Southern Methodist University

SMU Scholar

Multidisciplinary Studies Theses and Dissertations

Multidisciplinary Studies

Spring 5-13-2023

Grammatical Triples Extraction for the Distant Reading of Textual Corpora

Stephanie Buongiorno
sbuongiorno@smu.edu

Stephanie Buongiorno
Southern Methodist University, sbuongiorno@smu.edu

Follow this and additional works at: https://scholar.smu.edu/engineering_multidisciplinary_etds

 Part of the Other Computer Engineering Commons

Recommended Citation

Buongiorno, Stephanie and Buongiorno, Stephanie, "Grammatical Triples Extraction for the Distant Reading of Textual Corpora" (2023). *Multidisciplinary Studies Theses and Dissertations*. 3.
https://scholar.smu.edu/engineering_multidisciplinary_etds/3

This Dissertation is brought to you for free and open access by the Multidisciplinary Studies at SMU Scholar. It has been accepted for inclusion in Multidisciplinary Studies Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.

GRAMMATICAL TRIPLES EXTRACTION FOR THE DISTANT READING OF TEXTUAL CORPORA

Approved by:

Dr. Corey Clark and Dr. Jo Guldi
Computer Science, History
Dissertation Committee Chairperson

Dr. David Lin
Computer Science

Dr. Miju Ahn
Operations Research and Engineering Management

Tim Cassedy
English

Mark Fontenot
Computer Science
Northeastern University

GRAMMATICAL TRIPLES EXTRACTION FOR THE DISTANT READING OF TEXTUAL CORPORA

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Applied Science

by

Stephanie Buongiorno

M.A., English, West Virginia University
B.S., English, The University of Texas at Arlington

May 14, 2023

Copyright (2023)

Stephanie Buongiorno

All Rights Reserved

ACKNOWLEDGMENTS

I would like to thank my entire committee for supporting this dissertation. This is not a MySpace page, and the order in which I thank people individually does not equate to how much appreciation I have. I want to thank Dr. Corey Clark for his unrelenting support and for fostering spaces for innovation and creativity. His lab has become a home for my creative algorithms and processes. I want to thank Dr. Jo Guldi for providing the intellectual space to explore and grow diverse skills across the humanities and sciences. Working on her grant was a pivotal moment in my education that I would not have had otherwise. I want to thank Dr. Mark Fontenot for advocating for me during my transfer to the Applied Science PhD and then, after my transfer, equipping me with the skills to perform sophisticated feats of engineering. I want to thank Dr. David Lin for his patience, thoughtfulness, and willingness to review my work in statistics. I want to thank Dr. Miju Ahn and Dr. Tim Cassedy for their guidance and willingness to provide feedback during my degree. I would like to thank Austin Kasper for being a stable rock and a source of laughter and friendship leading up to, and throughout, my PhD. Without Kasper, continuing a PhD during a global pandemic would have been a much more difficult thing.

Buongiorno, Stephanie M.A., English, West Virginia University, 2017
B.S., English, The University of Texas at Arlington, 2014

Grammatical Triples Extraction for the Distant Reading of Textual Corpora

Advisor: Dr. Corey Clark and Dr. Jo Guldi

Doctor of Philosophy conferred May 14, 2023

Dissertation completed MONTH DAY, YEAR

Grammatical triples extraction has become increasingly important for the analysis of large, textual corpora. By providing insight into the sentence-level linguistic features of a corpus, extracted triples have supported interpretations of some of the most relevant problems of our time. The growing importance of triples extraction for analyzing large corpora has put the quality of extracted triples under new scrutiny, however. Triples outputs are known to have large amounts of erroneous triples. The extraction of erroneous triples poses a risk for understanding a textual corpus because erroneous triples can be unfactual and even analogous to misinformation. Disciplines such as the social sciences, history, and literature rely on accurate representations of events. In some cases, misrepresentations of language can be as problematic as describing a historical event that never occurred. The present research proposes a method of triples extraction that has been designed to meet the increasing need for high-accuracy triples outputs for the analysis of text. We propose a solution aimed at reducing errors related to: a) ungrammatical extractions; b) double counting; and c) the missed detection of triples. To improve the accuracy of triples extraction, we implement a series of 12 linguistic rules that leverage syntactic dependency parsing. For its case studies, this dissertation draws upon three data sets: a) Wikipedia; b) the 19th-century British Parliamentary debates, also known as "Hansard"; and c) half a year of online news articles (Aug. 2021 - Dec. 2021) from FOX News and NPR. In its final chapter, this dissertation offers a pedagogical piece that applies triples extraction to teach concepts related to data analysis. Extracted triples are thus evaluated through two means: a) in Chpater 1, precision

and recall is used to vet the accuracy of the present method and b) in chapters 2 and 3, we use human observation to show how the present method of triples extraction can give an accurate and insightful perspective into textual corpora that rivals and, in some cases, exceeds existing methods.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

This dissertation is dedicated to any being who has made any impact on me what-so-ever.

Chapter 1

Definitions

1.1 Subject

The person, place of thing performing the action of a sentence.

1.2 Verb

A word that describes physical action (e.g. "run"), mental action (e.g. "think"), or a state of being ("was").

1.3 Object

A noun or pronoun receiving the action of the verb.

1.4 Adjective Predicate

Adjective(s) that modify or describe the subject and are linked to the subject via a linking verb.

For example, in the sentence "the property raised in value," the phrase "raised in value" contains a verb followed by an adjective predicate.

1.5 Sentences with Leftward Syntactic Movement

Leftward syntactic movement occurs when an object or verb precedes a subject. English is considered a rigid "subject-verb-object" language, meaning that within a sentence, the subject usually comes first followed by the verb and then the object. However, sometimes grammatical sentences will feature the verb or object preceding the subject. Consider the following sentence:

"After watching it like 5 or 6 times, I think I see what happened."

In this example, the verb "watching" and the object "it" come before the subject "I."

Leftward syntactic movement also occurs when the object comes before the subject, but the verb follows the subject like in standard syntax.

1.5.1 Interrogative Sentences

Interrogative sentences are sentences that ask a question and are punctuated with a question mark. As mentioned earlier, subjects generally precede the verb or object in an English sentence. An exception is when asking a question. Questions are often structured by "wh-" fronting, where the object is transposed in front of the subject. Consider the following sentence:

> "Which sibling was in the image?"

In this example, the object "sibling" comes before the subject "image," the verb "was," and the preposition "in."

1.5.2 Compound Sentences, Complex Sentences, and Compound-Complex Sentences

Compound sentences are sentences where two or more independent clauses are joined together with a coordinating conjunction such as "for" or "yet," or a conjunctive adverb such as "however." Complex sentences are sentences that have one independent clause and one or more subordinate clauses that modify the main subject or verb of a sentence. Compound-complex sentences are sentences that contain at least three clauses: two independent clauses and one or more dependent clauses. In total, a compound-complex sentence can have three or more sets of subject-verb relations. Consider the following sentence:

> "He, however, intended to confine his observations within a narrow compass, not wishing to add one word to what had been so ably stated by his hon. and learned friend (Sir Charles Wetherell), although he did not agree with him entirely on some points."

This example contains independent clauses ("He, however, intended to confine his observations..."; and "he did not agree with him...") and a dependent clause ("not wishing to add one word...").

Chapter 2

Introduction

2.1 Introduction

The Digital Humanities (DH) is a field of academic research at the nexus of computation and the disciplines within the humanities (such as history, English, media studies and more). Scholarship in DH includes both the process of building computational methods for processing textual data, as well as the application of these methods for analysis of the data. The computational methods used in the humanities are often developed within scientific fields for scientific inquiry, and are later adopted and applied by humanities researchers to their own work.

The motivation for adopting computational techniques is that they provide insight into new dimensions of a corpus–dimensions such as the number of times a series of verses appears in 100 years of American news articles, the evolution of narrative across blogs or the changing topics within 19th-century British debates on property. [1], [2] [3]

This dissertation, however, explores how applying methods originally designed for scientific fields to humanities data can conversely provide insight into the method including areas of improvement. This dissertation makes this case by taking one method–triples extraction–and improving upon its design in a way that reckons with shortcomings in its performance on data sets traditionally studied in the humanities and adjacent fields.

In this way, this dissertation explores the potentials of software designed expressly for the humanities and social sciences. What makes software designed for the humanities different is that it embodies the ideals of humanistic inquiry in its design.

2.2 Digital History

DH can be broken into subfields that make specific assumptions about how textual data should be interpreted. Researchers in the field of literature, for example, might make as-

sumptions about how data should be argued about that are different from researchers in the field of history. While the computational methods explored by this dissertation (methods like triples extraction) could be applied to any of the subfields within DH that collect, analyze, and interpret textual data, this dissertation focuses on computational method and data as they are used within the field of digital history.

Digital history is a discipline within the humanities that uses computational methods to understand change over time. Research within digital history is often guided by the principles of "distant reading" or "longue durée analysis", both of which are styles of interpretation that favor a "birds-eye view" of the generalizable trends within time-series analysis instead of relying solely on qualitative, close-readings of individual passages for information about events.

Oftentimes, the computational methods used for analyzing textual data for digital history (as well as DH more broadly) draw upon the "bag-of-words" model of analysis. In this model, a text is represented by a multiset of words that disregards grammar or word order, but keeps multiplicity. The common methods drawn upon by digital history that use the "bag-of-words" model include basic word counts and summary statistics. While these basic methods are insightful, this dissertation advances research by focusing on triples extraction, a method that retains information about the relationships in a text–relationships such as which verbs are performed by which subjects, or which subjects act upon which objects. However, many of these more advanced methods–like triples extraction–would benefit from methodological changes to make them more relevant to textual analysis.

2.3 Problem Definition

This dissertation focuses on how a triples extraction method designed expressly for the humanities can address the growing needs of textual analysis.

"Triples" are relational sequences of words extracted from natural language sentences. They are often represented as arg1-rel-arg2 where "arg1" and "arg2" are noun phrases that share a semantic relationship determined by "rel," a verb phrase. While triples can be composed of different part-of-speech sequences, much existing research applying triples extraction for textual analysis has focused on triples that represent the subject-predicate relationships

in a sentence, such as subject, verb, object (SVO) relationships or the subject, verb, predicate adjective (SVA) relationships. The research in this dissertation will also focus on SVO and SVA relationships and explore methods of extraction that benefit textual analysis.

For its case studies, this dissertation draws upon two data sets relevant to the humanities and the social sciences. Here, "humanities data" is defined as data that is commonly the subject of study within the humanities such as legal records, newspapers, literature, and more. The two data sets used by this study are: a) the 19th-century British Parliamentary debates, also known as "Hansard"; and b) half a year of online news articles (Aug. 2021 - Dec. 2021) from FOX News and NPR.

