

Application Programming Interfaces (API) for Web-Scraping & Text Analysis

Jeff Sternberg
Research Methods
Alexandra Alden
Spring 2020



Northeastern University
NULab for Texts, Maps, and Networks

Workshop Agenda

- Learn about Web-Scraping & APIs for Data Collection
- Introduce and Explore the New York Times API
- Collect “Global Warming” News Coverage
- Conduct a Text Analysis of Global Warming News Coverage in Lexos

Slides, handouts, and data available at

<http://bit.ly/diti-spring2020-alde-textanalysis>

Learning Objectives

- Understand the definition and purpose of an API and web-scraping
- Understand the importance of API documentation
- Understand the affordances and limitations of using APIs to build a corpus
- Start to understand how to use digital tools to pull out novel insights and findings from text data

Words of Support and Advice

Take this lesson at your own pace. You are not expected to understand most of the code in the Jupyter Notebooks, instead pay attention to the output it produces, the type of data and information you can get from the New York Times, and what research questions you could ask using the New York Times API of you could control what Keyword and time period we are searching for articles from.

The same goes with Lexos. You will most likely have a few technical issues using it, and being that I am not in the classroom to deal with them, take note of them and power on. I will be holding open office hours soon to deal with these more directly and answer any questions you have as that came up working through the module about APIs in general, Python, Lexos, Text Analysis, etc.

Discussion Post

Write a paragraph Discussion Post on Blackboard Responding to these Questions:

If you could collect any kind of data you want from the internet, basically any type of information available digitally regardless of access, what would it be?

- What topic would you be interested in?
- What would your research question be?
- What sources would you like to use? Would you be dealing with business, government or institutional data? Would it be social media data (as in posts, text, images, videos)?
- What limitations to getting your dream data could you foresee?

Getting your Dream Data is Possible ...through APIs

A lot of data that you might be looking for is available online through APIs or Application Programming Interfaces. A lot of companies, governments, and institutions collect data, or track users utilizing their websites and platforms, and put all of this information onto a server where it can be accessed later: this is an API.

Many businesses maintain APIs for App developers and businesses to use, and sell access to their APIs for these purposes, we can think of Google Maps, Yelp, AirBnB, and other businesses who would like App developers to use their data to increase traffic to their sites.

Though APIs are mostly used and maintained for these business purposes, they can also be used for research! We are going to go over Webscraping this research data using APIs.

What is an API and what is web-scraping?

An API, or application programming interface, is a set of subroutine definitions, communication protocols, and tools for building software that ultimately allows applications to communicate with one another. An API may be for a web-based system, operating system, database system, computer hardware, or software library.

Web-scraping is the process of extracting large amounts of data from an internet source and downloading the data to a local repository. The scraping process can be done manually, but is usually automated by using software because of the large amount of data typically involved.

API Documentation

- When using APIs for web-scraping, it is necessary to refer to the API documentation and a link is usually found on the API homepage.
- Why?
 - While the concepts remain roughly the same, APIs differ and the syntax for accessing data can be very different.
 - You will likely need an API key, and the links for registering for the key will be found in the documentation.
 - There may be other unaccounted for differences and API specifics that require a close understanding of the API's structure.

API Documentation is essentially the code and instruction for working with APIs that tells you both what type of information is available through a given API, but also how to call different types of information, the limits on how many searches and web pulls you can do on the API at a time, etc. Before working with an API, read through the documentation carefully, as this will help you figure out if the information you are looking for is attainable here, as well as help you shape your research questions that this API can answer based both off of your interests as well as on what data is available. It is this sweet spot that is emblematic of Research which utilizes digital methods and data scraped from APIs and the web.

Popular APIs

- New York Times: <https://developer.nytimes.com/>
- Reddit: <https://www.reddit.com/dev/api/>
- IMDB: <http://www.omdbapi.com/>
- FBI: <https://crime-data-explorer.fr.cloud.gov/api>
 - Other Federal government APIs:
<https://api.data.gov/docs/>
- Twitter: <https://developer.twitter.com/en/docs.html>

Here is a list of popular APIs. Click on the links and look at the different types of searches they allow you to do, the types of data and information you can get from each, and think about how what data these APIs make available can produce certain research questions. Practice coming up with research questions as you peruse these different APIs and their documentations.

