6-2021

A Text Analysis of British Welfare Debates

Emily C. Maanum
The Graduate Center, City University of New York

A TEXT ANALYSIS OF BRITISH WELFARE DEBATES

by

EMILY MAANUM

A master's capstone project submitted to the Graduate Faculty in Digital Humanities in partial fulfillment of the requirements for the degree of Master of Arts, The City University of New York

2021

A Text Analysis of British Welfare Debates

by Emily Maanum

This manuscript has been read and accepted for the Graduate Faculty in Digital Humanities in satisfaction of the capstone project requirement for the degree of Master of Arts.

_____	_____
Date	Michelle A. McSweeney
	Capstone Project Advisor
_____	_____
Date	Matthew K. Gold
	Executive Officer

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

A Text Analysis of British Welfare Debates

by

Emily Maanum

Advisor: Michelle A. McSweeney

This white paper details the process of conducting a textual analysis of three British parliamentary debates. It also discusses the development of Tableau visualizations (https://public.tableau.com/profile/emily.maanum#!/vizhome/TextAnalysisofBritishWelfareDebates/Story1) to display the results of the analysis, along with the creation of a GitHub repository (https://github.com/Maanume/DH_Capstone_Project) for documentation of the procedure and code, as well as a GitHub Pages site (https://maanume.github.io/DH_Capstone_Project/)for displaying the visualizations. Through text analysis, this capstone project examines the rhetoric used by British members of parliament (MP) while debating the creation of a welfare state during three specific parliamentary debates in 1944: *National Health Service*, *Social Insurance Part 1 and 2*, and *Employment Policy*. This project examines how MPs discussed the introduction of a welfare state in postwar Britain by exploring word frequency, lexical density and speaker participation in the debates. In past studies, historians have qualitatively assessed the origins of the welfare state, but there has been little quantitative study of these debates. This

project quantitatively surveys the rhetoric of MPs during the preliminary welfare debates through categorization and analysis.

ACKNOWLEDGEMENTS

I would like to express my thanks and gratitude to all those who have guided and supported me through the Digital Humanities (DH) program and the completion of my capstone project at the CUNY Graduate Center. First, I am greatly appreciative of my capstone advisor, Dr. Michelle McSweeney, for her words of encouragement and direction while completing this project. With her help, my initial ideas were developed and sharpened into the capstone project presented today. I would also like to offer my sincere thanks to all my professors from the DH program. Their courses over the past 2 years have challenged and inspired me to think more critically about the responsibility of Digital Humanities in academia and the world. I would also like to thank Dr. Matthew K. Gold for his guidance through the program.

I would like to express my deep gratitude to Dr. Lauren Tilton at the University of Richmond. Her undergraduate course, Introduction to Digital Humanities, opened my eyes to a whole new field of research and learning that greatly excited and deeply inspired me. Without her help and guidance, I would not be where I am today.

To my friends and family, thank you for your unwavering support during these past two years, especially this last one. Your encouragement and pep talks helped keep me sane through it all.

TABLE OF CONTENTS

DIGITAL MANIFEST

I. Capstone Whitepaper (PDF)

II. WARC files

   a. Project Website

      i. Archived version of GitHub repository at time of deposit.

      ii. Archived version of GitHub Pages with Tableau visualization at time of deposit.

III. Code and Other deliverables

   a. Zip file containing contents of GitHub repository at time of deposit.

A NOTE ON TECHNICAL SPECIFICATIONS

The README.md found within the GitHub repository provides information about the project and details the contents of the repository. The repository contains three different file types. The files that end in .md are markdown files that contain the Python code I utilized for my text analysis. These files also contain some explanation (in markdown) of what each code section does, as well as some instructions for getting started with either Anaconda or Bash to run the Python script. The files that end in .py are python files and strictly contain the code found in each of the .md files. This file format functions for easy download and reuse for those interested in using these sections of code in their own Python scripts. The .html file contains the backend code for displaying my Tableau visualizations on my GitHub Pages site. It also contains the link to my Tableau Public page where the visualizations are hosted.

The Tableau Public page contains visualizations created from the data I collected during my analysis. I created all the visualizations using Tableau 2020 software, and the data for the visualizations is stored in .csv files. These .csv files were used to manage the data and serve as a local source when I created my visualizations using Tableau Desktop.