2.4 Grammatical Triples Extraction

Grammatical triples extraction has become increasingly important for the analysis of large, textual corpora. By providing insight into the sentence-level linguistic features of a corpus, extracted triples have supported rich interpretations of some of the most relevant problems of our time. They have been used to track the development of anti-vaccination narratives originating on "Mommy Blogs" (Tangherlini et. al, 2016) and descriptions of climate change across online news sources. [4] They have been used to model scientific literature on COVID-19, and to trace how Parliamentary discourses on the rise of peoples' rights have changed over time and may be influencing the politics of our present moment. [4], [**?**]

One reason triples extraction has such wide-reaching usefulness across the many disciplines that analyze text is because triples extraction can distill large amounts of natural language text (e.g. language as it is spoken by humans in the real-world) into minimal units that are encoded with meaningful information about the language patterns within a corpus. Triples are often

represented as arg1-rel-arg2 where "arg1" and "arg2" are noun phrases that share a semantic relationship determined by "rel," a verb phrase. While triples can be composed of different part-of-speech sequences, existing research has focused on triples that represent the subject-predicate relationships in a sentence, such as subject, verb, object (SVO) relationships or the subject, verb, predicate adjective (SVA) relationships. Consider the following sentence from Hansard:

"But it is urged that landlords may exercise a grievous oppression over tenants."

Within this sentence is the triple: "landlords-exercise-oppression." This triple is a meaningful abstraction that retains the core action described by this sentence.

One could use triples extraction to see a general sense of change over time by aggregating all instances where the subject "landlords" the verb "exercise" and the object "oppression" are present. This abstraction has enabled researchers to quantify how the language of action has changed over time–a problem that will be explored in greater detail throughout this dissertation.

2.5 Dissertation Organization

This dissertation is organized according to the "three-article model" where, instead of being written as a single book manuscript intended for publication at a university press, this dissertation is broken into three stand-alone sections that have been brought together as a cohesive piece. The choice to write a three-article dissertation is meant to foster a future career that focuses on innovative application and actionable deliverables, not single authorship.

The three sections that make up this dissertation are: Section 1, "Method," Section 2, "Application," and Section 3, "Pedagogy." Together, these sections generate research at the nexus of computer/information science and the humanities. Along with these written pieces, each section produces open-source resources such as data sets and user-friendly software packages. The following paragraphs describe the different sections of this dissertation including plans for disseminating the deliverables generated while performing research.

2.5.1 Section 1: Method

Section 1, entitled "Syntactic Dependency Relations and the Extraction of Grammatical Triples," introduces a novel method of grammatical triples extraction, posextract, that was expressly designed for the analysis of textual corpora.

This section grew out of a modest question: How might an analysis of triples in the 19th-century British Parliamentary debates (also known as "Hansard") give researchers insight into how language has changed over time? As I began to explore the potentials of triples extraction to answer this question, I realized that existing methods had overall poor

accuracy when used on the verbose and even flowery diction of this historical vernacular. Reviewing the literature on triples extraction shows that published methods were tested on contemporary corpora–usually Internet speech–and that no benchmark exists for evaluating triples extraction methods on historical corpora. Therefore, I developed a novel method of triples extraction designed on both contemporary and historical data that improves the precision and recall measures of existing methods for textual analysis for digital history.

posextract was designed for textual analysis. It responds to criticism directed at low-quality triples extraction outputs that inhibit human interpretation. When applied to humanities data for the sake of interpretation, erroneous extractions pose great risks because erroneous triples can be unfactual or even analogous to misinformation. Disciplines such as history, literature, and the social sciences, rely on accurate representations of actions and events. In some cases, misrepresentations of language can be as problematic as describing a historical event that never occurred.

Therefore, posextract has been designed to meet the increasing need for high-accuracy triples extraction for textual analysis. The solution described by this chapter is aimed at reducing errors related to: a) ungrammatical extractions; b) double counting; and c) missed detection of triples. It also enhances analysis by returning clean, analysis-ready triples structured by a canonical form (that is, a standardized form making generalization easier for analysis).

Along with supplying a new method of triples extraction, Chapter 1 introduces the first benchmark data set for triples extraction run on historical corpora. While other benchmarks do exist, they are exclusively made up of contemporary text. The Hansard corpus was selected for this benchmark because its verbose and flowery languages pose challenges to grammatical parsing.

Dissemination: This chapter is written as both a dissertation chapter and as a stand-alone article that could be submitted to conferences in NLP (e.g. ACL, EMNLP, COLLING) if I were to continue an academic career. The software described by this chapter, posextract, is designed for distribution across major platforms such as pypi and GitHub. pypi is an open-source distribution platform with 2.5 billion monthly active users. Software submitted to pypi can be installed easily from their service with a single command: pip install posextract.

GitHub is an open-source hosting platform designed for long-term hosting of code. It is used by over 83 million developers and professional software engineers around the world.

2.5.2 Section 2: Application

Section 2, entitled "What the Entities Named in Half a Year of News Articles on COVID-19 Can Tell Us About Media's Selective Attention to Information and Authority," serves as a case study demonstrating how grammatical triples extraction can enhance textual analysis. Here, grammatical triples are extracted to gain deeper insight into the nature of news coverage on COVID-19 by showing how certain actions are attributed to different people, organizations, and events. In this way, Section 2 uses grammatical triples extraction to tell a "present day" history of the COVID-19 pandemic as it is represented by these two news outlets.[1]

Chapter 3

Method: Syntactic Dependency Parsing and the Extraction of Grammatical Triples

3.1 Introduction and Motivation

Information extraction (IE) extracts relational sequences of words, known as "triples," from natural language sentences (e.g. sentences as they are spoken by a human as opposed to structured for a computer) [5]. Triples are often represented as arg1-rel-arg2 where "arg1" and "arg2" are noun phrases that share a semantic relationship determined by "rel," a verb phrase. While triples can be composed of different part-of-speech sequences, existing research has focused on triples that represent the subject-predicate relationships in a sentence, such as subject, verb, object (SVO) relationships or the subject, verb, predicate adjective (SVA) relationships. Large repositories of extracted grammatical triples have benefited a wide range of natural language processing (NLP) objectives such as answering questions, improving decision making, improving ontology learning, and summarizing the language features or ideas present in a data set [6].

In recent research, extracted triples have become increasingly important for the analysis of large, textual corpora. By providing insight into the linguistic features of a corpus, extracted triples have supported rich interpretations of some of the most relevant problems of our time by giving the analyst insight into topics such as causality and action as they are encoded within the grammatical structure of a sentence. [7], [8], [9], [10], [11] Triples have been used to track the development of anti-vaccination narratives originating on "Mommy Blogs," descriptions of climate change across online news sources, to model scientific literature on COVID-19, and to trace how Parliamentary discourses on the rise of peoples' rights have changed over time and may influence the politics of our present moment. [12], [1], [4], [13]

The growing importance of triples extraction for analyzing large corpora has put the quality of extracted triples under new scrutiny, however. Results are known to have large

amounts of erroneous triples. Some studies have proposed extensive post-processing measures to validate extracted triples. [4] Others have proposed that erroneous representations of information are inherent to triples extraction as a method, and that different part-of-speech extraction patterns may serve as a more meaningful alternative. [8] In analyses of textual corpora–where factual representations of current or historical speech is paramount–a high number of erroneous and missed triples could be uninformative at its best and equivocal to misinformation at its worst.

3.1.1 Problem Definition

The need for a method of triples extraction designed for textual analysis is clear: the extraction of erroneous triples, and the missed detection of triples, poses a risk because these outputs can lead to misinformation. Disciplines such as history, literature, and the social sciences, rely on accurate representations of events. In some cases, misrepresentations of language can be as problematic as describing a historical event that never occurred, or obscuring an event that did occur.

The present research proposes a method of triples extraction, posextract, which has been designed to meet the increasing need for high-accuracy triples outputs for the analysis of text. This method can run on a laptop without additional training, which saves computer resources and user time, and makes the software more accessible to users with different levels of technical understanding. Unlike existing methods, posextract exports triples while running, decreasing the amount of memory used by the program at a given time.

The proposed solution is aimed at reducing errors related to: a. ungrammatical extractions; b. double counting; and c. the missed detection of triples. We define ungrammatical extractions as cases where the identified triple is composed of parts-of-speech that do not share a grammatical SVO/SVA relationship. For example, a subject is assigned the wrong object from a sentence. We define double counting as cases where the same knowledge is extracted multiple times. We define missed triples as cases where a grammatical triple is present in a sentence, but is not identified by the triples extraction method. To improve the accuracy of triples extraction, we implement a series of 12 linguistic rules that leverage syntactic dependency parsing.

Like previous studies on triples extraction methods, we evaluate the effectiveness of posextract by comparing its precision and recall against existing methods. Also, like previous studies, posextractś performance is evaluated on text from a contemporary, online source. But because the present study responds to the growing need for a high-accuracy method of triples extraction for the analysis of textual corpora, we also evaluate posextract and existing triples extraction methods on historical records, specifically the 19th-century British Parliamentary debates, also known as Hansard. We chose Hansard because of its popularity across the many disciplines that refer to 19th-century legal documents as their foundation, such as many subdisciplines within history, law, sociology, and political science. This study thus sets a precedent for evaluating triples extraction methods on sentences with complex linguistic features that are different from those usually found within contemporary online sources.

3.1.2 Canonical Form for Extracted Triples

This study also addresses issues that cause ambiguity in triples outputs (which may limit the usefulness of triples for the analysis of text). As this research suggests, ambiguity is often caused by the absence of a canonical form. Few existing methods structure their triples outputs by a standardized form in which the parts-of-speech making up a triple are consistently ordered, despite the usefulness of form in helping analysts understand the content of a triple.

Our canonical form establishes that the first noun of a triple is always the subject and the last word is always the object or adjective predicate, regardless of their order in the original sentence. Superficial parts-of-speech, like the article "the" or conjunctions like "and," are not extracted. If negative determiners or a negating adverb are present, they always appear before the word they modify. The following represents the structure of a grammatical triple, where parts-of-speech in parentheses are conditional:

(negDeterminer)
subject
(negAdverb)
(aux verb)

verb

(preposition)

(negDeterminer)

object|adjective predicate

(object prepositional phrase)

3.1.3 Minimal Units

Triples extraction aims to distill sentences into a set of small, isolated units that are encoded with meaning from the original sentence. As asserted by Stanovsky and Dagan [14], the span of each unit detected by a triples extraction system should be as minimal as possible while still preserving the original information [reductive reading]. Having shorter entities was found to be useful in several semantic tasks [6] [14].

