset of the corpus, you can re-run the analysis by including only the selected text).

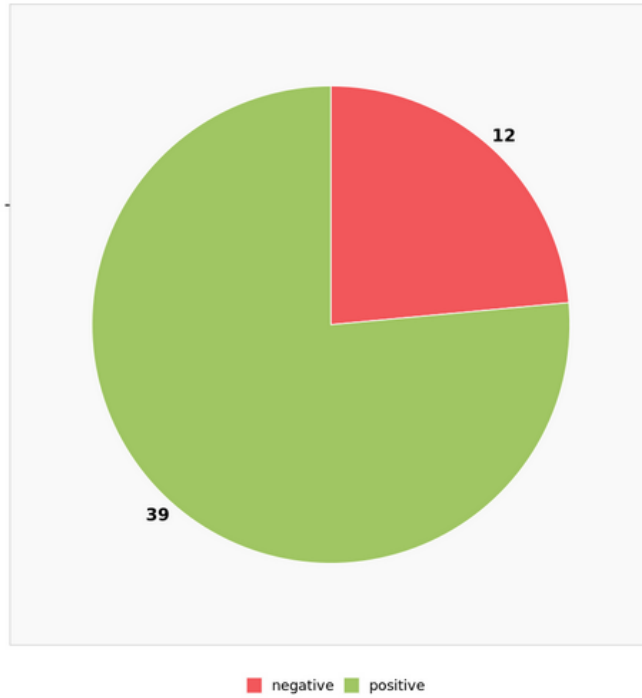
	Posts count	Relative posts count
<b>Total</b>	54	100%
<b>negative</b>	12	22%
<b>neutral</b>	3	6%
<b>positive</b>	39	72%

**Valence repartition.** The number of posts that fall into different valence categories summarized by their absolute and relative values

Two additional outputs are useful in analyzing the results, a Valence histogram and a Valence pie chart:



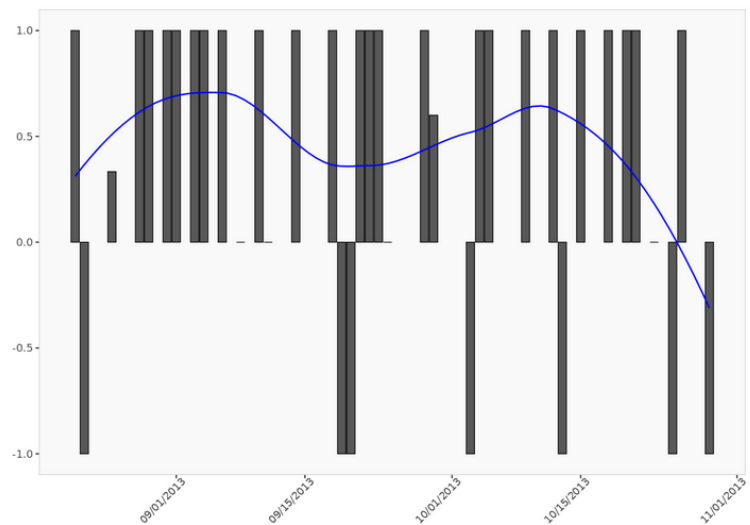
**Valence histogram.** The valence histogram indicates the number of posts by their valence



**Valence distribution without uncategorized posts.** Graphic summary of the relative sizes of the number of posts classified by valence after ignoring posts that could not be categorized (i.e., neutral posts)

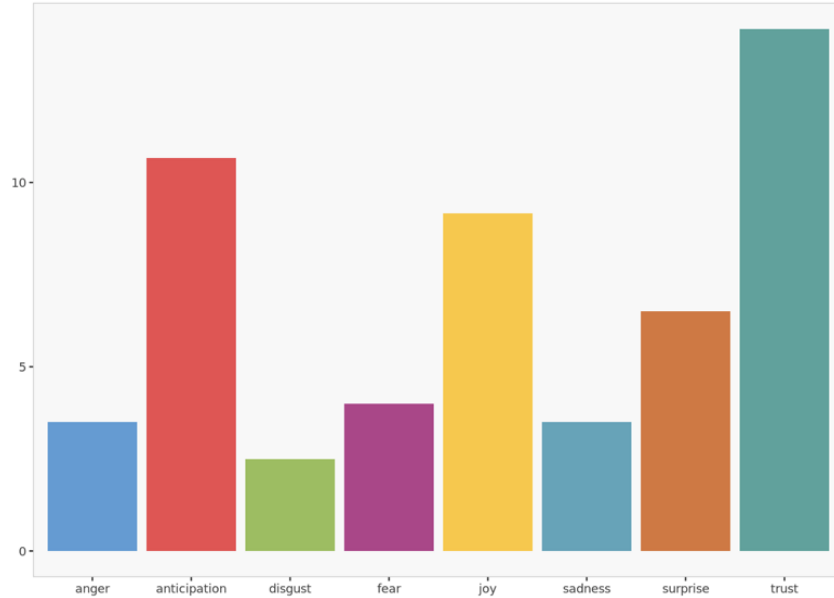
Also included in the sentiment output is the valence evolution, which shows the evolution over time if Date was included in the analysis:

**Valence evolution**

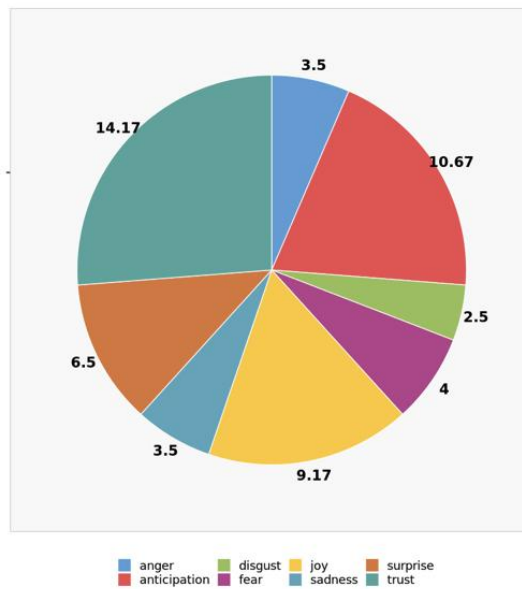


**Post valence ratio.** The post valence ratio graph indicates the daily average number of positive posts. The blue smoothing line helps visualize the trend.





**Emotion histogram.** The emotion histogram indicates the number of posts by their emotion.



**Emotion distribution without uncategorized posts.** Graphic summary of the relative sizes of the number of posts classified by emotion after ignoring uncategorized posts.

Using emotions associated with various words as determined from publicly available emotion lexicons (for common emotions such as happy, excited, discouraged, disappointed), we can generate a word cloud of emotions associated with a corpus (shown below).