NARRATIVE

Background

This project focuses on three British parliamentary debates that took place in 1944. In part, these debates occurred due to the introduction of the Beveridge Report to Parliament in November 1942, which provided a blueprint for social policy in Britain. The 1942 Beveridge Report was a forward-looking document that contained recommendations for the future of welfare in Britain. It has become central to modern debates about the origins of the welfare state and is renowned for its recommendations of universal welfare for all citizens. The report also attempted to redefine how the state interacted with the people by looking to establish a basis for a universal welfare initiative that included social insurance, health service and full employment.[1]

The Beveridge Report led to Parliamentary debates in the House of Commons discussing its contents as well as the future of the British welfare initiative and, in turn, prompted the government to issue the subsequent White Papers — a set of plans for post-war reconstruction regarding welfare. These White Papers, split into three parts and formally titled "A National Health Service," "Social Insurance Part I and II," and "Employment Policy," outlined the government's plan concerning health care, insurance and employment of its citizens. In the 1944 House of Commons, these three White Papers inspired additional parliamentary debates on their named topics: national health, social insurance and employment policy. The parliamentary debates concluded when the Commons finally welcomed the government's intention to create a comprehensive welfare scheme. By 1948, many policies outlined in these white papers and later debated in Parliament were implemented by the Labour Government, which came into power in

[1] Keith Laybourn, *The Evolution of British Social Policy and the Welfare State*, (Keele: Keele University Press, 1995), 213.

1945. Remnants of this welfare state can still be seen in Britain today, most notably in the enduring National Health Service.

Data

My project analyzes the 1944 British parliamentary debates in the House of Commons, which centered around the creation of a welfare state in Britain. The data originated as seven plain text files, each representing a day of debate, spanning three debates. As stated before, these debates discuss social insurance, the national health service and employment policy. I downloaded these text files from Hansard (https://hansard.parliament.uk), which is the official archive for all the UK Parliamentary sessions and debates dating back to the early 1800s.

Text Analysis Methods

Cleaning and Pre-processing

I used a Jupyter notebook to write my scripts and run my code. I found that using Jupyter notebook made it significantly easier to edit my code and run different sections as opposed to running the script in its entirety. I also found that using this type of computational notebook made it easier for me to run my scripts, to reproduce calculations with different data. I needed only to change the name of my text file to have the script run over my folder of text files.

Before starting my analysis, I combined my files to create four separate files — one for each debate topic and the final one containing all three debate texts. I did this by appending the files to each other using a Python script. My next step before analysis was to prep my data by cleaning and normalizing the text. To start, I processed my files by breaking down the text using a process called tokenization, which entails splitting text up into meaningful segments. Text can be tokenized by sentence or by word, and I chose to tokenize the text by word — meaning each word in the text, including punctuation, is its own token. Next, I created a function in Python to

remove punctuation and change all the letters to lowercase. Then, I removed stopwords from the text by importing the English stopword list from the Natural Language Toolkit (NLTK) library, then creating another function that looped through my text to remove them. Stopwords are commonly used words that usually dominate a text but would not provide much insight into its meaning. Any words that were not on the stopword list were added to a new document and saved to my computer. I repeated this process for each of the four files, and saved the documents as a "cleaned" version to use in my analysis. I chose to remove stopwords from the text because I felt a lot of commonly used words such as "the," 'that," 'is," "this," *etc*., can overrule an analysis. Also, they usually do not offer much insight into the text. I selected the NLTK's stopword list because it contained most of the words I wanted to remove and I also wanted to maintain consistency, as I used many NLTK libraries and functions while conducting my analysis. I did not add any additional words to the stopword list because I was concerned with reviewing the initial results of the text analysis — noting which words appeared with the highest frequency without interference from my potential addition.

After cleaning and normalizing the text, I used my newly created, cleaned text file to do some pre-processing. I again tokenized the text so each word is a token. While this might be a redundant step, I wanted to make sure the works were completely processed as tokens. Next, I conducted parts of speech (POS) tagging on the text. POS tagging is used to provide context to text; all the words in the text are categorized to different word classes, such as nouns, verbs and adjectives. I used the NLTK POS tagging function to tag the words in my text. This extra information attached to words helps with further processing like lemmatization, which was the next step in my process.