The current research appeals to this model of triples extraction while also presenting the user with choices for how much information these minimal forms should retain. We believe these choices will provide users with greater flexibility in their analyses. Consider the following sentence:

"The soldiers were terminally ill with Syphilis."

This sentence can be represented by two triples:

"soldiers were terminal" "soldiers were ill"

Or by a single triple that retains additional information about the relationship between "terminal" and "ill":

"Soldiers-were-terminally-ill"

Or the sentence can be represented by a triple that retains information about the prepositional phrase–"with Syphilis"–that shares a relationship with "ill."

"Soldiers-were-terminally-ill-with-Syphilis"

The method proposed by the present research gives users the ability to choose how the exported triples should be represented, giving researchers more control over the information they extract and the analysis they perform.

3.2 Previous Work

The problem of extracting meaningful triples has been addressed several times before. This section describes existing triples extraction systems and the data set(s) upon which they were evaluated. posextract uses syntactic dependency parsing and applies linguistic rules to extract triples. Therefore, this section focuses on extraction methods that are deterministic [15]. Algorithms that use dependency parsing, like posextract, often have a higher computational cost than those that do not. However, methods that use dependency parsing often return better results. [16]

3.2.1 TextRunner

The first popularized Open IE system was TextRunner, which used a Naive Bayes model with unlexicalized POS and NP features. It was trained on the Penn Treebank wordbank. TextRunner was designed with the assumption that for a triples extraction method to be desirable, it must be fully automated and be able to run on a large, heterogeneous corpus (instead of a manually selected corpus tailored to an extraction method). It makes the assumption that these results can only be achieved if the system ignores the lexical features of words and handles text as an unbound set of relations. [17], [18] TextRunner's design purposefully rejects all hard-coded linguistics rules, citing their potential to limit the range of corpora upon which a triples extraction method can be run. TextRunner's output, however, is rife with ungrammatical and unfactual triples.

3.2.2 WOE

The WOE system increased the accuracy of triples extraction and showed ways that syntactic dependency parsing can increase precision and recall. [19] WOE was trained on a corpus made up of Wikipedia articles. WOE only nominally uses syntactic dependency parsing and largely ignores the features of words while handling text as an unbound set of relations. And while the integration of syntactic dependency parsing improved its accuracy,

the triples output is still rife with errors. A major driver behind WOE's limited use of grammatical rules is a decision to optimize the method for speed at the expense of accuracy.

3.2.3 ReVerb

ReVerb uses a series of syntactical and lexical constraints applied to the verbs in a sentence. It first identifies relational phrases that match its syntactic and lexical constraints, and then finds a pair of noun phrase arguments for each relational phrase. It was trained on sentences from Yahoo's random link service. ReVerb more than doubled the precision of TextRunner and WOE. However, this approach to pattern matching can often cause errors where the wrong parts-of-speech are depicted as sharing a grammatical relationship.

3.2.4 OLLIE

OLLIE substantially improves the accuracy of triples detection by implementing dependency parsing and bootstrapping methods to create a large corpus of lexical patterns to guide the extraction process. [20] Its learned templates for extraction patterns are mapped onto dependencies in a sentence, from which it extracts relational phrases and associated arguments. OLLIE was evaluated on three different data sets: a corpus of news, Wikipedia, and a biology textbook. The news and Wikipedia data is made up of a random subset of the data set Wu and Weld used for evaluating WOE. OLLIE's interventions substantially improved the accuracy of triples extraction, but its templates can extract inconsistent configurations of grammatical triples (where superficial part-of-speech, like "the," may or may not be extracted from a sentence), and it is untested on historical data.

3.2.5 Stanford's OpenIE

Stanford's OpenIE introduced multiple interventions to the process of extracting triples through its operations that are guided by linguistic reasoning and the identification of a sentence's underlying grammar. It uses a classifier made up of fourteen hardcoded linguistic patterns that identifies the lexical features of a sentence. The classifier is based on natural logic. Natural logic is a type of logic that is applied to the syntactic structure and semantic properties of a sentence's lexical constructions to determine the parameters of a sentence's lexicon. [21] After identifying the lexical features of a sentence, the system uses a greedy

search pattern to traverse each dependency tree recursively, at each step predicting whether an edge should yield an independent clause. Incorrect predictions result in incomplete extractions, which are identified and discarded via natural logic. However, even with these linguistic rules, the accuracy of its triples yield might be considered too low for robust, textual analysis, for which researchers may want to use triples extractors.

3.2.6 Comparison of Triples Extraction Forms

This section gives examples of extracted triples from ReVerb, OLLIE, and OpenIE to show how ordering triples with a canonical form might benefit analysis by providing a consistent representation of extracted information.

Table 3.1 shows various triples constructions extracted by ReVerb.

ReVerb Extractions		
Row	Sentence	Extracted Triple(s)
1	"I have not said that the manufacturers do not keep a register." (C19 Hansard)	(the manufacturers do not keep a register)
2	"Having thrown forth this observation, he would next remark that more blood had been shed in Ireland during the Administration of the right hon." (C19 Hansard)	(he would next remark that more blood) (he had been shed in Ireland)

Table 3.1: Examples of triples extracted with ReVerb. Each unit of the triple is separated by multiple spaces.

ReVerb regularly retains additional information from superficial parts-of-speech such as articles. In Table 3.1 row 1, “the manufactures” would be counted separately from just “manufactures” without having undergone additional post processing. Similarly, “do not” would be counted separately from “did not” or “don’t.” Row 2 from Table 3.1 show how an inconsistent form can contribute to ungrammatical extractions (e.g. “he had been shed in

Ireland" instead of "his blood" being shed).

Table 3.2 shows the various triples constructions extracted by OLLIE. Research behind OLLIE argued for the importance of a canonical form as a way to facilitate analysis, but OLLIE has inconsistencies in the amount of information that it retains in a triple and it frequently double counts.

OLLIE Extractions		
Row	Sentence	Extracted Triple(s)
1	"Advertisers would pay for the privilege of having one of their hilarious images in the compilations and thereby fund the work." (Reddit)	(Advertisers ; would pay for ; the privilege having one of their hilarious images in the compilations)
2	"They say that they want cheapness, and to have our industries worked cheaply; but cannot they see that if they put taxation upon our own manufactures which is not put upon foreign manufactures, our own producers cannot compete with the foreigner as thoroughly as if both were taxed equally." (C19 Hansard)	(they ; want to ; cheapness) (they ; put ; taxation) (they ; put taxation upon ; our own)

Table 3.2: Examples of triples extracted with OLLIE. Each unit of the triple is separated by a semicolon as added by OLLIE.

Comparing row 1 and row 2 from Table 3.2 shows the stark difference in the amount of information that may be extracted. In row 1, the extracted triple retains nearly the entire original sentence (including numerous superficial parts-of-speech such as "would" or "the"), yet row 2 shows an example where the extracted triple is distilled to just its subject, verb, and object. While the triple represented in row 1 is grammatical and factual, it does not lend insight into high-level patterns or trends, which might just be advertisers paying for

privileges, or compilations having images.

Without adhering to a strict canonical form, OLLIE also frequently returns multiple triples for a single set of SVO/SVA relations in text, causing these triples to be "double counted" as the same information is extracted twice. Consider row 2 from Table 3.2, where both "they put taxation" and "they put taxation upon our own" are extracted.

Like OLLIE, Stanford's OpenIE frequently double counts because it returns the same triples in multiple different forms.

Stanford OpenIE Extractions		
Row	Sentence	Extracted Triple(s)
1	"Here was a certain expense to be incurred by the purchaser, large or small." (C19 Hansard)	(Here was certain expense) (Here was certain) (Here was expense)
2	"They say that they want cheapness, and to have our industries worked cheaply; but cannot they see that if they put taxation upon our own manufactures which is not put upon foreign manufactures, our own producers cannot compete with the foreigner as thoroughly as if both were taxed equally." (C19 Hansard)	(they put taxation) (they put taxation upon our own) (both were taxed) (both were taxed equally)

Table 3.3: Examples of triples extracted with OpenIE. Each triple is surrounded by parentheses added by the author. Each unit of the triple is separated by multiple spaces.

Three versions of triples are extracted from the sentence in Table 3.3 row 1. "Here was certain expense" contains both triples, whereas "Here was certain" and "Here was expense" represents the same knowledge but broken into two separate triples.

This same problem can be seen in row 2 of Table 3.3. A triple with the subject, verb,

object sequence, “they put taxation,” is extracted along with a version of the same triple with a prepositional phrase appended, “they put taxation upon our own.” Another triple, “both were taxed,” is extracted with the same triple with an adjective appended, “both were taxed equally.” While it can be useful to the analyst to retain more parts-of-speech, OpenIE does not indicate that “both were taxed” and “both were taxed equally” represent the same triple but with or without an adjective, contributing to double counting.

3.2.7 Limitations of Previous Evaluations

Our research follows many of the benchmarking techniques used by previous methods, but we also address three major limitations that we identified in previous triple extraction evaluations:

Limitation 1: No existing method has been evaluated on historical data. Despite that each method has been expressly developed to run on a broad range of textual data–regardless of genre or source–no research has been published on whether these methods are effective on texts from different historical periods.

Limitation 2: Previous evaluations do not describe how extractions from one method were compared with another method if the extractions from both methods are ordered using a different canonical form. The parts-of-speech included within extractions are often different across methods, and without addressing this problem, it is unclear whether this comparison is accurate. Subsequently, these evaluations do not address strategies for dealing with one method that might combine multiple grammatical triples into a single extraction and another method that might separate each triple into its own extraction.

Limitation 3: Previous evaluations do not address the pervasiveness of double counted triples across extraction methods. Double counted triples can be misleading for analysis because they obscure the true amount of times a triple was stated. Methods like OLLIE and OpenIE sometimes extract an excessive number of triples. OpenIE, for example, makes a total of 95 extractions from the following sentence (which only contains 12 triples):

> "In adopting that course, however, the noble Lord has only performed the crowning act of his treatment of my gallant relative, for during his lifetime Lord Raglan was treated by the noble Lord

> in a manner which very few men would have had the temper to put up with, and now the noble Lord takes this opportunity of, by his silence, disparaging his memory." (C19 Hansard)

In the following sentence, OpenIE makes 141 extractions despite there being just two possible grammatical triples, “college is part” and “college is institution.”