New York Times API

- The New York Times has many different active [APIs](#) providing access to a variety of different text data sources, holding Articles, Movie Reviews, Book Reviews, User Comments, etc
- For our purposes, investigating news article coverage of the “Global Warming”, we will be utilizing the [New York Times Article Search API](#)

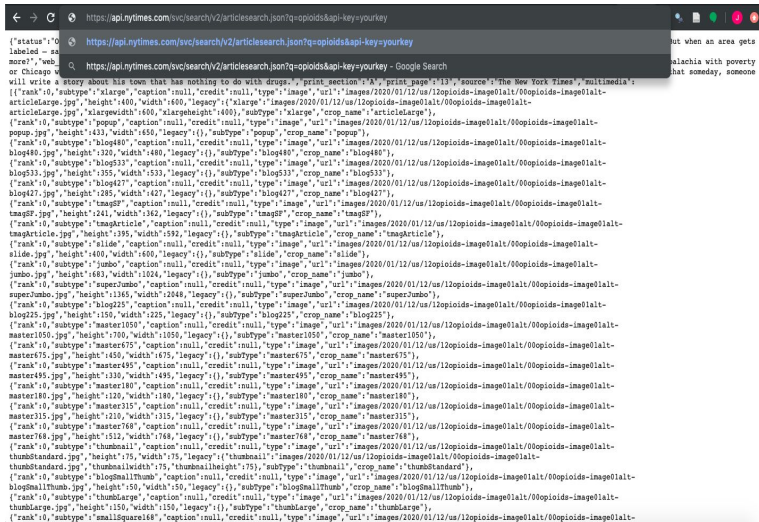
For this lesson, we are going to be using the New York Times Article API. There are a few different APIs that the New York Time has available, click around to see what types of searches they allow you to do.

We are going to use the NY Times Article API, which will allow us to search the NY Times database for all articles produced in a given time period relating to a keyword of our choice. We are going to be searching for articles by the keyword of “global warming” for 2010 to 2019, to see how coverage of global warming has changed over time.

We will be getting both article counts relating to global warming for each year, as well as more text information about each article including it's author, title, date of publication, tags used to categorize it, and bits of text from the actual articles themselves including a snippet, abstract, and lead paragraph from the article itself.

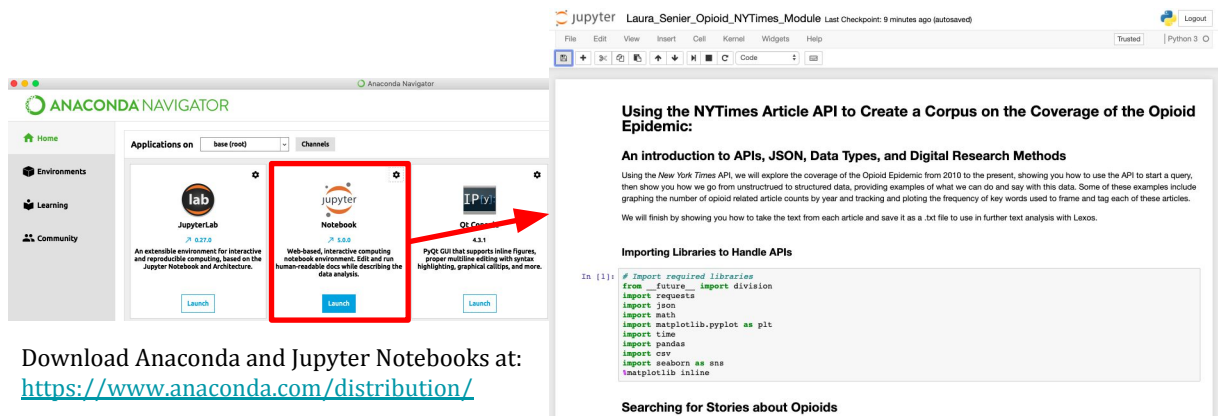
I will show you how we can use this data to ask and answer different research questions, as well as collect the text from different articles to do a computational text analysis on using Lexos, a open source text analysis platform.

Querying Using URL and Browser



- We access the API through our web-browser, giving it the API URL with our query as q="global warming" and our API key after that.
- This returns us a mess of a json file presented in html in our browser.
- **What do we do with this? How do we make it useable?**

The Answer? Parsing using Python and Jupyter Notebooks!



The image shows two overlapping screenshots. On the left is the Anaconda Navigator application window, displaying a sidebar with 'Home', 'Environments', 'Learning', and 'Community'. The main area shows 'Applications on base (root)' with three options: 'JupyterLab', 'Jupyter Notebook' (highlighted with a red box), and 'Qt Console'. A red arrow points from the 'Jupyter Notebook' box to the right. On the right is a Jupyter Notebook interface titled 'Laura_Senier_Opioid_NYTimes_Module'. The notebook content includes a title 'Using the NYTimes Article API to Create a Corpus on the Coverage of the Opioid Epidemic:', an introduction to APIs, JSON, Data Types, and Digital Research Methods, and a code cell titled 'Importing Libraries to Handle APIs' containing the following Python code:

```
In [1]: # Import required libraries
from future import division
import requests
import json
import math
import matplotlib.pyplot as plt
import time
import pandas
import csv
import seaborn as sns
matplotlib inline
```

Below the code cell, the text 'Searching for Stories about Opioids' is visible.