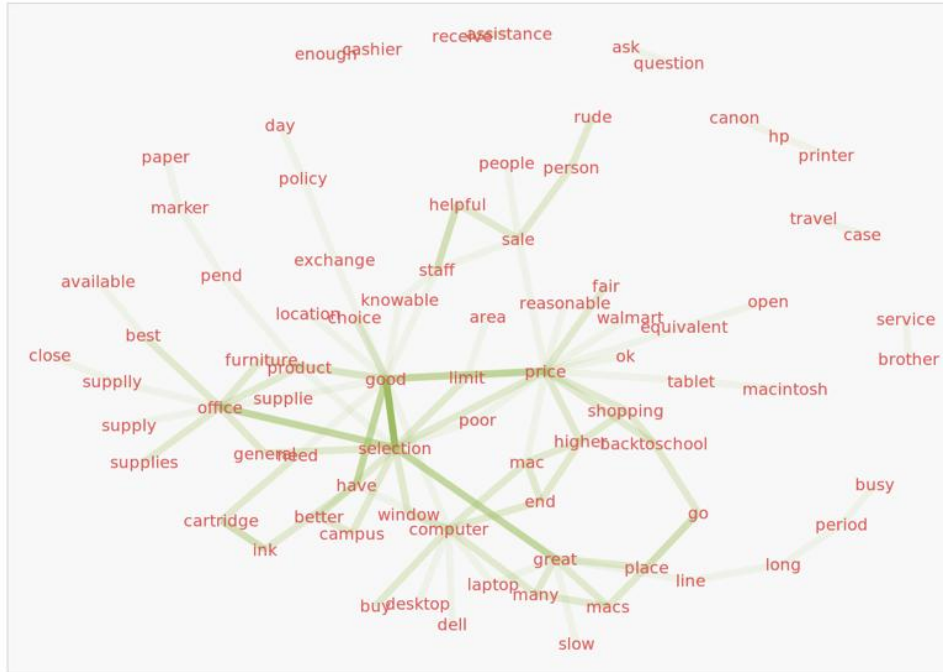


**Emotion word cloud.** Even if a post has multiple emotions its words will be shown only in one of those emotions.

## Results from Advanced analysis options

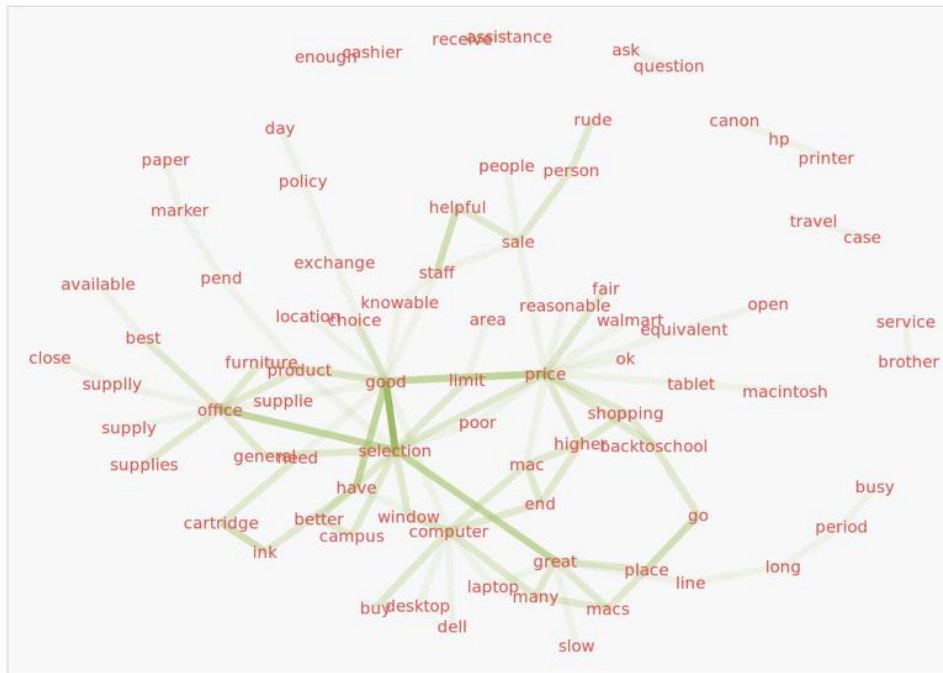
### Word co-occurrences between adjacent words in a corpus

The first output in this section is a word co-occurrence graph that summarizes the relative frequencies of pairs of words that are adjacent to each other when we consider the entire corpus as a single document. For every word, adjacent words provide a context. The relative frequencies of pairs of words are compiled into a “co-occurrence matrix” and the important elements from this matrix are displayed as a graph. We only include for display words that are likely to most useful for interpretation, namely, nouns, verbs, or adjectives. In the output here, we see that “good selection” “great selection” “ink cartridge,” “helpful staff” etc. go together often relative to other word pairs in the corpus. The strength of a connection is highlighted by the depth of the color of the line connecting two words.



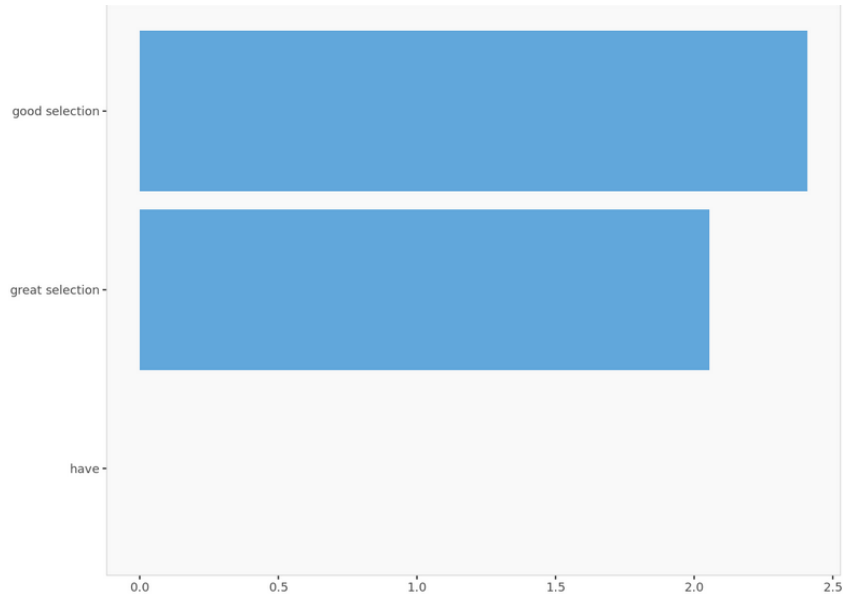
### Word co-occurrences between adjacent words in a corpus

The next graph shows word co-occurrence when we consider word co-occurrences within sentences, i.e., the pair of words can occur *anywhere* within a sentence in a document, i.e., the context is broader than the adjacent word restriction for co-occurrence in the graph we considered above (see Appendix A for additional details). Again, we display only nouns, verbs, and adjectives. Although there are substantial similarities between the two graphs for this data set, the analysis at the sentence level could alter the strengths of the connections between words.