> "A Christian college is an education institution or part of an educational institute dedicated to the integration of Christian faith and learning in traditional academic fields." (Wikipedia)

Other sentences of a similar structure might have the same number of extractions as there are triples, causing this issue to be inconsistent.

3.3 posextract

This section introduces posextract, a high-accuracy Open IE system designed for the extraction of grammatical subject-verb-object (SVO) and subject-verb-adjective predicate (SVA) triples from textual corpora. Its design assumes that grammatical triples can be identified using 12 linguistic rules applied to sentences parsed with their syntactic dependencies.

The method introduced by the present research restricts the triples form based on the options selected by the user. Table 3.3 shows the triples extracted from each of the above sentences using the default settings.

As shown by Row 4, posextract’s restricted form is more minimal than existing methods, but still provides a fine-grained analysis of the distinct triples present in a sentence, thus potentially retaining more meaningful information for analysis. We also anticipate that a canonical form will minimize the need for additional cleaning or restructuring after the triples have been extracted. Analysts can instead move straight to visualization.

3.4 Architecture

posextract extracts and outputs triples in seven steps: 1. data import; 2. syntactic dependency parsing and part-of-speech annotation; 3. triples extraction using a depth-first search;

posextract Extractions		
Row	Sentence	Extracted Triple(s)
1	I have not said that the manufacturers do not keep a register." (C19 Hansard)	(Manufacturers not keep register)
2	"Having thrown forth this observation, he would next remark that more blood had been shed in Ireland during the Administration of the right hon." (C19 Hansard)	(he throw observation) (blood shed in Ireland) (blood shed during Administration)
3	"Advertisers would pay for the privilege of having one of their hilarious images in the compilations and thereby fund the work." (Reddit)	(advertiser pay for privilege)
4	"They say that they want cheapness, and to have our industries worked cheaply; but cannot they see that if they put taxation upon our own manufactures which is not put upon foreign manufactures, our own producers cannot compete with the foreigner as thoroughly as if both were taxed equally." (C19 Hansard)	(they want cheapness) (industry work cheaply) (producer not compete with foreigner) (producer not compete thoroughly) (they put taxation) (they put upon manufacture) (producers put upon manufacture) (producers put taxation) (both taxed equally) (which put not put upon (foreign) manufacture)

Table 3.4: Examples of lemmatized triples extracted with posextract without option to combine adjectives and nouns.

4. postprocessing to resolve contractions; 5. postprocessing to resolve redundant triples; 6. postprocessing coreference resolution for subjects; and 7. triples export. posextract can refer to an option configuration file to improve triples for analysis.

3.4.1 Data Import

posextract can be used over the command-line or interactively in Python and R. [22] It is designed to be accessible to many different users, including users who work closely with data

but have little experience writing or running code. Over the command-line, data is imported by simply pointing posextract to a data file. posextract can automatically detect and read different kinds of popular data formats such as plain text, comma-separated variables (csv), tab-separated variables (tsv), or a single character string provided as an argument. In cases where the data has multiple columns, users must specify from which column triples should be extracted.

3.4.2 Syntactic Dependency Parsing and Part-of-Speech Annotation

The imported data is parsed for syntactic dependencies and tagged with parts-of-speech using spaCy NLP, a natural language processing pipeline that annotates words using a predictive language model. Existing methods, like Stanford's OpenIE, also use parsers that are separate from the triples detection and extraction system.

3.4.3 Extraction Algorithm

posextract extracts one triple for every set of SVO/SVA relations in the text. Its design assumes that all triples relations exist in a single sentence, and may exist beyond clausal boundaries. For each sentence, posextract represents the syntactic relationship between words in an adjacency list. In terms of dependency grammar, these relations are referred to as a word's "head" and its "children." posextract identifies the verb and performs a depth first search to identify the parts-of-speech that share a relationship with the verb. Because a single verb might have multiple subjects or objects or multiple adjective predicates, the search continues until all relational words have been exhausted.

Similar to Stanford's OpenIe, posextract follows search patterns based the words' Syntactic dependencies. posextract uses a depth-first search to identify grammatical triples and restricts the search pattern based on the syntactic annotation of the verb. The 12 different rules address main verbs (ROOT), verbs found after conjuncts (conj), verbs found in relative clauses (relcl), clausal complements (ccomp), prepositional complements (pcomp), adverbial clauses (advcl), adnominal clauses (acl), and open clausal complements (xcomp). posextract checks the parent and child nodes of the verb. If the node is a nominal subject or passive nominal subject that shares a direct, syntactic relationship with the verb, then posextract attaches that subject to the verb. Here we define a "direct" syntactic relationship as a rela-

tioship where the the subject is the first neighbor of the verb, or the subject is accessible via a depth first search via any number of verbs between the subject and the verb from which the search began if these verbs share a syntactic relationship with the subject and starting verb and a subject has not already been discovered. If the child of the subject is a negative determiner, it is attached to the subject. posextract recognizes two types of objects: direct object and object of a preposition. It checks whether a preposition, direct object, or adjective modifier share a direct with the verb. Direct object(s) are attached to the verb. Adjective modifiers are if the dependency is a conjunction. If a preposition has a relationship with the verb then posextract checks whether an object of a preposition shares a relationship with the preposition. Similarly posextract checks whether a negative adverb is the child of the verb, or whether a negative determiner is the child of the object or adjective predicate and attaches these parts of speech when relevant. Other relations are considered ungrammatical and are not extracted.

3.4.4 Time Complexity

posextract's time complexity for its graph traversal and extraction process is shown in below:

- N: Number of tokens
- V: Number of verbs
- S: Number of subjects
- J: Number of objects
- D: Number of documents
- R: Number of rules
- : Number of loops

Rule matching:

O(S$*J*$ $*$ R)

$Visitingtheverb$:

$O(V * N) + O(rule_matching)$

$Visitrelatedtokens$:

$O(D * visit_verb)$

$Extract$:

$O(T) * (L)$

3.4.5 Postprocess Triples: Returning Contractions to Individual Word Form

Shown by Table 3.5, extracted triples undergo postprocessing by returning contractions to their individual word form. This additional step is designed to improve the consistency of extracted triples for analysis. Consider the following example:

Contractions Representations		
Row	Sentence	Triple Representation
1	"I did not bike home."	I not bike home
2	"I didn't bike home."	I not bike home

Table 3.5: Examples of How Contractions are broken into Individual Words

Both sentences contain the same knowledge, but are represented differently based on whether "did not" is expressed in full, or abbreviated as the contraction "didn't." By selecting to clean the triples before export, analysts are guaranteed more consistent triples for analysis.

3.4.6 Postprocess Triples: Identifying and Removing Redundant Triples

To minimize double counting, posextract identifies and removes redundant triples by checking whether the indices of the words making up a triples sequence are repeated. We check the indices of the words, not the words themselves, because the same knowledge can be repeated multiple times in a sentence and, in these cases, we wish to retain the other triple(s). If a triples sequence contains the same indices as an already identified triple, the

new triple is discarded. From the following sentence, the extracted subject the "who" is replaced with the subject to which it refers.

> "Then there came Lord Mansfield, who, in addition to the highest name as a lawyer, was himself a Scotchman, and long employed as an advocate in Scotch appeals." (C19 Hansard)
>
> "who employed as advocate" —> "Lord Mansfield employed as advocate"

3.4.7 Triples Export

When using the command-line tool, extracted triples are exported to a comma-separated values (CSV) file. The text file retains any original fields from the original data set, but adds new fields for the whole triple as well as each part-of-speech. Each row contains one triple. In Python, the triples extraction function is applied to a string of text and returns a "triples" object. In R the triples extraction function is applied to and returns a data frame.

3.4.8 Check Optional Config File

posextract checks an optional configuration file to determine whether it should include auxiliary verbs, including verbs indicating future and past tenses. If this is enabled, posextract extracts auxiliary verbs if one is a child to the verb. This configuration file is also used to check whether lemmatization is enabled and whether adjectives should be added to the extraction class of the object. Lastly, this file can be used to specify whether compound subjects should be extracted such as "Indian Government" as opposed to just the noun "Government." By default, compound subjects are extracted.

3.5 Experiment

The present research compares posextract to ReVerb, OLLIE, and OpenIE by evaluating each method on two data sets made up of distinctive sentence grammars: a data set of historical sentences from the 19th-century British Parliamentary debates (also known as "Hansard") and a data set of contemporary sentences from Wikipedia articles that is meant to resemble the data sets commonly used by existing methods.

We chose to evaluate each triples method on historical data to judge their effectiveness for a broader range of textual analysis. Unlike contemporary online speech, British Parliamentary speech has been known to be complex, verbose, and flowery.

In previous studies, the performance of each method was evaluated on contemporary Internet writing, which provides only a limited testing ground for understanding a method's performance or its overall effectiveness for analyzing a broad range of textual corpora. Contemporary online sources, like Wikipedia or The Wall Street Journal, are often composed of short, American-English sentences. They use more declarative sentences (sentences that make a statement) than interrogative sentences (sentences that ask a question). Sources like Wikipedia are often characterized by the prevalence of writing conventions not found in other sources, such as an excess of lists describing items.

3.5.1 Data Selection and Sampling

We evaluate posextract on two different data sets: a contemporary corpus made up of the popular, online sources Wikipedia and Reddit, and a historical corpus made up of sentences from the 19th-century Hansard debates. The different writing conventions of the contemporary American corpus and the historical British corpus test each triples extraction system on a broader range of language constructions.

We use a stratified sampling method to create data sets with varied syntactic constructions. The historical data set is made up of sentences drawn from the 19th-century Hansard corpus. It consists of: 150 sentences drawn at random; 50 sentences with leftward syntactic movement (not due to "wh-" fronting); 50 interrogative sentences (with "wh-" fronting); and 50 compound, complex, and compound-complex sentences. The contemporary data set resembles data used by existing methods, and is made up of sentences from Wikipedia articles. It consists of 150 sentences drawn at random and 50 compound, complex, and compound-complex sentences. We did not create leftward movement or interrogative data sets for the contemporary corpus because these types of sentences were difficult to find on Wikipedia. In total, the evaluation data has 550 sentences with 1,916 possible triples.

We selected these different categories of sentences for a fine-grained analysis of each method's performance when run on different kinds of syntax. While a single sentence can

fall into multiple categories (for instance, an interrogative sentence can also be a complex sentence), we evaluate each sentence solely by the category for which it was drawn. The following sections define each of the four categories.

3.5.2 Sentences Selected at Random

Random sentences are sentences drawn without manual intervention. They may include any sentence type such as simple sentences (e.g. sentences composed of a single, independent clause) to compound or compound-complex sentences (e.g. sentences with multiple, dependent clauses).