Download Anaconda and Jupyter Notebooks at:
<https://www.anaconda.com/distribution/>

Link to the [Jupyter Notebook](#), follow along!

Click on the Jupyter Notebook link, read through and follow along. Don't worry about understanding the code, pay more attention to the comments around the code that tell you what it is doing, as well as what the code is outputting. Look at what we get originally from the base search, a bunch of unstructured and messy data, and how we go about turning that into structured and useable information, and the different types of analyses and visualizations we can perform using this data.

If you know any Python, you can install anaconda and run the Jupyter notebook yourself, and change parameters, but it is not at all expected you should do this, or really that anybody would be able to. I am more signalling here you can learn to use python and jupyter notebooks, and once you do, you can use these particular notebook to use the NY Times API, changing around search terms and date ranges to do your own article searches.

If anyone is interested in learning python, jupyter notebooks, and how to use APIs and do these analyses yourself, sign up for my Summer 2 Course INSH 1500: Digital Methods for Social Science, where we will learn to do this and more (including some Mapping & GIS, a bit of Network Analysis, Image Analysis, and go much more in depth on Computational Text Analysis using Python, teaching you new research methods to further social science inquiries.

Follow the Link to the [Jupyter Notebook](#)

- Scroll and read through the Jupyter Notebook, reading each section and attempting to understand how we go from unstructured API data, to structured data and findings
- You will not understand most of the code, and you aren't meant to, you should be mainly looking at what kinds of data the API provides and how we go from data to findings, to more research questions
- After you are done, return to this presentation and go to the next slide

What do we do with these Articles?

Now that we have collected 100 sample articles on Global Warming from 2010-2019 with some text snippets, what do we do with it?

Well, we could read the text of each of these articles qualitatively and begin making arguments about how the coverage has changed over time based on our sample. But there's 100 articles to read and could be time consuming. How could we do this faster or at scale?

This is where computational text analysis comes in! Computational text analysis uses computers to read a lot of documents, having them count frequent terms and use these to compare and say something about a bunch of documents quickly. This is distant reading and something that can be very useful if you have a lot of texts to look at that simply reading through doesn't allow. It can also tell us different things than a qualitative content analysis would.

I'm going to show you one tool for this, Lexos: an open source text analysis platform, to show you what type of findings we can get from text analysis, as well as give you a tool to use on texts of your own interest past this lesson.

What is Computational Text Analysis?

Computational text analysis refers to an array of methods that can be used to “read” texts with a computer. This form of analysis can range from basic word frequency counts to more advanced techniques like machine learning.

Text analysis is often used on a **corpus**, or a collection of multiple texts, and provides a glimpse into patterns across the texts. Some people also perform text analysis on larger individual documents, like novels.

Computers are *really really really* good at counting, much better than people are. They can parse through a mass amount of data very quickly, do basic and more advanced calculations, and provide results for us then to interpret. For example, word count frequencies are a basic way to just count how often a word shows up in a text or corpus. Corpus, btw, is a collection of texts you are using to do research. Word embedding models, which is a more advanced form of text analysis, can measure the relationship between words and how similarly words are used together.

Why Computational Text Analysis?

Computational text analysis can help us analyze a **ton** of data and discover **patterns** in texts.

Particular disciplines care **deeply** about the language used and how this language may reach intended audiences. Text analysis provides another method for approaching these discourses.

It's an important method because

- a) it can work with a ton of data, like millions upon millions of words or texts – although the web-browser tools we will be using cannot work with that much data. You'll have to learn programming languages like Python or R or purchase a license to a more advanced software.
- b) show patterns in the text that we might have not caught in the first place. For example, when you're reading a novel, you might be paying close attention and analyzing as you're reading, but maybe you do not notice how often the author uses particular words over and over. CTA results can show you the most often words used by the author, which can provide another perspective for understanding text.

BIG DATA

This might sound familiar! Sort of similar to what we talked about with **big data**. However, big data is usually done in real time & has algorithms that are constantly adjusting themselves to be correct. CTA is usually done by collecting data first and then analyzing (although it can be done in real time). In fact, I bet there are CTA methods used in big data.

POLITICS

When thinking about politics, politicians are *known* for their expertise in choosing language that reaches their intended audience. CTA provides another method to understand politicians' rhetorical choices.

Notes on Creating a Corpus (in General)

1. Choose the texts you want to include in your corpus
2. Create a folder on your computer titled “corpus” or something even more specific
3. Copy and paste your texts into a **plain text editor** (on Macs: Text Edit; on Windows: Notepad)
 - a. Mac users, you will need to make your Text Edit into a plain text editor. Open Text Edit, go to Preferences, and make sure “plain text” is selected
4. Save each text as a different plain text file (with a .txt extension). Name your files so you know what is in them!