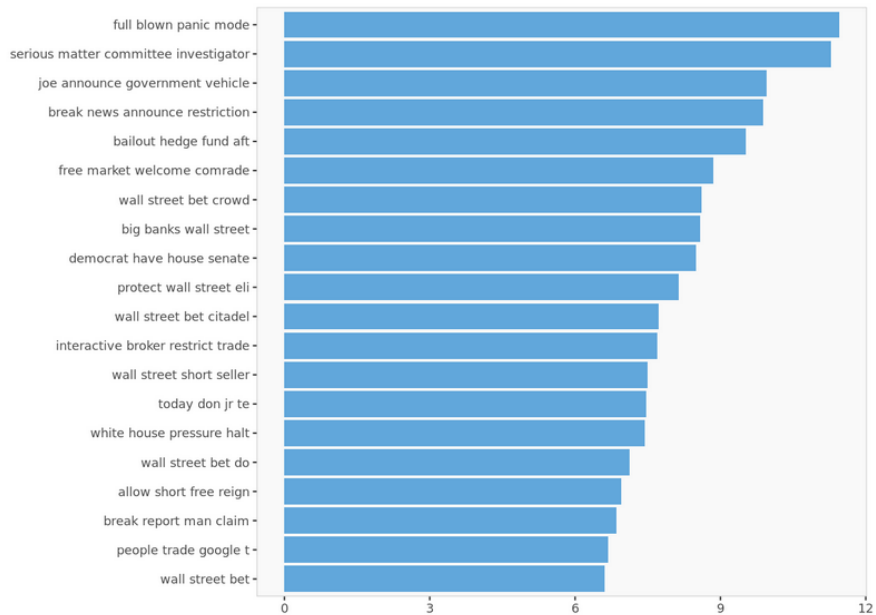
## RAKE Analysis (Rapid Automatic Keyword Extraction)

The next set of results summarizes output from RAKE (Rapid Automatic Keyword Extraction) (see Appendix A for additional details). The small data set used in this tutorial is not sufficient to generate an interesting output. We basically see the same results that we get from the sentence-level co-occurrence graph, showing that two bigrams (“good selection” and “great selection”) are the most important keywords in this text corpus.



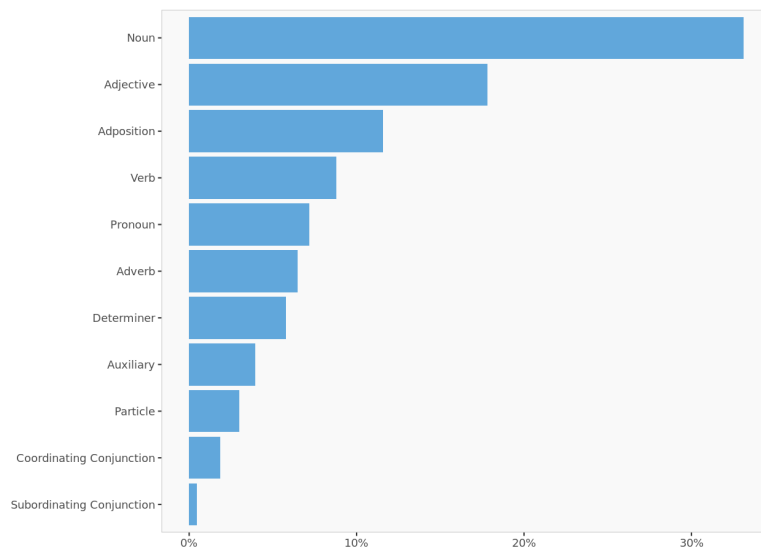
**Keywords with highest RAKE values.** The top keywords (i.e., contiguous sequence of words ignoring irrelevant words) were identified with minimum frequency of occurrences set to 0.01% of total word count

With larger text corpora, we will get more informative RAKE analysis outputs, as in the example below, which shows keywords which are informative. **Note: the example below was not generated from our OfficeStar data but is a representative output from RAKE analysis.**



**Keywords with highest RAKE values.**

A related piece of information that Enginius provides is distribution of parts of speech tags in the corpus. Here nouns, verbs, and adjectives (which are the only words shown in the co-occurrence graphs above) make up about 60% of all the words in this corpus.



## Topic Model

The final set of results from the advanced analysis is the topic model, which refers to a class of machine learning methods for analyzing and classifying text into broad thematic categories to help us understand the underlying structure of the contents of a text corpus (see Appendix A for additional details). Topic modeling is a data summarization tool for text, like what factor analysis is for summarizing numeric data. Each topic represents a theme (a grouping of words), but the analysis does not provide you a label for each topic. It is up to you to interpret the theme captured by each topic by examining the most important words in a topic. Although the example data here is too small to develop a good topic model that is interpretable, we can nevertheless see some patterns. For example, the first topic is likely about the characteristics of a retailer – carries computers, is in town, can walk to shop, good price, etc. It also contains some discrepant themes, such as “rude” and “slow.” In Enginius, you can pick up to 15 topics, and for each topic, it reports the top 15 words. Although each topic is likely to contain a different set of words, there may be some words that may appear in more than one topic, indicating their importance in several topics. Note also that the model is specified on the stemmed text, which adds to the challenges of interpreting the topics.