I lemmatized my text using the WordNet Lemmatizer from NLTK. The process of

lemmatization involves taking words of the text and returning them to the root form of that word. For example, before lemmatization, “hospital” and “hospitals” are read as two different tokens. After lemmatization, “hospitals” is read as “hospital” as well, bringing terms down to their root word. This is helpful when counting words, so words with the same root are changed and counted together. There are potential drawbacks to lemmatizing, such as loss of precision and nuance. However, I chose to use it because the debates were centered around specific topics, and thus it is probable to speculate that members of Parliament were using words in similar ways, lowering the risk of losing the exact meaning of words.

In addition to my four cleaned text files, I further developed data I created in another course using the same raw text files I downloaded from Hansard. I transformed the raw text files into rectangular format using a pandas dataframe and saved them as a csv file. Then, using a Python script, I appended the csv files together to create one large file that contained all three debates. I initially separated the data by speaker, making each new speech addition into a new row. Each row was given a Key ID, such as ‘nh1’ for the first speaker of the national health debate, to help make the data more organized. Then, I added the text of each speaker in a new column within the same row of the speaker who spoke text. For this dataframe, I did most of the initial cleaning using Excel functions. I cleaned the text by removing extra spaces and punctuation using the substitute function and the trim function in Excel. I also made all letters lowercase using the lower function. In addition, I removed stopwords from the text using the substitution function and the stopword list provided by NLTK. My columns of the dataframe before any analysis included: Key Id, Debate Line, Date, Debate, Speaker, Raw Text and Cleaned Text. I conducted POS tagging on this dataframe to find and count the nouns, verbs, adjectives and pronouns used in every line of the debates. The counts for each POS tag is its own

column.

Text Mining

After cleaning and pre-processing my data, the next phase was mining the text. I conducted most of my text mining by using NLTK. My initial step for analysis involved finding the total length of the raw text and its word count, before removing stopwords by using the len function. For example, the total length of the National Health debate is 75,255 words. After finding the length of the text, I found the number of unique words in my text. I found this number by using the set function which groups words together that are the same. Next, I used the len function to count the length of the set function which grouped all the unique items together. For the National Health debate, the unique word count is 4,890. With these two numbers, it was then possible to find the lexical density of the National Health debate. The lexical density of a text essentially signifies the linguistic complexity of a text. To find lexical density you take the unique word count divided by the total word count and then multiply it by 100. The higher the percentage, the more complex a text is, which means more varying vocabulary is used within a text. The lexical density of the National Health debate was 6.50%. I created a blank .csv file and recorded the total word count and unique word count for all three debates, and the totals for the combination of all three.

After finding the total word count, total unique word count and lexical density, I decided to examine word frequency. For this part of the process, I used my cleaned text because stopwords were removed and the remaining words were lemmatized. I found word frequency by using the FreqDist function from NLTK, and decided to find the top five most common words for each debate. For the National Debate the top words were “hospital,” “service,” “doctor,” “voluntary” and “medical.” I added the top five words to my .csv file and recorded the number of

times these words occurred during each debate.

In relation to words and word frequency, the next part of my analysis involved finding words that frequently appeared together, known as collocations. A collocation is a pair or group of words that are habitually juxtaposed. Some examples of two-word collocations in the English language are "fast food" and "pay attention." For my analysis, I chose to search for collocations that were bigrams, *i.e.,* two words. I applied the BigramCollocationFinder function from NLTK to search for bigram collocations in the text. Like the top five most common words, I chose to limit the collocations to the top five as well. For the National Health debate, the top five collocations were "white paper," "voluntary hospital," "hon member," "right hon" and "medical profession." In my .csv file, I recorded the top five collocations for each debate and the top 10 for the combination of the three debates. I repeated the process outlined above for all four of my cleaned text files.

GitHub repository and webpage

After completing my text mining, I created a GitHub repository using Git through my local terminal to host the Python script I created for my text analysis. The repository also holds .md files that contain my Python script along with some explanation of each section of code. As a coding beginner, I thought adding my commentary would be helpful for other newcomers looking to start their own text analysis projects. The inclusion of both these files in the repository gives visitors to the site access to the Python script and instruction on how it works to help further their knowledge. I chose to use Git to create my repository to enhance version control and organization as I edited both my .py and .md files. Git makes it easy to make changes to files on a local machine and then upload them to the GitHub repository without going online. Through

my GitHub repository, I also generated a GitHub webpage to display the Tableau visualizations, which are hosted on Tableau Public. I wanted my Tableau visualizations to be easily accessible from my GitHub repository, so visitors can have access to the tools I used to create the data for these visuals.