3.5.3 Sentences with Leftward Syntactic Movement (Without "Wh-" Fronting)

Leftward syntactic movement occurs when an object or verb precedes a subject. English is considered a rigid "subject-verb-object" language, meaning that within a sentence, the subject usually comes first followed by the verb and then the object. However, sometimes grammatical sentences will feature the verb or object preceding the subject. Consider the following sentence:

"After watching it like 5 or 6 times, I think I see what happened."

In this example, the verb "watching" and the object "it" come before the subject "I." Leftward syntactic movement also occurs when the object comes before the subject, but the verb follows the subject like in standard syntax.

3.5.4 Interrogative Sentences and Sentences with "Wh-" Fronting

Interrogative sentences are sentences that ask a question and are punctuated with a question mark. As mentioned earlier, subjects generally precede the verb or object in an English sentence. An exception is when asking a question. Questions are often structured by "wh-" fronting, where the object is transposed in front of the subject. Consider the following sentence:

"Which sibling was in the image?"

In this example, the object "sibling" comes before the subject "image," the verb "was," and the preposition "in."

3.5.5 Compound Sentences, Complex Sentences, and Compound-Complex Sentences

Compound sentences are sentences where two or more independent clauses are joined together with a coordinating conjunction such as "for" or "yet," or a conjunctive adverb such as "however." Complex sentences are sentences that have one independent clause and one or more subordinate clauses that modify the main subject or verb of a sentence. Compound-complex sentences are sentences that contain at least three clauses: two independent clauses and one or more dependent clauses. In total, a compound-complex sentence can have three or more sets of subject-verb relations. Consider the following sentence:

> "He, however, intended to confine his observations within a narrow compass, not wishing to add one word to what had been so ably stated by his hon. and learned friend (Sir Charles Wetherell), although he did not agree with him entirely on some points."

This example contains independent clauses ("He, however, intended to confine his observations..."; and "he did not agree with him...") and a dependent clause ("not wishing to add one word...").

3.6 Evaluation

The number of distinct triples extracted by existing methods cannot be easily quantified. This is in part because–absent of a strict, canonical form–these methods do not clearly delineate the words that belong to a single triple. Multiple triples can exist in a single extraction. Consider the sentence:

> "Which regiments should we assign to home service, and which should we banish to India?"

This sentence contains four distinct triples: 1. we assign to service, 2. we assign regiments, 3. we banish regiments, and 4. we banish to India. The same knowledge could also be represented by two triples: 1. we assign regiments to service and 2. we banish regiments to india. Both representations of triples are grammatically correct but are structured differently.

Considering the differences between extractors, our evaluation follows these rules to ensure accurate evaluation:

- An extraction is counted as a triple if it contains a grammatical subject, verb, and object or subject, verb, and adjective predicate.
- Only triples are considered by this evaluation. Extractions that are not triples are not considered.
- Two triples are counted as the same if they contain the same knowledge.
- The form of an extracted triple is not considered when judging whether a triple is grammatical.
- If the triples extraction contains part of a compound word, not the whole word, it is still considered grammatical.
- Whether a single triple is extracted more than one time does not determine whether a triple is grammatical or ungrammatical.
- A triple is grammatical if it contains the exact SVO/SVA relationships or the co-referents that stand in for nouns (such as the word "which").
- A triple is considered ungrammatical if it is missing the negating adverb or determiner "not" before the subject, verb, object, or adjective predicate. Similarly, a triple is ungrammatical if it incorrectly includes a negating adverb or determiner.
- Extraneous parts-of-speech included in an extraction (such as qualifiers, articles, adjectives, conjunctions) are not considered when determining whether a triple is grammatical or ungrammatical.
- Prepositions are not considered when determining whether a triple is grammatical.
- If a single extraction from ReVerb, OLLIE, or OpenIE contains multiple triples then each triple is counted separately.

- If an extraction by ReVerb, OLLIE, or OpenIE is ungrammatical but does not contain a subject, verb, object, or adjective predicate, the ungrammatical extraction is not considered during the evaluation. These extractions are considered outside the scope of the current study, which is only interested in SVO/SVA triples extractions.

3.7 Results Validation

Separating the data sets into four syntactic categories provides a more granular look at each method's performance. After running the extractors, the first author of the present study independently evaluated each triple extraction as "grammatical" or "ungrammatical," or "missed." The number of double counted extractions are added together and considered in the overall usefulness of the extraction method for analysis.

The precision and recall of each system's output is measured using a binary classification system that accounts for the number of "grammatical," "ungrammatical," and "missed" extractions. The precision and recall of each method are combined into a single, weighted F-score that determines the overall effectiveness of each method. We placed more weight on "precision" because, we argue, unfactual triples are the greatest risk to analysis–more so than missed triples.

3.7.1 Overall Historical Data Set Results

This section shows the overall performance of each extraction method on the 19th-century Hansard Parliamentary Debates. posextract has consistently higher grammatical triples extractions and lower missed extractions.

3.7.2 Overall Contemporary Data Set Results

This section shows the overall performance of each extraction method on Wikipedia data. posextract has consistently higher grammatical triples extractions and lower missed extractions.

3.7.3 Double Counted Triples

This section shows the number of times a triple was extracted more than once, not the total number of times the same knowledge was extracted. ollie and Stanford's OpenIE double count the most frequently.

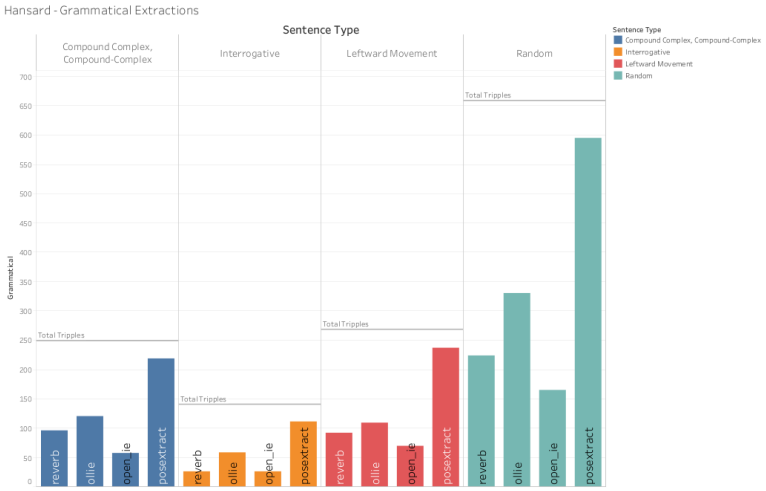


Figure 3.1: Total Number of Grammatical SVO/SVA Triples Extracted from Hansard Per Extraction Method

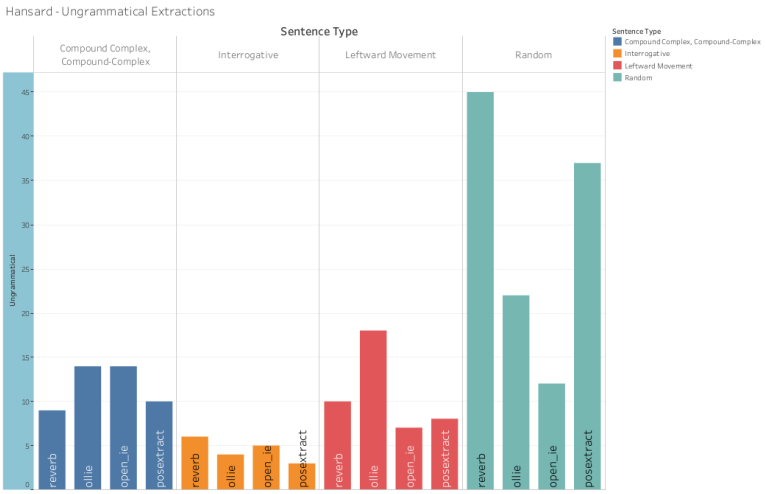


Figure 3.2: Total Number of Ungrammatical SVO/SVA Triples Extracted from Hansard Per Extraction Method

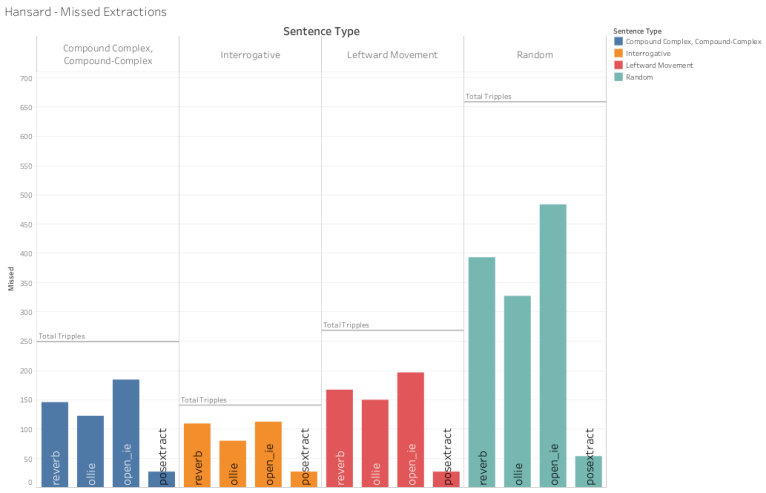


Figure 3.3: Total Number of Missed SVO/SVA Triples Extracted from Hansard Per Extraction Method

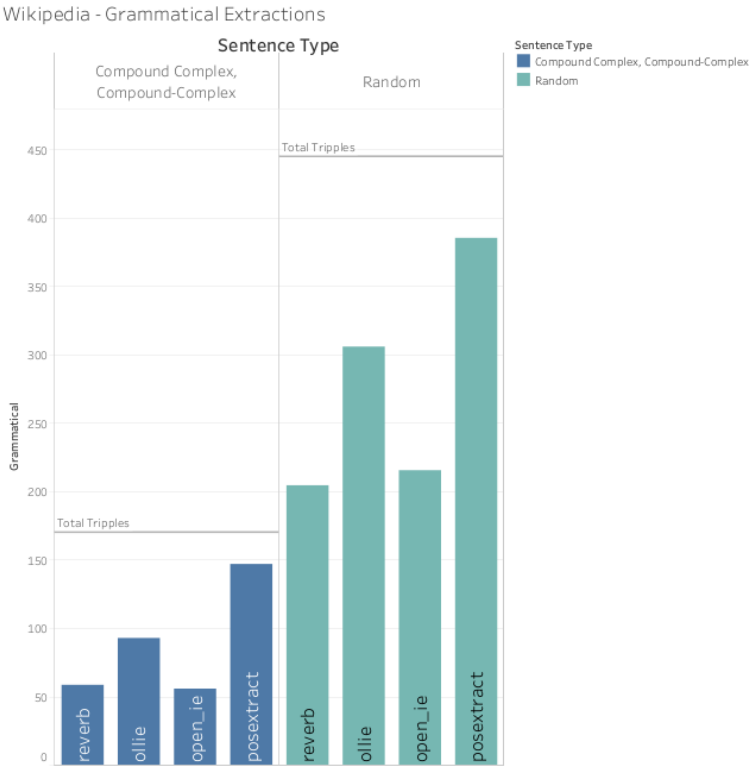


Figure 3.4: Total Number of Grammatical SVO/SVA Triples Extracted from Wikipedia Per Extraction Method