The output also includes two metrics to help you assess the adequacy of the topic model: Mean topic coherence and Mean topic exclusivity. Good models will have a mean topic coherence close to zero and mean topic exclusivity greater than 1. The model here has a high exclusivity score, suggesting the topics refer to different themes. There is no clear benchmark to evaluate the computed value of mean topic coherence. Roughly speaking, with 15 words in a topic (which is what is built into Enginius), a mean topic coherence score that is smaller than -240 (i.e., a larger negative number) suggests that, on average, any pair of words in a topic is likely to occur together in less than 10% of the documents. A score that is smaller than -315 suggests that, on average, any pair of words in a topic is likely to occur together in less than 5% of the documents and the score is -480 for 1%. On this basis, the topics seem to have an adequate level of coherence. We could consider increasing or decreasing the number of topics to check if that increases mean topic coherence.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
1	select	great	good	offic	price
2	ink	sale	staff	product	help
3	onlin	suppli	fair	furnitur	mac
4	open	comput	cus	receiv	higher
5	printer	ask	town	peopl	store
6	assist	paper	choic	area	cartridg
7	pen	help	day	backtoschool	limit
8	checkout	window	select	shop	place
9	great	person	price	onlin	bought
10	mac	servic	dell	walk	happi
11	shop	laptop	reason	averag	comput
12	walmart	slow	locat	buy	rude
13	brother	backtoschool	exchang	extrem	equival
14	pleasant	place	polic	macintosh	canon
15	need	item	question	copier	knowabl

**Summary of top 15 words (ordered according to their importance) for the top 5 topics in the text corpus (also ordered according to their importance) in the text corpus (ignores custom stop-words).**

Mean topic coherence = -149.367

Coherence is a metric based on the co-occurrence of the top words (say top 15) within a topic in each document of the corpus. For each pair of words in a topic, we compute the log of the probability that a document containing the higher ranked word also contains at least one instance of the lower-ranked word. The overall coherence value for a topic is the sum of scores for each pair of words. A number close to 0 (the highest possible value of this metric) indicates high coherence. We report the average coherence value across the topics.

Mean topic exclusivity = 13.549

Exclusivity, measured using a metric called FREX, captures the extent to which the top words in a topic are exclusive to that topic (i.e., are not as likely to occur in the other topics). The exclusivity score for each top word in a topic is the harmonic mean of two equally-weighted components: (1) the rate at which the word occurs within a topic relative to its rate of occurrence in the other topics, and (2) the frequency (number of times) of a word's occurrence within a topic relative to its frequency in the other topics. The exclusivity score for a topic is the average of the exclusivity scores of the words in a topic. The computed score is a positive number ( $> 0$ ), with values substantially greater than 1 indicating a topic's high exclusivity. We report the average exclusivity value across the topics.

## Appendix A

### Co-occurrence graphs

In this analysis, we look at two words that co-occur together in the same document (e.g., a single tweet). We specify a sliding window of a given size to indicate how far apart the two words can be with respect to each other – are they just one-word apart (i.e., next to each other), within two-words apart, within three-words apart, etc. This separation context can be specified using a skip-gram value, which for example, is equal to 1 if we restrict the two words to occur next to each other (can occur before or after a focal word). In Enginius, we do two types of analyses, the first with a default value of 1 for the skip-gram, and the second with a “variable skip-gram” that allows co-occurrence anywhere within a sentence. In both cases, the program creates a symmetric co-occurrence matrix specifying how many times in a corpus each pair of words occur together within the specified context (skip-gram). This matrix is processed further to assess the probabilities of co-occurrence of each pair. The words are then plotted on a co-occurrence graph (the words that have higher probabilities of being connected are highlighted graphically with thicker shading). As a default, we plot up to 100 top words on the graph.

### Parts-of-speech tagging

POS tagging assigns a part-of-speech to every word in the text considering the context in which the word occurs. The tags include nouns (people, places, or things, including abstractions such as health or beauty), Verbs which refer to actions and processes, adjectives that specify properties of nouns, and adverbs that modify a verb, an adjective, or another adverb. There are, of course, some ambiguities – for example words like park or train could be a verb or a noun, but many of these ambiguities can be resolved by the context in which the word occurs. POS tagging of English words is pretty accurate, with most programs able to get over 97% accuracy.