Tableau visualizations

Utilizing Tableau 2020 software, I created a Tableau storyboard containing five dashboards with the data I collected during my analysis. This data includes the .csv file I created with the results of my analysis from my .txt files, along with the pandas dataframe I further developed during this project. The first dashboard is an overview of all three debates. It includes an introduction of the debates and outlines their importance. The first visual is a bar chart that examines the number of speakers per debate. I included this visual because I think it is important for viewers to consider how many members of Parliament participated and spoke during these debates, as they directly influenced the rhetoric used. I chose to use a bar chart for easy comparison of the debates.

The second visual is a list of the top 10 bigram collocations across all the debates. This is important because it reveals some of the most important debate topics, as they appear on this list due to frequent use. I chose to display the collocations in an ordered list so that viewers could understand how each collocation compared to another. The third visual is a bar chart that features the lexical density of each debate. I chose to chart lexical density because it highlights how linguistically diverse and complex each debate was. An interesting aspect of this chart is that the Employment Policy debate had the largest number of speakers but the lowest lexical density. Likely, members of Parliament (MPs) were repeatedly using the same words when discussing

employment policy.

The fourth visual on this dashboard is a word cloud that displays the top 20 words used during the entirety of the debates. I included this visualization because like the collocations list, this word cloud hints at significant areas of discussion for these debates — such as “hospitals” in the National Health debate and “industry” during the Employment Policy debate. The fifth and final visualization is a stacked bar chart displaying the top 15 MPs who spoke the most words during the debates. The colors indicate which debate they participated in and how many words each MP said. I added this chart so that viewers could conceptualize who dominated the debates. In the tooltips, I included the position of the MPs of Parliament if applicable. For example, Sir J. Anderson was the Chancellor of the Exchequer, so that title is included in the tooltip and provides context for the individual.

The following three dashboards take an in-depth look at each debate, examining the words and speakers of each debate. The debate dashboards ordered to correspond to the date when they occurred. The National Health debate occurred in March 1944, the Employment Policy debate occurred in June 1944 and the Social Insurance debate occurred in November 1944. The first dashboard in this series, second in the Tableau storyboard, is dedicated to examining the National Health debate. The first visual is a bar chart presenting the top ten MPs that spoke the most words during this specific debate. The bar chart allows for assessment of each MP in comparison to one another. I included this visual because I wanted viewers to understand which individuals spoke the most, thus influencing the words used throughout the debate. This visual also includes additional information about Parliament positions if applicable. The second visual is a bubble chart of the top five words. I chose to use a bubble chart to use size to represent the number of times each word was used. The tooltips provide additional context for

each word, helping viewers understand their significance to the debate.

The third visualization is a bar chart which displays the ten MPs that spoke the most times during the debate. This is different from the other bar chart because it examines instances of speech, as opposed to number of words spoken. I included this visual to show that the individuals who spoke most words were not always the ones who spoke the most times. Many of the MPs who spoke more were often asking clarifying questions or giving answers to questions asked. The final visual is a line chart that displays the word count over the time of debate. I included this graph to provide viewers with a look at the rhythm of the debate. The graph is divided into two different colors, each one representing a separate day of the debate, allowing for comparison of each day. Each point on the graph is an instance that a MP spoke, with the word count giving insight into when the most impassioned speeches took place, denoted by large word counts following each other. Running over the line with a cursor, viewers are given the name of the MP who spoke at a specific point in the debate, along with their word count. This provides further data for analyzing the visualization. During the National Health debate, MPs spoke at length, revealing large spikes in the graph at the end of the first day, rivalling the second day where there was the most time between longer speeches by MPs.

The third dashboard covers the Employment Policy debate and the fourth dashboard highlights the Social Insurance debate. Each of these dashboards contain the same types of visuals from the National Health dashboard to allow for easy comparison between each debate. I chose to format the individual debate dashboards in the same way to provide uniformity and equal analysis. On the Employment Policy dashboard, the line graph of word count over time represents the three days of debates. The first day was the shortest with many long speeches in the middle of the debate, the second day had increased discussion at the end and the third day

saw the most discussion at the very beginning of the day. It is interesting to see how each day varied and how MPs responded to each other — indicated by their word count. On the Social Insurance dashboard, the line graph shows that the first day of the debate was longer than the second, but only one instance of speech was longer than 2,000 words. The second day, however, had many instances of MPs speaking over 2,000 words per speech, but during a shorter time frame. This means MPs were replying to each other at length during most of this day of debate.