We have the NY Times Global Warming articles as our corpus for this lesson, but how would you go about making your own corpus to analyze using the tools showed to you in this lesson?

We have provided general corpus-creating instructions on the handout. These are pretty much the exact steps we followed to create our corpus, and ones that we hope you can follow too when you are building your own corpora!

Our corpus is a collection of plain text files. Each file is a different news story. These were sent to you via email, and are also available for download at the [bit.ly](#) link under the “Data” folder.

Lexos

Lexos is an open source text analysis platform hosted at Wheaton College that allows you to upload, manage, clean, analyze and visualize texts and corpora all in your browser. It is easy to use and shows you how text analysis works, step by step. It should be easy to use, and doesn't require any Python, R, or computer coding knowledge. This platform shows you what is possible in terms of computational text analysis, and is meant to give you a tool you can use in the future, as well as pique your interest to this methodology and let you know you can learn to do this stuff directly if you pick up a bit of Python or R.

Lexos: <http://lexos.wheatoncollege.edu/upload>

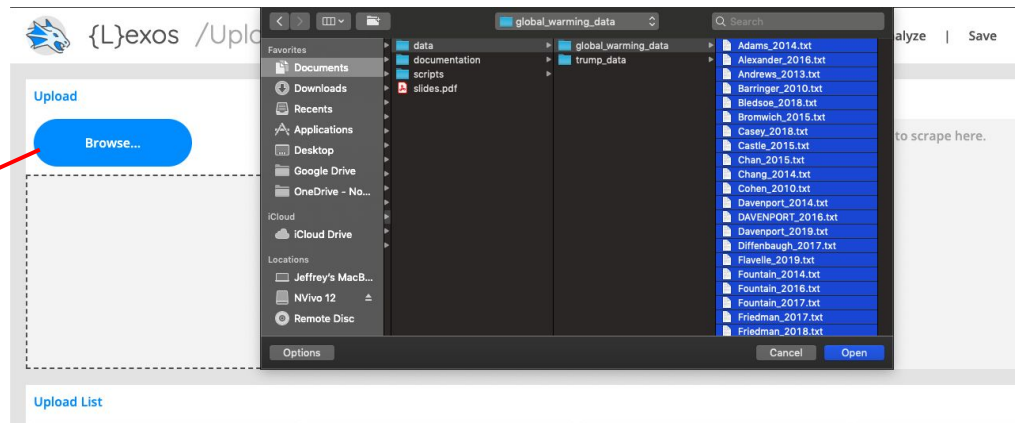
Lexos provides a step-by-step guide for corpus uploading, preparation, and analysis.

- **Upload:** upload your corpus (your separate .txt files)
- **Manage:** select the files you want to prepare and analyze
- **Prepare:** prepare your corpus for analysis
- **Visualize:** create visualizations of patterns across your corpus or in single texts
- **Analyze:** analyze your corpus, including comparing texts

Go to the Lexos link and we will move through each of these steps together using the NY Times Global Warming Article Data

Lexos: Upload

Click Browse
and select your
entire corpus
(or drag and
drop)



Take the global warming articles sent to you in the email, or download them at this [github link](#), and upload them to Lexos

Lexos: Manage



Upload **Manage** Prepare Visualize Analyze | Save Reset | Help

Make sure all the documents in the corpus you want to use are selected (blue = selected, gray = not selected)

Active	#	Document	Class	Source	Excerpt	Download ?
<input checked="" type="checkbox"/>	1	Adams_2014		Adams_2014.txt	When dealing with China on climate change, the motto must be praise but verify — and keep pushing for more.HONG KONG — Whether... ..uth lies in between.When dealing with China on climate change, the motto must be praise but verify — and keep pushing for more.	
<input checked="" type="checkbox"/>	2	Alexander_2016		Alexander_2016.txt	America must expand its clean-energy plans to make room for new, safer reactors.If 20 fire marshals came around and told us our... ..er plants that produce carbon-free electricity.America must expand its clean-energy plans to make room for new, safer reactors.	
<input type="checkbox"/>	3	Andrews_2013		Andrews_2013.txt	In a new book, William Nordhaus of Yale provides a lucid review of the climate-change problem from both an economic and a meteo... ..ordhaus of Yale provides a lucid review of the climate-change problem from both an economic and a meteorological point of view.	

Once you upload the files, click the “Manage” Tab. It will take you to the manage page, where you can select which of the articles you want to include in the analysis. For our example, make sure they are all chosen.