Across a range of English texts, the percentage of nouns varies from 17% to 25%, verbs from 14 to 17%, prepositions from 9 to 14%, pronouns from 5 to 11%, adverbs 6 to 8% and adjectives 3 to 6% (see, for example, <http://infomotions.com/blog/2011/02/forays-into-parts-of-speech/>). Lincoln’s Gettysburg address had 18.8% verbs, 15.1% nouns, 10.7% pronouns, 11.4% adverbs, etc. Lincoln’s speech was a bit more “action-oriented” than the average text corpus.

### RAKE (Rapid Automatic Keyword Extraction)

RAKE is based on the observation that keywords (a set of contiguous words) rarely contain punctuation or stop words. Thus, a candidate keyword is any set of contiguous words (i.e., 2-gram or bigram for two contiguous words, 3-gram or trigram for three contiguous words, and more generally n-gram) that doesn't contain phrase delimiters or stop words, i.e., candidate keywords are content-bearing words. Example bigrams are “customer service,” “merry christmas,” and “thank you.” Example trigrams are “new york times,” “miles per hour,” and “my credit card.”

In this sense, RAKE looks for specific sets of words within each document that characterize (i.e., gives meaning) to a document, and thereby provides a more nuanced interpretation of the key contents of a set of documents than is provided by word co-occurrence graphs that only consider pairs of words. The input parameters for RAKE are a list of stop words, a set of phrase delimiters (e.g., a comma or period signifying the beginning or end of a phrase), a set of word delimiters, and the n-gram level for extraction.

Once all possible candidate n-grams in a set of documents are identified, RAKE selects the most relevant ones that characterize the entire corpus. To do this, it first computes a RAKE score for each word, which is equal to its  $\frac{\text{degree}}{\text{frequency}}$ , where degree is the number of times that word co-occurs with another word in another keyword, and frequency is the total number of times that the word occurs overall in the corpus. This ratio favors words that predominantly occur in longer keywords, whereas degree of a word favors those words that occur often and in longer candidate keywords, and frequency favors words that occur frequently regardless of the number of words with which they co-occur. The RAKE score for each keyword is the sum of the RAKE scores of all the words in that keyword. Because of this summation, RAKE scores typically tend to be higher for longer sequences of words. All candidate keywords are rank-ordered from the highest to lowest on their RAKE scores. In Enginius, we set the minimum frequency for words to be included in the rank-ordered list to be at 0.01% of the total word count, and we list up to 20 keywords that contain at most four contiguous words each (4-gram words).

For more information about RAKE, please see Rose, Stuart, Dave Engle, Nick Cramer and Wendy Cowley (2010), "Automatic keyword extraction from individual documents," in Text Mining: Applications and Theory, edited by Michael W. Berry and Jacob Kogan, John Wiley & Sons. Ltd.

## Topic Model

Topic modeling is based on an underlying probability model of a stylized data generating process that helped create the latent (hidden) topics in a set of documents. The stylized data generating process is as follows: (1) For each document, the author selects topics from an unknown (Dirichlet) probability distribution of topics, where the number of topics is fixed beforehand. (2) Likewise, the author selects each word for a given topic using an unknown probability (Dirichlet) distribution over a fixed number of words. A Bayesian model estimates the parameters of the unknown probability distributions such that the estimated probability distributions recover as closely as possible the known set of words in each document.

Here is an example of the stylized process of how two documents focused on two topics, sports and fashion, are crafted by their writers. Let's say the first document is mostly about sports but with a few components related to fashion, and let's say the second document is mostly about fashion but with some discussion of sports. For both documents, the writers choose a combination of words that characterize these two topics, namely, when some words appear together, they convey something about sports, and other words when they appear together convey something about fashion. When there are many documents in a corpus, each document will contain a set of topics characterized by various word combinations. Even though each document may contain only a few topics, a text corpus may contain dozens of topics. Topic modeling is designed to leverage this stylized notion of how documents are generated, to identify a few topics and the relatively unique combination of words that pertain to each topic.

To compute mean topic coherence, Enginius uses the following metric called Intrinsic UMass measure based on the empirical conditional log-probability,  $\log p(w_i|w_j)$ , where  $i$  is a higher-ranked word in a topic than word  $j$ . The empirical score for a pair of words in the ordered list of words in the topic is then equal to:

$$UMass(w_i, w_j) = \text{Log} \frac{D(w_i, w_j) + 1}{D(w_i)}$$

$D$  is a counting function to count the number of documents in which a word occurs. The numerator is a count of the number of documents that contain both words, and the denominator is the count of the number of times the

higher-ranked word occurs. The extra 1, a smoothing factor, is to allow for situations where  $D(w_i, w_j) = 0$ . The topic coherence score is then specified by summing up scores for all pairs of words in the ordered list of the top 15 words in a topic. With 15 words in a topic, and with assuming 10% average overlap of the words across documents, the benchmark value for topic coherence is -241.8, and with 5% average overlap, the benchmark value is -314.6. A computed value of topic coherence that is higher than these values (i.e., closer to 0), could be considered to have good coherence.