The final dashboard highlights two visualizations of interest. This dashboard provides deeper context and discusses two interesting points that emerged from the textual analysis. The first is the list of collocations from the entirety of the debates first shown on the initial dashboard. The text positioned next to it explains why some of these words appear on the list and provides an explanation for what they mean within the context of these debates. The second visualization is the bar chart of lexical density, also from the first dashboard. For this visual, I explain what lexical density is and its importance to understanding texts. I also help interpret this graph and why it is noteworthy to the analysis of these debates. I added this dashboard to provide more analysis and information so viewers have a better understanding of this analysis and why these results bring about new questions to be answered.

The practices that worked best for conducting my text analysis included using pre-existing tutorials from the Digital Humanities Research Institute (DHRI) at the CUNY Graduate Center to help shape the process of my analysis. I also often utilized the NLTK book which provides an introduction into natural language processing (NLP) and the Natural Language Toolkit. When I struggled to fix my Python code or find answers to other questions that were not available at these resources, I usually searched online and was able to find a solution on blogs dedicated to text analysis and NLP. For both GitHub and Tableau, I have tutorials for past

courses that I often referenced. However, when I had questions related to either of these software, I utilized pre-existing tutorials from the DHRI and other resources.

Relationship to Focus Area and Previous Course Study

The inspiration for this project came from my interest in British welfare debates and a desire to conduct a text analysis, as well as improve my Python coding skills. This project explores the rhetoric of the British welfare debates, but it also stands an example, for those new to coding, to explore and learn from. This project has advanced alongside my development in the Digital Humanities program at the CUNY Graduate Center. The topic for this project was originally part of my undergraduate thesis, which explored the origins of the British welfare state through primary sources, specifically local newspapers and parliamentary debates. The seed for this project was originally planted during my first semester in the Digital Humanities program. The course Methods of Text Analysis, taught by Dr. Lisa Rhody, introduced me to the language of Python and the process of conducting a textual analysis. Throughout the semester, we worked with different types of text and explored the various methods of text analysis like word frequency, stylometry and topic modeling. I often referred to the assignments of this course to check my work for this project, as they gave helpful instructions related to the method and Python code.

During my time in Introduction to Digital Humanities, taught by Dr. Matt Gold and Dr. Kelly Baker Josephs, I had the opportunity to experiment with Voyant Tools, an online text analysis tool. For one assignment, we were directed to use any text we like and explore Voyant's tools and features. I chose to use a text file of one of the parliamentary debates to explore the tool. The preliminary results from this assignment sparked an interest to explore Python, text analysis and these debates on a deeper level.

Another course that was critical to the creation of this capstone was Introduction to Data Visualizations and Design: Fundamentals, which was taught by Dr. Michelle McSweeney. This course introduced me to Tableau, a data visualization software, and taught me the basics of creating visualizations as well as outlined important elements of design. For the final project of this class, I transformed the debate text files into a pandas dataframe and performed some data manipulation in Excel to explore these debates. I used this data to create a variety of visualizations in Tableau. The knowledge and experience gained from this course was instrumental in the formation of visualizations for this project.

A course that helped improve my Git and GitHub knowledge and skills was Patrick Smyth's Software Design Lab. I learned the basics of Git and how to create a GitHub repository to help with version control and file updates. Over the course of the semester, I made weekly updates to the files in to my repository for this class using Git. This helped me practice using my terminal to edit local files and instilled a habit of repetition that is important when using Git to edit and upload files. My final project for this course included a GitHub repository which detailed how to remove stopwords from a text file in Python using NLTK. The code for this project informed the cleaning script I use in my capstone.