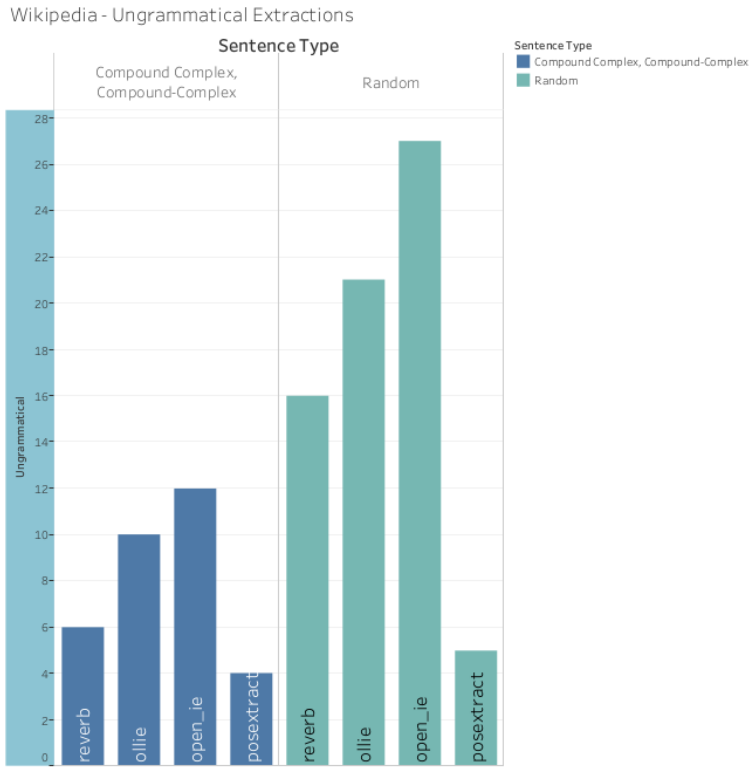


Figure 3.5: Total Number of Ungrammatical SVO/SVA Triples Extracted from Wikipedia Per Extraction Method

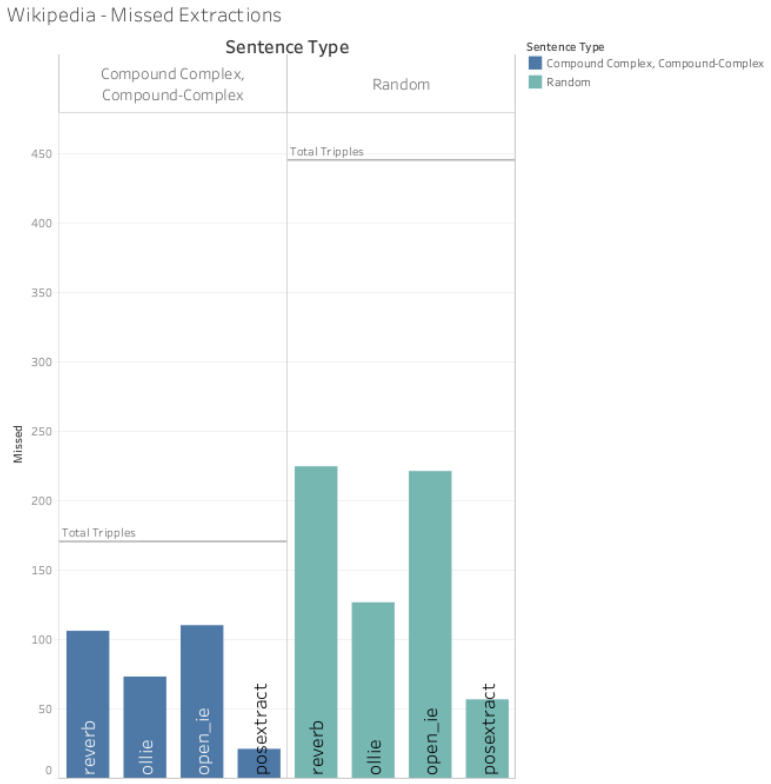


Figure 3.6: Total Number of Missed SVO/SVA Triples Extracted from Wikipedia Per Extraction Method

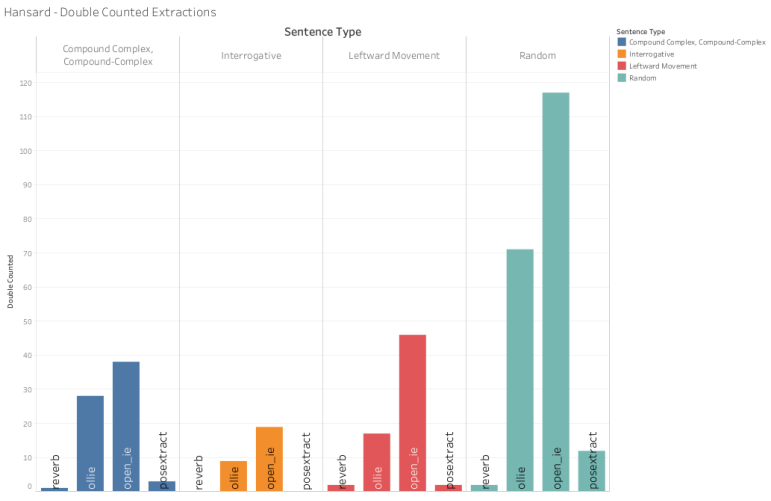


Figure 3.7: Total Number of Times Each Method Extracted the Same Knowledge Two or More Times

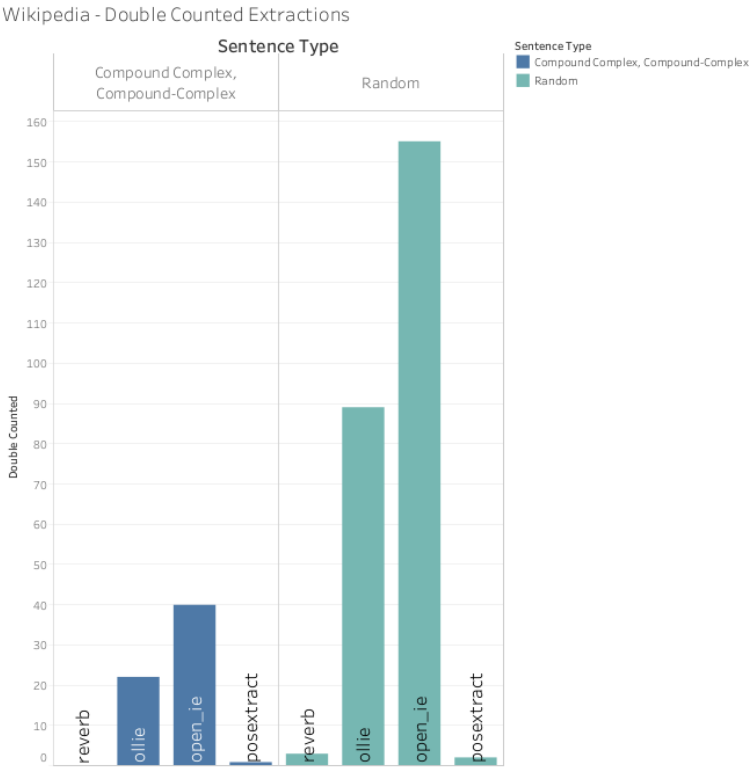


Figure 3.8: Total Number of Times Each Method Extracted the Same Knowledge Two or More Times

3.8 posextract Error Anaysis

This section provides a more in-depth look at the issues related to posextract's performance.

3.8.1 Ungrammatical Triples Extractions

The highest number of ungrammatical triples was extracted from the historical corpus. Ungrammatical extractions are often related to the presence of an indefinite pronoun, like "it." spaCy often tagged indefinite pronouns as subjects in a sentence. In other cases, syntax that is unconventional in contemporary speech was related to mistagged words. For example, this sentence starting with "I."

> "I as it is still, that it would not have been in our power, considering the loss of authority we were continually suffering from, to deal in a satisfactory manner with the difficulties and intricacies of the subject of local taxation and all the consequences it involves." (C19 Hansard)

3.8.2 Missed Triples Extractions

In some cases posextract misses triples. Many missed triples are caused by incorrect tags by spaCy.

For example, from the sentence:

> "The larvae feed on Fleurya capensis, Boehmeria nivea, Australina, Boehmeria, Pouzolzia, and Urtica species." (Wikipedia)

Six triples should be extracted:

larvae feed on Fleurya capensis

larvae feed on Boehmeria nivea

larvae feed on Australina

larvae feed on Boehmeria

larvae feed on Pouzolzia

larvae feed on Urtica

However, the subject "larvae" is tagged as an adjective by spaCy, and the verb "feed" is tagged as the subject.

3.9 Limitations

This section describes the limitations of the present study and the proposed method, posextract, by addressing issues related to resource usage and the evaluation data.

3.9.1 Resource Usage

By taking advantage of modern NLP software, posextract offers meaningful triples for analysis. posextract's memory management allows it to run on many types of computers including older laptops. Syntactic dependency parsing, however, relies heavily on complex statistical modeling that tags words with their part-of-speech. At scale, syntactic dependency parsing can be expensive because it uses a lot of energy/power and can take a substantial amount of time to complete.

3.9.2 Lack of Linguistic Diversity in the Contemporary Corpus

Multiple existing data sets were evaluated solely or predominantly on Wikipedia articles. Therefore, our study, too, uses Wikipedia. Even though Wikipedia articles cover a broad range of topics, they each use very similar sentence structure, including lots of lists. Given their encyclopedic nature, they typically don't include interrogative sentences or sentences with leftward syntactic movement, despite the prevalence of these sentence patterns within contemporary speech. Therefore, their absence of linguistic diversity does not provide deep insight into how a triples extraction method might perform on different kinds of text.