$$\sum_{i < j} UMass(w_i, w_j)$$

To compute mean topic exclusivity, we use the FREX (Frequency, Exclusivity) measure proposed by Bischof and Airoldi (2012). This metric captures the extent to which the top words in a topic are exclusive to that topic (i.e., are not as likely to occur in the other topics). The exclusivity score for each top word in a topic is the harmonic mean of two components for which we assign equal weights as default: (1) the rate at which the word occurs within a topic relative to its rate of occurrence in the other topics, and (2) the frequency (number of times) of a word's occurrence within a topic relative to its frequency in the other topics. The exclusivity score for a topic is the average of the exclusivity scores of the words in a topic. By default, Enginius considers the top 15 words in a topic. The computed score is a positive number ( $> 0$ ), with values substantially greater than 1 indicating a topic's high exclusivity. We report the average exclusivity value across the topics.

Bischof, Jonathan M. and Edoardo M. Airoldi (2012), "Summarizing topical content with word frequency and exclusivity." In International Conference on Machine Learning, volume 29, Edinburgh, Scotland, UK.

#### R – Packages and routines used in Enginius

- tm – text manipulation
- stringr – remove graphical characters
- SnowballC – stemming of words
- Wordcloud – Plot word cloud
- RCurl – html download
- XML – html text extraction
- scales – pie chart modifications
- syuzhet – emotion selection
- curl, httr, ROath, httpuv, rtweet – tweet extraction
- topicmodels, textmineR, text2vec – LDA model (topic model)
- textclean, spacy – (removing emogis and replacing with text)
- udpipe – POS, Co-occurrence, and RAKE analysis

## Appendix B

### Stop Words – Short List

a	have	other	we
about	haven't	ought	we'd
above	having	our	we'll
after	he	ours	we're
again	he'd	ourselves	we've
against	he'll	out	were
all	he's	over	weren't
am	her	own	what
an	here	same	what's
and	here's	shan't	when
any	hers	she	when's
are	herself	she'd	where
aren't	him	she'll	where's
as	himself	she's	which
at	his	should	while
be	how	shouldn't	who
because	how's	so	who's
been	i	some	whom
before	i'd	such	why
being	i'll	than	why's
below	i'm	that	with
between	i've	that's	won't
both	if	the	would
but	in	their	wouldn't
by	into	theirs	you
can't	is	them	you'd
cannot	isn't	themselves	you'll
could	it	then	you're
couldn't	it's	there	you've
did	its	there's	your
didn't	itself	these	yours
do	let's	they	yourself
does	me	they'd	yourselves
doesn't	more	they'll	
doing	most	they're	
don't	mustn't	they've	
down	my	this	
during	myself	those	
each	no	through	
few	nor	to	
for	not	too	
from	of	under	
further	off	until	
had	on	up	
hadn't	once	very	
has	only	was	
hasn't	or	wasn't	

## Stop Words – Long List

able	as	co	etc	happens
about	a's	co.	even	hardly
above	aside	com	ever	has
abroad	ask	come	evermore	hasn't
according	asking	comes	every	have
accordingly	associated	concerning	everybody	haven't
across	at	consequently	everyone	having
actually	available	consider	everything	he
adj	away	considering	everywhere	he'd
after	awfully	contain	ex	he'll
afterwards	back	containing	exactly	hello
again	backward	contains	example	help
against	backwards	corresponding	except	hence
ago	be	could	fairly	her
ahead	became	couldn't	far	here
ain't	because	course	farther	hereafter
all	become	c's	few	hereby
allow	becomes	currently	fewer	herein
allows	becoming	dare	fifth	here's
almost	been	daren't	first	hereupon
alone	before	definitely	five	hers
along	beforehand	described	followed	herself
alongside	begin	despite	following	he's
already	behind	did	follows	hi
also	being	didn't	for	him
although	believe	different	forever	himself
always	below	directly	former	his
am	beside	do	formerly	hither
amid	besides	does	forth	hopefully
amidst	best	doesn't	forward	how
among	better	doing	found	howbeit
amongst	between	done	four	however
an	beyond	don't	from	hundred
and	both	down	further	i'd
another	brief	downwards	furthermore	ie
any	but	during	get	if
anybody	by	each	gets	ignored
anyhow	came	edu	getting	i'll
anyone	can	eg	given	i'm
anything	cannot	eight	gives	immediate
anyway	cant	eighty	go	in
anyways	can't	either	goes	inasmuch
anywhere	caption	else	going	inc
apart	cause	elsewhere	gone	inc.
appear	causes	end	got	indeed
appreciate	certain	ending	gotten	indicate
appropriate	certainly	enough	greetings	indicated
are	changes	entirely	had	indicates
aren't	clearly	especially	hadn't	inner
around	c'mon	et	half	inside