Lexos: Prepare (scrub)

Lexos demonstrates the different options you have for preparing your corpus. By “scrubbing,” you are transforming the texts in your corpus and making choices that will impact your results. Here are some possibilities:

- **Make Lowercase:** make all your letters lowercase. Even though you know “A” and “a” are the same letter, the computer treats these as two separate characters. Lowercasing removes this distinction.
- **Remove Punctuation:** remove punctuation, which may influence your results.
- **Stop/Keep Words:** remove a list of words. Usually these would be **stopwords**, or the most common words in a language (English: the, a, she, her, it, him, they, etc).
- **Lemmas:** standardize to the *stem* of word. For example, you can stem all forms of talk: talking, talked, talks, etc. to “talk”

Lexos: Removing Stopwords

Get a list of English stopwords here: <https://gist.github.com/sebleier/554280> (we also sent you a .txt file). Copy and paste the stopwords (or upload the .txt file) into the “Stop/Keep Words” box then select “Stop”

Under the Scrubbing section, make sure to check the “Make Lowercase”, “Remove Digits” and “Remove Punctuation” boxes.

Play around with checking the other boxes in the Scrubbing Options Section and applying them to our texts to see how they change the output.

The screenshot shows the Lexos web interface. At the top, there is a navigation bar with links: Upload, Manage, Prepare, Visualize, Analyze, Save, Reset, and Help. The main content area is divided into several sections. On the left, the 'Scrubbing Options' section is highlighted with a red box. It contains a grid of checkboxes: 'Make Lowercase' (checked), 'Remove Digits' (checked), 'Remove Punctuation' (checked), 'Remove Spaces' (unchecked), 'Remove Tabs' (unchecked), 'Remove Newlines' (unchecked), 'Scrub Tags' (unchecked), 'Remove Hyphens' (unchecked), 'Keep Apostrophes' (unchecked), and 'Keep Ampersands' (unchecked). Below this are sections for 'Lemmas' and 'Consolidations', each with an 'Upload' button. To the right of the 'Scrubbing Options' section is the 'Stop/Keep Words' section. It has a 'Remove Upload' button and a red box around the 'Upload: NLTK's list of english stopwords' text. Below this, there are radio buttons for 'Off', 'Stop' (selected), and 'Keep'. Further right is the 'Previews' section, which shows a list of documents with titles like 'Adams_2014', 'Alexander_2016', and 'Andrews_2013'. Each document has a 'Preview' button and a 'Download' button. The 'Preview' button for 'Adams_2014' is highlighted with a red box. The 'Preview' button for 'Alexander_2016' is also highlighted with a red box. The 'Preview' button for 'Andrews_2013' is highlighted with a red box.

Lexos: Applying your Preparations

Once you have made decisions about your preparations, click “**Apply**” and wait a few minutes. Because the program is going through each document and completing all the processes you selected, it needs some time. Then, you will see the final results of your preparation! You can also **download** your new corpus.

BEFORE PREP

Previews Preview **Apply** Download

[Adams_2014](#)
When dealing with China on climate change, the motto must be praise but verify — and keep pushing for more.HONG KONG — Whether... ..uth lies in between.When dealing with China on climate change, the motto must be praise but verify — and keep pushing for more.

[Alexander_2016](#)
America must expand its clean-energy plans to make room for new, safer reactors.If 20 fire marshals came around and told us our... ..er plants that produce carbon-free electricity.America must expand its clean-energy plans to make room for new, safer reactors.

[Andrews_2013](#)
In a new book, William Nordhaus of Yale provides a lucid review of the climate-change problem from both an economic and a meteo... ..ordhaus of Yale provides a lucid review of the climate-change problem from both an economic and a meteorological point of view.

AFTER PREP

Previews Preview **Apply** Download

[Adams_2014](#)
dealing china climate change motto must praise verify keep pushing morehong kong whether china climate hero climate villain ma... ..na foremost investor renewable energy truth lies betweenwhen dealing china climate change motto must praise verify keep pushing

[Alexander_2016](#)
america must expand cleanenergy plans make room new safer reactorsif fire marshals came around told us houses burn wed buy fire... ..operating nuclear power plants produce carbonfree electricityamerica must expand cleanenergy plans make room new safer reactors

[Andrews_2013](#)
new book william nordhaus yale provides lucid review climatechange problem economic meteorological point viewiäve long looked... ..truly understood new book william nordhaus yale provides lucid review climatechange problem economic meteorological point view

Analysis and Visualization

In the next section, we will explore and try out some of Lexos' different visualization and analysis features, to give you an idea of the different insights we can gain from computational text analysis, and how this speaks to different research questions we could ask of our data.