3.10 Conclusion

posextract offers a high-accuracy method for triples extraction that requires no additional training and can be run on an average laptop. Its intervention lies in it being designed for informing textual analysis. The contributions of this study on posextract are as follows:

- We identify and analyze problems related to ungrammatical triples extraction outputs

and show their prevalence for each existing information extraction system.

- We provide an in-depth look at the accuracy of each grammatical triples extractors output based on different kinds of grammatical structures (interrogative, wh-fronting), as opposed to existing papers that treat different grammars indiscriminately.
- We support the growing need for triples extraction for textual analysis by using this on contemporary and historical data.
- We make posextract and the data used in our experiments publicly available.

3.10.1 Future Work

In the future we plan to use posextract to compile a giant knowledge base of grammatical triples from a comprehensive data set of the U.S. Congressional records (1870 - 2022) and the Hansard Parliamentary Debates (1803 - 2022) for distribution among communities in which natural language processing and textual analysis based on linguistic features is central to research.

To increase the accuracy of extracted triples, after their detection we intend to assign a confidence score and only export triples with a high score.

3.11 Acknowledgements

This research was supported in part by NSF grant number 1520103.

Chapter 4

Application: Analyzing the Entities Named in Half a Year of News Articles on COVID-19 to Learn About About Media's Selective Attention to Information and Authority

4.1 Introduction

The COVID-19 pandemic is a public health crisis that has affected the lives of people around the world. In response to this crisis, the United States government implemented policies aimed at lowering hospitalization rates and mitigating serious illness or death, many measures of which have been unprecedented in living memory. [23] Early in the pandemic these policies included large-scale interventions such as mandating business and school go online. Mask mandates were enforced to decrease community spread even as businesses returned to normal capacity. In some communities these measures were more or less welcomed, but in others they evoked anxiety towards authority and elicited a greater fear not for public health, but for individual freedoms. [24] Now, over a year and a half into the pandemic the nation is still heavily divided about which policies are (or were ever) warranted, and to what extent medical or governmental institutions should define them.

Research has shown that where communities receive information about the COVID-19 pandemic affects members' beliefs about the nature of the crisis and their attitudes towards policies. [25] News media have been a main conduit for information about COVID-19, but information is not spread evenly, nor is shared information always shared impartially. [26] [27] Many major news outlets emphasize or downplay aspects of the pandemic depending on their perceived partisan preferences. [28] Their selectivity in the information covered contributes to the mixed beliefs held by the public where some perceive COVID-19 policies as suspect–having been heavily mediated by government dogma and political lies–and others perceive policies as non-partisan and meant to support no other agenda but health and safety. [29]

While research has shown that exposure to different media realities contributes to the public's mixed attitudes towards COVID-19 policies, less work has been done to explore how a news outlet's selectivity in information coverage manifests in the actual language of its articles. [30] This study suggests that a news outlet's bias can be seen in the major entities it associates with the pandemic and the amount of attention it gives to them. To make this point, this study analyzes half a year of COVID-19 news articles from two prominent news outlets with disparate readership–NPR and FOX News–and uses a predictive named entity recognition system to identify major organizations, people, and events. It assumes that the amount of attention given to entities can be measured by the proportion in which an entity is mentioned compared with other entities of its category. To gain a better understanding of how these entities are imagined by each news outlet, this study applies triples analysis to look at the verbs assigned to entities given the most attention.

This study notes a stark difference in the amount of attention each outlet gives to major entities associated with the pandemic. NPR, for example, shows a preference for publishing articles that describe COVID-19 in relation to health organizations and infectious disease experts, whereas FOX News shows a preference for publishing articles that describe COVID-19 in relation to politicians and government legislation. It can be seen that many major entities in FOX News are described as performing actions with more emotionally encoded verbs, creating an emotionally charged worldview for its audience.

Subsections are in normalfont. Subsections are numbered and appear in the Table of Contents.

4.2 Background

The COVID-19 pandemic has been described as both a health crisis and a crisis of communication. [31] Early in the pandemic, during his presidential campaign, Donald Trump described Democrats as "politicizing" COVID-19, and his response to the pandemic as their "new hoax". [32] Perceptions about the nature of the virus were muddied by the reality that COVID-19 was a novel virus and new, abstract, or indiffinitive information about the nature of COVID-19 was shared daily. [33] Policies aimed at mitigating the spread of COVID-19 were introduced, vetted, and then later rewritten. The World Health Organization (WHO)

called this period an "infodemic"–a crisis in which too much information–including false or misleading information–has saturated online spaces and has resulted in "confusion" and "mistrust towards authorities." [34]

In some cases, the public's disparate attitudes towards COVID-19 policies were attributed to partisanship and a desire to match one's own beliefs with the leaders of their political parties. Indeed, partisanship became a greater indicator of whether communities practiced mitigation efforts–like social distancing–than factors such as a county's COVID-19 case rate or its population density. [35] But less research has been done to explore how the very language of the information consumed by the public led some to experience COVID-19 policies as partisan to begin with, while others experienced COVID-19 policies as non-partisan and meant for no other purpose than to mitigate a public health crisis. Where communities receive information about COVID-19 affects attitudes towards policies and beliefs about the nature of the pandemic. News outlets were, and continue to be, a major source of information about COVID-19, and even a predictor for one's preference for vaccination. [36] Conservatives' trust in medical science, for example, decreased since the start of the pandemic, and county-level consumption of FOX News was related to reduced adherence to various recommendations by the Center of Disease Control and Prevention (CDC). [37]

The present research explores how an outlet's selective coverage manifests in the amount of attention given to different organizations, people, and events. [38] As the present research suggests, the major entities most associated with the pandemic vary substantially depending on the source. For some, organizations like the WHO, and once obscure infectious diseases experts–like Dr. Fauci–became household names. Others developed a greater sense of distrust towards the medical profession. This outcome may be associated with the way in which major entities are imagined by each outlet, where entities from FOX News are described in more emotionally charged ways.

4.3 Method and Results

4.3.1 Data

The data for this study was scraped from online articles about COVID-19 published by NPR and FOX News over a six month period, from the start of August 2021 through De-

cember 2021. The data totals 950 articles. NPR and FOX News were chosen because of their popularity across the nation and their appeal to different readerbases. 75% of conservative identified citizens report trusting FOX News, but 77% of liberal identified citizens report distrusting it. [37] Numerous studies have shown that FOX News has even influenced how people respond to COVID-19. [39] NPR tends to be received by the public as a more centrist or left-of-center outlet. [37]

To limit the study to stories that focused on COVID-19, the data set only includes articles whose URL headlines mentioned the pandemic. The article text was scraped from the unique URLs listed on NPR's "archive" page, and from the unique URLs listed on FOX News's "lifestyle," "us," "opinion," and "politics" pages if they contained the word "covid," "coronavirus," "vaccine," "omicron," "delta," "vaccinat," "vaxx," "pandemic, or "mask." In total, article text was scraped from 402 articles from NPR, and 548 articles from FOX News.

4.3.2 Named Entities Detection

Named entities were detected from article text labeled with the "paragraph" HTML anchor. This study is only interested in the named entities mentioned in the article narrative, so text labeled with different anchors–such as headlines or subtitles–was ignored. In some cases text that did not belong to the article narrative, like caption text, was still labeled as part of a paragraph and named entities were extracted from it. Therefore, manual post-processing was done to remove entities from caption text, such as the names of reporters, editors, image credits, or the names of sponsors if they were stated frequently.

The remaining named entities underwent superficial cleaning. Leading and trailing white space was removed, as well as leading articles and possessive inflection markers. This process transformed named entities like "The FDA" to just "FDA," and possessives like "Fauci's" to "Fauci," to reduce the number of names that were separated due to trivial differences.

After cleaning, 27,963 person entities, 38,703 organization entities and 10,109 event entities were kept from NPR. 29,818 person entities, 29,039 organization entities, and 4,932 event entities were kept from FOX News.

4.3.3 Analysis

Analyzing named entities and extracting triples shows how, depending on the media coverage, different organizations, people, and events were given more attention by NPR or FOX News, and that the kind of attention given to major entities might be more or less emotionally charged.

Attention is measured by the proportion in which an entity is stated compared with others of its category (e.g. the number of times a person name was mentioned is measured against the total number of person names, not the total number of entities overall). To limit the study to the most relevant named entities, this study analyzes just the top 30 from each category. Depending on the category, the top 30 names made up anywhere from 34

To detect how major entities (names like "Biden") are described by each outlet, dependent parts-of-speech are extracted using a triples extractor. [22] The verbs assigned to major entities are tagged with their guessed sentiment.

4.3.4 Named Organizations

Organizations were sorted into four sub-categories representing the major groups affiliated with the pandemic: "Health Organizations," which include disease control agencies, vaccine manufacturers, and research institutions; "Political Organizations," which include partisan organizations like political parties; "Enforcement," which includes organizations that enforce lawful order such as OSHA, military, or paramilitary organizations; and "Other" for any organization that is mentioned in COVID-19 articles, but does not fit within the other three categories. Sorting organizations this way allows a look into how much attention is given to the different groups that have had prominent roles within the pandemic, but may play different roles in solving issues related to COVID-19.

Figure 4.1 shows the top 30 organizations mentioned in NPR's articles by proportion. The top 30 organizations make up 52.1% of the total organizations detected across NPR news articles. NPR shows a preference for mentioning the names of health organizations, which include: the Center for Disease Control and Prevention (CDC), the Food and Drug Administration (FDA), the World Health Organization (WHO), the National Institute of Health (NIH), as well as vaccine manufacturers like Pfizer, Moderna, Johnson & Johnson,

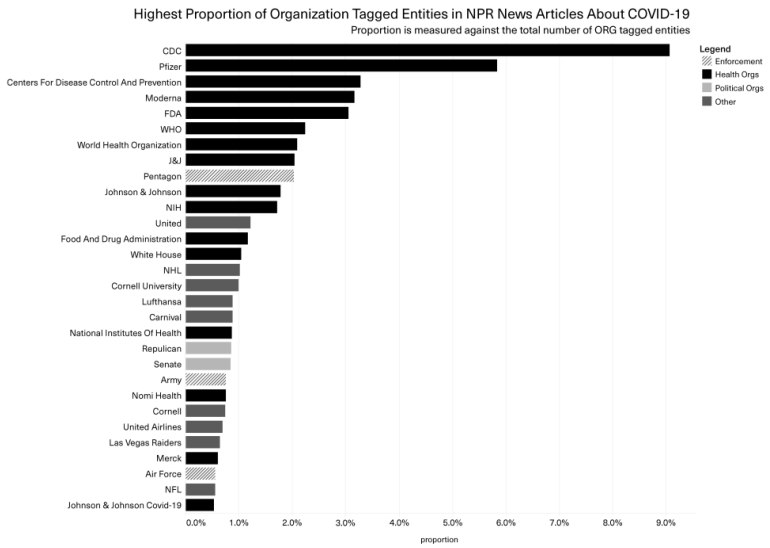


Figure 4.1: Top Organizations Measured by Proportion in NPR News Articles About COVID-19

and Merck. Legislative bodies and partisan groups are addressed, but relatively less attention is given to them compared to health organizations. Only 4.8% of the top 30 organizations are partisan in nature.