insofar	meanwhile	off	regards	specifying
instead	merely	often	relatively	still
into	might	oh	respectively	sub
inward	mightn't	ok	right	such
is	mine	okay	round	sup
isn't	minus	old	said	sure
it	miss	on	same	take
it'd	more	once	saw	taken
it'll	moreover	one	say	taking
its	most	ones	saying	tell
it's	mostly	one's	says	tends
itself	mr	only	second	th
i've	mrs	onto	secondly	than
just	much	opposite	see	thank
k	must	or	seeing	thanks
keep	mustn't	other	seem	thanx
keeps	my	others	seemed	that
kept	myself	otherwise	seeming	that'll
know	name	ought	seems	thats
known	namely	oughtn't	seen	that's
knows	nd	our	self	that've
last	near	ours	selves	the
lately	nearly	ourselves	sensible	their
later	necessary	out	sent	theirs
latter	need	outside	serious	them
latterly	needn't	over	seriously	themselves
least	needs	overall	seven	then
less	neither	own	several	thence
lest	never	particular	shall	there
let	neverf	particularly	shan't	thereafter
let's	neverless	past	she	thereby
like	nevertheless	per	she'd	there'd
liked	new	perhaps	she'll	therefore
likely	next	placed	she's	therein
likewise	nine	please	should	there'll
little	ninety	plus	shouldn't	there're
look	no	possible	since	theres
looking	nobody	presumably	six	there's
looks	non	probably	so	thereupon
low	none	provided	some	there've
lower	nonetheless	provides	somebody	these
ltd	noone	que	someday	they
made	no-one	quite	somehow	they'd
mainly	nor	qv	someone	they'll
make	normally	rather	something	they're
makes	not	rd	sometime	they've
many	nothing	re	sometimes	thing
may	notwithstanding	really	somewhat	things
maybe	novel	reasonably	somewhere	think
mayn't	now	recent	soon	third
me	nowhere	recently	sorry	thirty
mean	obviously	regarding	specified	this
meantime	of	regardless	specify	thorough

thoroughly	was	would	describe	beginnings
those	wasn't	wouldn't	detail	begins
though	way	yes	due	biol
three	we	yet	eleven	briefly
through	we'd	you	empty	ca
throughout	welcome	you'd	fifteen	date
thru	well	you'll	fifty	ed
thus	we'll	your	fill	effect
till	went	you're	find	et-al
to	were	yours	fire	ff
together	we're	yourself	forty	fix
too	weren't	yourselves	front	gave
took	we've	you've	full	giving
toward	what	zero	give	heres
towards	whatever	a	hasnt	hes
tried	what'll	how's	herse	hid
tries	what's	i	himse	home
truly	what've	when's	interest	id
try	when	why's	itse"	im
trying	whence	b	mill	immediately
t's	whenever	c	move	importance
twice	where	d	myse"	important
two	whereafter	e	part	index
un	whereas	f	put	information
under	whereby	g	show	invention
underneath	wherein	h	side	itd
undoing	where's	j	sincere	keys
unfortunately	whereupon	l	sixty	kg
unless	wherever	m	system	km
unlike	whether	n	ten	largely
unlikely	which	o	thick	lets
until	whichever	p	thin	line
unto	while	q	top	'll
up	whilst	r	twelve	means
upon	whither	s	twenty	mg
upwards	who	t	abst	million
us	who'd	u	accordance	ml
use	whoever	uucp	act	mug
used	whole	w	added	na
useful	who'll	x	adopted	nay
uses	whom	y	affected	necessarily
using	whomever	z	affecting	nos
usually	who's	l	affects	noted
v	whose	www	ah	obtain
value	why	amount	announce	obtained
various	will	bill	anymore	omitted
versus	willing	bottom	apparently	ord
very	wish	call	approximately	owing
via	with	computer	aren	page
viz	within	con	arent	pages
vs	without	couldnt	arise	poorly
want	wonder	cry	auth	possibly
wants	won't	de	beginning	potentially

pp	ts
predominantly	ups
present	usefully
previously	usefulness
primarily	've
promptly	vol
proud	vols
quickly	wed
ran	whats
readily	wheres
ref	whim
refs	whod
related	whos
research	widely
resulted	words
resulting	world
results	youd
run	youre
sec	
section	
shed	
shes	
showed	
shown	
shows	
shows	
significant	
significantly	
similar	
similarly	
slightly	
somethan	
specifically	
state	
states	
stop	
strongly	
substantially	
successfully	
sufficiently	
suggest	
thered	
thereof	
therere	
thereto	
theyd	
theyre	
thou	
thoughh	
thousand	
throug	
til	
tip	