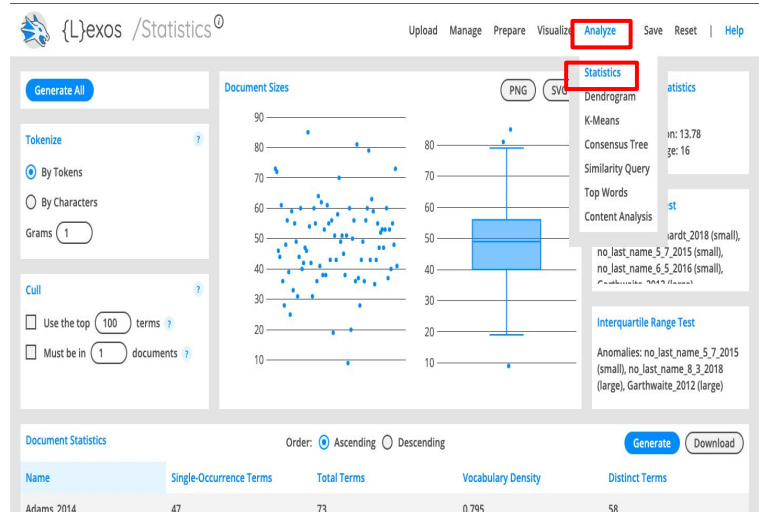
Please follow along, and perform the same steps on the data yourself slide by slide.

Lexos: Statistics

Click on the “Analyze” Tab to open the drop down menu, then click on “Statistics”.

This will take you to the Statistics Page where you can get descriptive statistics and information on all of our Global warming Articles. You can see how long each of the articles are, related to each other through the scatter plot and box and whiskers plot, the number of unique terms per document, the average number of terms for the corpus as a whole, a list of outliers, etc.

Take a look and see what kinds of information it gives you!



Lexos: Visualize



Word Cloud:
visualize a
wordcloud across
the entire corpus.

Multi Cloud: visualize wordclouds for each individual document/text



Click on the Visualize Tab and go down to World Cloud. This will produce a wordcloud of the whole corpus of Global warming related NY Times Articles, telling us which words appear the most in the corpus. We can play around with the “Term Count” parameter which will change which Words appear in our cloud.

If you go to the visualize tab again and select Multi Cloud, Lexos will generate two individual document word clouds for the first two articles in our corpus. Multi Cloud can only generate these document level word clouds for two documents at a time, for comparing the documents, so it chooses the first two in our corpus. If you want to change which documents are being used to generate these multi clouds, go back to the Manage Tab, and deselect all the articles and choose the two you are most interested in.

Play around with the Term Count Parameter, and select different articles to compare!

Lexos: Top Words

Lexos Top Words

Upload Manage Prepare Visualize **Analyze** | Save Reset | Help

Comparison Method

- ☒ Each Document to the Corpus
- ☐ Each Document to Other Classes
- ☐ Each Class to Other Classes

Tokenize

- ☒ By Tokens
- ☐ By Characters

Grams:

Cull

- ☐ Use the top
- ☐ Must be in

Statistics

- Dendrogram
- K-Means
- Consensus Tree
- Similarity Query
- Top Words**

Top Words

Generate Download

Class Divis

Document "Adams_2014" Compared To The Corpus	Document "Alexander_2016" Compared To The Corpus	Document "Andrews_2013" Compared To The Corpus
china 8.7	cleanenergy 7.3608	book 9.2694
motto 7.3084	expand 7.3608	climatechange 9.2694
praise 7.3084	room 7.3608	economic 9.2694
pushing 7.3084	safer 7.3608	lucid 9.2694
verify 7.3084	us 7.3608	meteorological 9.2694
dealing 5.3741	make 5.9025	nordhaus 9.2694
Document "Barringer_2010" Compared To The Corpus	Document "Bledsoe_2018" Compared To The Corpus	Document "Bromwich_2015" Compared To The Corpus
massachusetts 9.4825	warned 8.0192	kind 10.5043
puts 9.4825	greenhouse 7.4576	living 10.5043
taken 9.4825	scientist 7.4576	activist 8.5017

Go to the “Analyze” Tab and select “Top Words”. This will take you to the “Top Words” page where Lexos will generate information on which words both have the highest word count in each document, but also compare these top words in each document to the top words in the whole corpus, selecting out which words are most unique to each document and set the article apart from others.

You can also play around with how Lexos compares each document, as well as how it counts words in each document. Just make sure to make these changes then hit the “Generate” button to apply them to the corpus and see the output.

You can change how these top words are compared by changing your selection in the “Comparison Method Section”, though we have not “classed” or categorized our documents for comparison on these other fronts. How could we go about categorizing our documents into group that would make these comparisons useful?