Evaluation

There were obvious challenges and setbacks when drafting the Python scripts for the text analysis, which I believe to be typical when working with code. Many times, when writing and testing my scripts, my code did not work. Usually, the issue was my code was syntactically invalid or its function was not supported by the environment. When I encountered an issue, the error messages pointed out where in the code the problem occurred and gave helpful suggestions of what it believed the problem to be. This made it easy to fix the issue and allow the code to run

successfully. If I received an error message I did not understand, I researched online and found its meaning and solutions to resolve this error. If I encountered issues with structure, I referenced the DHRI tutorials to help remedy these issues. Since there are a variety of ways to stricture code to perform functions, I encountered a lot of trial and error to ascertain which segments of code were successful in performing the types of functions I wanted.

The success of the project include the creation of workable code for analysis, a GitHub repository and informative Tableau visualizations. The Python scripts include code for cleaning and pre-processing text and for textual analysis exploring word frequency and lexical density that can be utilized by others. The GitHub repository holds these python scripts along with markdown files that provide instruction and explanation for what I did and reasoning behind these choices. The Tableau visualizations transform the data collected during the analysis into understandable and digestible content. These visualizations open the door for new questions and future investigations of these welfare debates.

An aspect of my original proposal that is not part of the final project is results from an analysis using topic modeling. During the development of this capstone I did perform topic modeling on these debates by using a Python script. However, the results did not reveal any new knowledge or interesting patterns. For topic modeling, it is more productive to have a rather large corpus documents. I think topic modeling was not successful because my corpus was on the smaller side. I also think because the topic of each debates was very specific, assessing new topics is not as clear as if the corpus covered a larger scope of documents, providing more variety. Even though topic modeling did not reveal any new information about the debates, I am glad I conducted this analysis because it helped to improve my Python coding knowledge and skills as this script dealt with new packages and functions I was unfamiliar with.

Continuation of Project

I do not plan to continue working on the specific objectives of this project after the completion of my MA. I believe this project has fulfilled the goals I set out to meet. In the future, however, I do plan to expand my analysis of these British welfare debates. I plan to further investigate the members of Parliament who spoke the most during these debates to evaluate their sentiments toward the British government and the creation of a welfare state. For another study, I also hope to expand my corpus of documents to include other debates related to other British welfare initiatives that span from 1944 to the present. With a larger corpus, I plan to perform topic modeling to examine if any interesting topics or patterns emerge from this larger quantity and extended timeframe. I am also interested in creating a classifier to measure sentiment from these three 1944 welfare debates to use on other British welfare debates that occurred after 1944. I hope to classify instances of speech with a debate as positive, negative, or other.

SELECT BIBLIOGRAPHY

Addison, Paul. *The Road to 1945: British Politics and the Second World War*. London: Jonathan Cape LTD, 1975.

Bengfort, Benjamin, Rebecca Bilbro, and Tony Ojeda. *Applied Text Analysis with Python*. O'Reilly Media, Inc., 2018.

Bird, Steven, Ewan Klein, and Edward Loper. *Natural Language Processing with Python--- Analyzing Text with the Natural Language Toolkit*. 1st ed. O'Reilly Media, 2009.

Cohen, Deborah. *The War Come Home Disabled Veterans in Britain and Germany*. Berkeley: University of California Press, 2001.

Fraser, Derek. *The Evolution of the British Welfare State: A History of Social Policy since the Industrial Revolution*. London: Palgrave Macmillan, 2003.

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy. "Text as Data." *Journal of Economic Literature* 3, no. 57 (2019): 535–74.

Laybourn, Keith. *The Evolution of British Social Policy and the Welfare State*. Keele: Keele University Press, 1995.

Nguyen, Dong. "How We Do Things With Words: Analyzing Text as Social and Cultural Data." *Frontiers in Artificial Intelligence*, 2020.

Pedersen, Susan. *Family, Dependence, and the Origins of the Welfare State: Britain and France, 1914-1945*. Cambridge: Cambridge University Press, 1993.

Pedersen, Susan. *The Guardians: The League of Nations and the Crisis of Empire*. Oxford: Oxford University Press, 2015.

Rawson, Katie, and Trevor Muñoz. "Against Cleaning." *Debates in the Digital Humanities*, 5, 2019.

Westerling, Kalle. "Introduction to Machine Learning." GitHub Repository. DHRI-Curriculum, April 30, 2020. https://github.com/DHRI-Curriculum/machine-learning.

Westerling, Kalle. "Introduction to Text Analysis with Python and NLTK." GitHub Repository. DHRI-Curriculum, March 9, 2021. https://github.com/DHRI-Curriculum/text-analysis.