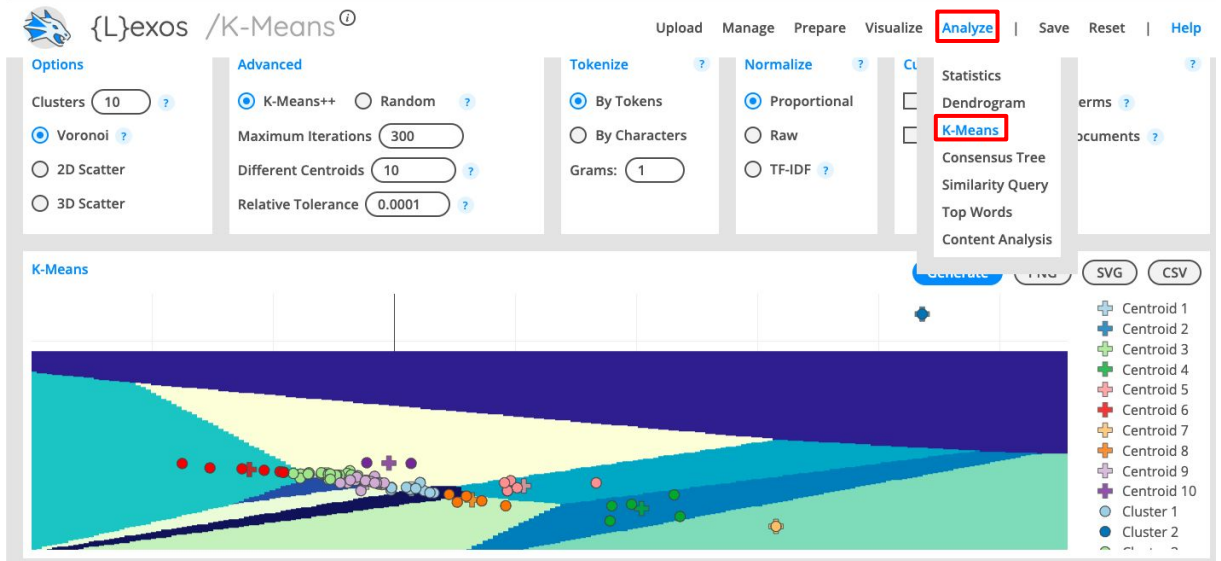
We can also go to the tokenize section and choose how we are counting words. The default is to use tokens, which takes all forms of a word like “talking”, “talked”, and “talk” and reduces them all to the stem word “talk” in order to count them as a whole. You can select to tokenize “By characters” which will tell you the top letters in each document, if that is helpful to you for some type of research question based around composition. The “Grams” field is set at 1 by default, which means it is only counting single terms/words. If you change this to 2, it will count pairings of words, so every time two words appear together, and give you the top pairings. You can increase this number as much as you want, with 3 giving you clusters of three words used together,

etc. This can help you look for turns of phrase, or larger concepts, or the way that different subjects are being described, seeing how adjectives and nouns are paired, etc.

You can also play around with the “Cull” section where you can select how many of the top words to use and see for comparison. The default is the Top 100 Words which would show you how the Top 100 words in the corpus frequency wise are distributed in each document. Changing the number changes the number of words used for comparison. You can also enable the “Must be in X Documents” option and narrow down the top words used for comparison to words that appear in a certain number of documents. If we want only the words that appear in the whole corpus, we would put in the total number of documents in our data set, and this would show you which words are most common in the whole dataset, giving us an idea of which words unite our articles in theme.

Play around with the parameters and see what outputs you get!

Lexos: K-Means Clustering



Go to the “Analyze” Tab and select “K-Means”. This analysis will work to cluster our documents together into different groups based on their similarity, using the k-means scores for each document. This can be useful from an inductive standpoint to see which articles are similar and fall together. If we haven't read the documents in our corpus, we can use these clusters to identify thematic or subject area clusters, articles that are talking about similar topics in our dataset, like one cluster could be Global warming Articles talking about pollution. We could look at a cluster, see what articles are in it, read a few of them and see what is common between them, and identify these topic clusters. This could be useful in guiding your research and reading, especially on things you haven't read, allowing you to identify clusters, sample them to see what is in each, and only read articles in the cluster that is most interesting to you or covers your research question or topic of interest.

You can change the output by going to the “Options” Tab, and change the number of clusters generated. The default number of clusters is half the number of documents you are analyzing. There is no hard and fast rule for choosing the number of clusters, it is more of a testing and experimental process where you choose different numbers of clusters, see what is in each, and choose the final number of clusters based on what number gives you the most clear, cohesive, and interpretable categories. You can also change the visualization of the clustering. The default is Voronoi, which is simply a scatter plot where the centroids for each cluster are plotted along with each document, and the areas around each cluster are colored and drawn to show what part of the graph is included in each cluster. You can use the 2D scatterplot which gets rid of the background colors, or the 3D scatterplot which does the same but has

slightly more visual depth.

You can also go to the “Tokenize” tab and change how word counts are being done to calculate similarity and produce clusters, in the same way the “Tokenize” tab in the “Top Words” page functioned, changing from tokens to characters, and changing the Gram level to move from single word, to double word and beyond counts for clustering comparisons.

Play around with the parameters and cluster the documents in different ways. Hover your mouse over any of the points in the cluster to see the document name. Use this to see which documents are clustered together and look at the text of the articles to get an idea of why these documents are being clustered together!

Lexos: Save or Reset Your Results

Lexos allows you to **save** your results as a Lexos file. If you do this, you can re-upload the Lexos file any time to access your cleaned-up corpus as well as the different analyses you've done.

You can also save individual visualizations as images (PNGs).

Finally, if you want to start over, you can "Reset" your Lexos dashboard. Lexos is an Open Source Software hosted at Wheaton College, so it might be prone to bugs, lag, or other issues that require you to reset it. If you find Lexos stops working or is producing weird results, reset Lexos and start over from the beginning.

What Other Keywords would be Interesting to Search for?

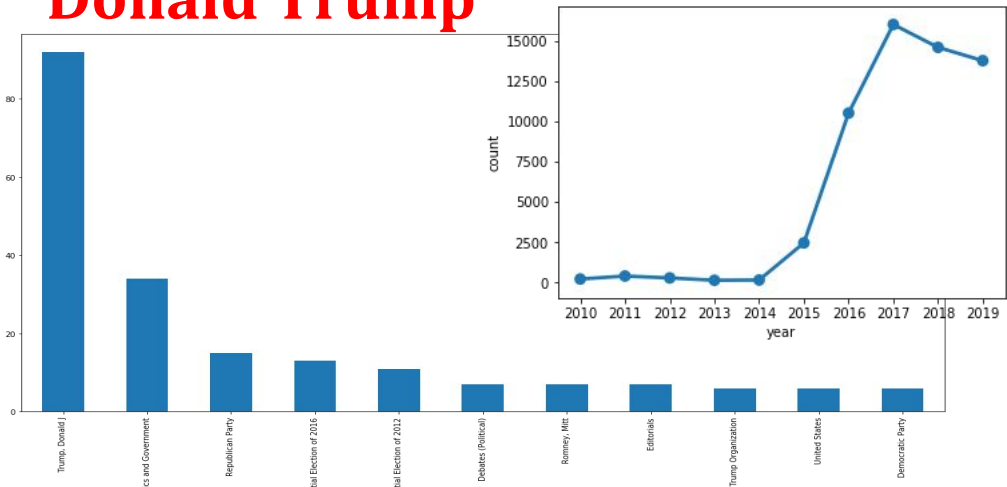
- We could use this script to use the NY Times API for any type of keyword search
- What would we find if we searched for “Donald Trump”?

```
# set search parameters
search_params = {"q": "donald trump", #this is where you enter your search term
                 "api-key": key,
                 "begin_date": str(year)+"0101", # create year range from the element variable
                 "end_date": str(year)+"1231"}
```

- You can follow along in this [Jupyter Notebook](#) if you want, or look at the results on the next slide

NY Times Articles related to Donald Trump

Frequency of Trump
Related Articles 2010-2019



Top Keywords for Trump
Articles

0	2010	207
1	2011	394
2	2012	276
3	2013	131
4	2014	148
5	2015	2453
6	2016	10522
7	2017	16029
8	2018	14607
9	2019	13779

Assignment: Use Lexos on the NY Times Trump Data

- Take the Trump Article Data provided over email, or at this [github link](#), and enter it into Lexos
- Perform the same upload, management, cleaning, analyzing and visualization processes we did for the Global Warming articles on the Trump articles
- Take a screenshot of the cleaned article window after you have applied stopwords
- Take a screenshot of one analysis or visualization you performed on this data
- Send the Screenshots to Professor Alden

Want to Learn More and How to do this Yourself? Take INSH 1500 for Summer 2

If anyone is interested in learning python, jupyter notebooks, and how to use APIs and do these analyses yourself, sign up for my Summer 2 Course, INSH 1500: Digital Methods for Social Science & Humanities, where we will learn to do this and more (including some Mapping & GIS, Network Analysis, Image Analysis) including going much more in depth into Computational Text Analysis using Python, teaching you new research methods to further social science inquiries. No coding experience Necessary!

INSH 1500: Digital Methods for Social Science & Humanities

Professor: Jeff Sternberg

Associated Term: Summer 2 2020 Semester (6/29/2020 - 8/18/2020)

CRN: 61178

Monday, Tuesday, Wednesday Thursday, 9:50 AM - 11:30 AM (Though Probably Online)

Course Description

"Introduces programming skills and computational methods through application to topics in the social sciences and humanities. Methods include computational text analysis, network analysis, mapping software and analysis, computational approaches to data, big data, and/or social simulation. Offers students an opportunity to develop an understanding of the use and significance of computational tools for social sciences and humanities. No previous programming experience required"

Contact and Resources

If you have any questions, contact me at:

Jeff Sternberg

DITI Research Fellow

sternberg.je@husky.neu.edu

Slides and data available at <http://bit.ly/diti-spring2020-alDEN-textanalysis>

Sign up for office hours at <https://calendly.com/sternberg-je/15min>