Contrasting NPR's and FOX News's top organizations underscores FOX News's preference for mentioning COVID-19 among partisan groups and NPR's preference for mentioning health organizations. Figure 4.2 shows the top 30 organizations mentioned in FOX News articles by proportion, which make up 34.1% of the total organizations detected across its articles. Alone, the White House–the top named entity–makes up 5.5% of all organizations. Health organizations such as the CDC and FDA are included in the top, but political organizations have the highest proportion of mentions overall. Political organizations make up nearly half of the top 30 organizations mentioned at 48.7% (compare this with NPR's 4.8% for politicians), whereas health organizations make up 36.2% of the top 30 named entities. As shown by Figure 4.2, "Democrats" and "Republicans" are frequently mentioned across FOX News articles on COVID-19, along with mentions of the legislative branch of the government (Congress and its chambers, the House and the Senate).

Along with giving a relatively greater amount of attention to political organizations, FOX

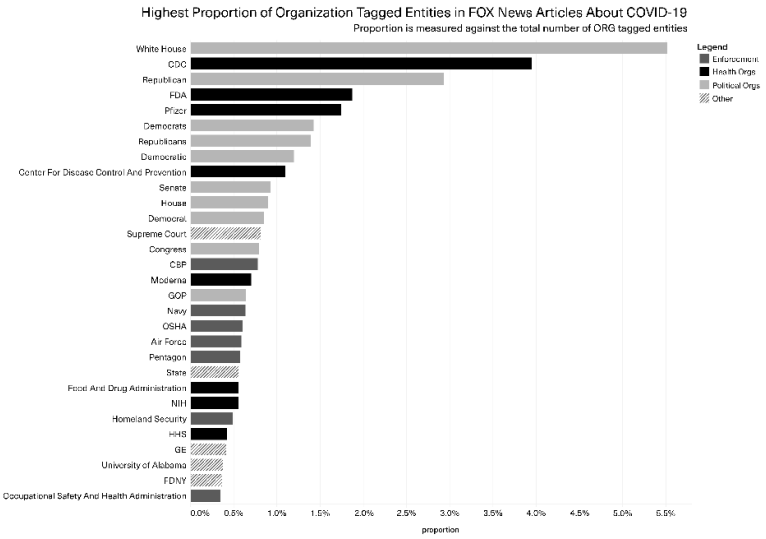


Figure 4.2: Top Organizations Measured by Proportion in FOX News Articles About COVID-19

News also focuses more on organizations related to enforcement and regulation. The Center for Border Patrol (CBP) and Homeland Security make the top most mentions, whereas the WHO–the leading health agency of the United Nations that former Republican president Donald Trump petitioned to withdraw–is not in the top most organizations. The names of vaccine manufacturers are deemphasized overall, with only Pfizer and Moderna appearing in the top.

The amount of meaningful attention given to health organizations is important considering the prevalence of misconceptions about these different health organizations. Surveys have shown, for instance, that Democrats are more likely to trust the WHO than Republicans.

4.3.5 Person Names

Person names were sorted into three categories that represent a person's vocation in relation to the pandemic: "Infectious Diseases Expert," "Politician," and "Other." As the sorting implies, the names of medical professionals who are not infectious diseases experts are categorized as "Other." Comparing the types of people mentioned by NPR and FOX News shows a difference in their preferences for mentioning infectious diseases experts or politicians. NPR has nearly 12 times more mentions of infectious diseases experts in its top

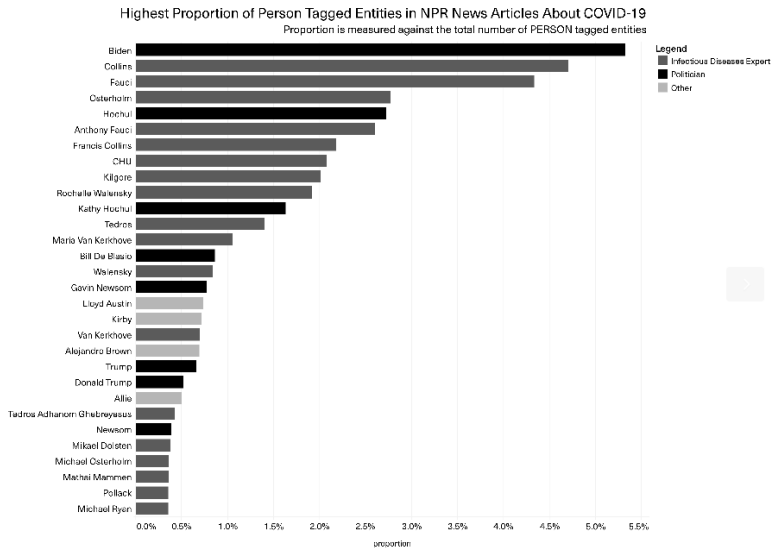


Figure 4.3: Top Person Names Measured by Proportion in NPR News Articles About COVID-19

mentions than FOX News, and FOX News has nearly 14 times more politicians in its top mentions than NPR. These observations are consistent with NPR's demonstrated preference to mention health organizations and FOX's demonstrated preference to mention political organizations in relation to the COVID-19 pandemic.

Figure **??** shows the top 30 person names from NPR, which make up 44.44% of the total person names identified. NPR frequently mentions high profile names in the field of infectious diseases medicine including Dr. Francis Collins, director of the NIH; Dr. Anthony Fauci, director of the National Institute of Allergy and Infectious Diseases and Chief Medical Advisor to the president; Rochelle Walensky, director of the CDC; and Tedros Adhanom Ghebreyesus, Director-General of the WHO. Contagious diseases experts make up 65% of the top 30 names. In total, few politicians appear at the top.

Figure **??** identifies the top 30 names from FOX News, which makes up 39.23% of the total person named entities. With 85.67% of the top 30 person names belonging to politicians, FOX News minimizes the relationship of contagious diseases experts with COVID-19 while giving more attention to politicians. "Biden," referring to Democratic President Joe Biden, is given the single most attention, receiving nearly 17.5% of the total person names mentioned.

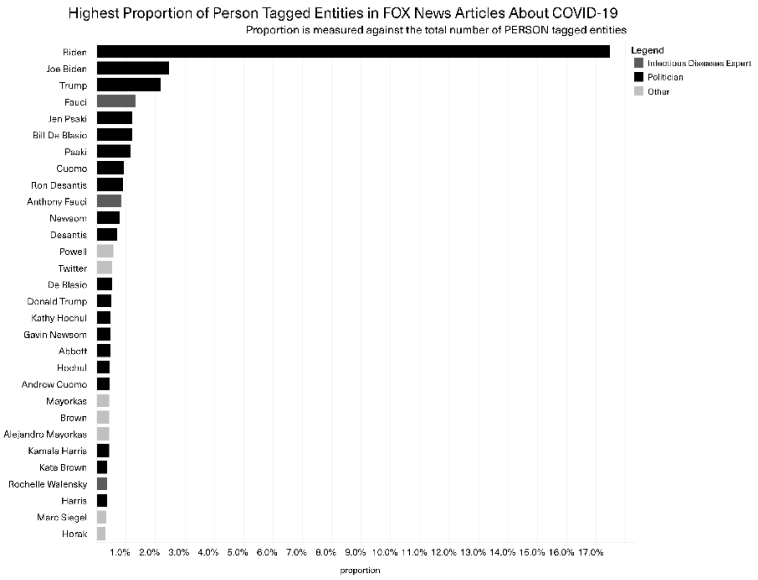


Figure 4.4: Top Person Names Measured by Proportion in FOX News Articles About COVID-19

Other politicians mentioned in FOX News articles include former Democratic mayor of New York, Bill De Blasio; former Democratic governor of New York, Andrew Cuomo; Republican governor of Florida, Ron Desantis; Democratic governor of California, Gavin Newsom; Democratic governor Kathy Hochul; and Republican governor of Texas, Greg Abbott. The only infectious diseases experts in the top most people mentioned in articles about COVID-19 are Dr. Anthony Fauci and Dr. Rochelle Walensky and they only make up 8.9% of the top 30 mentioned names.

Between the two outlets, there is a stark difference in the amount of attention given to contagious diseases experts or politicians. This difference may contribute to the publics' different perceptions on whether COVID-19 is predominantly a partisan or nonpartisan problem.

4.3.6 Named Events

Events are sorted into four categories representing major moments during the pandemic. These categories are: "Pandemic," for events that refer to the pandemic itself; "International Event," for events that take place internationally, even if this event has been known to

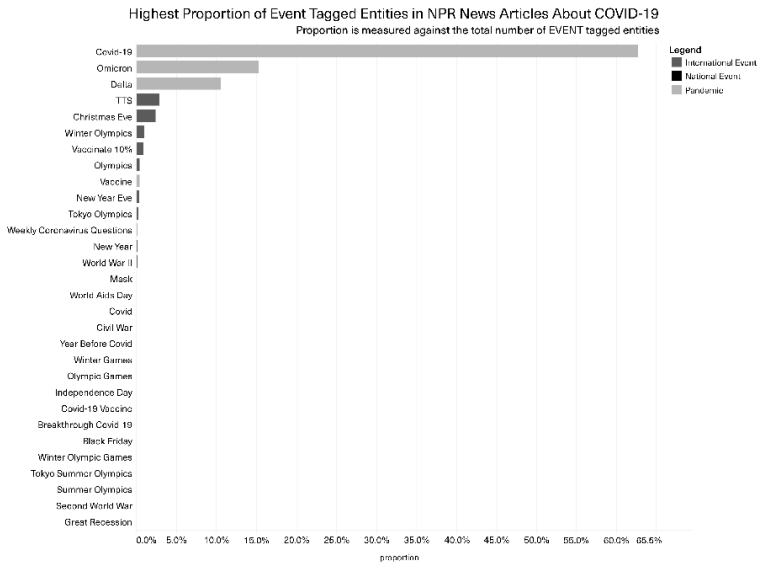


Figure 4.5: Top Events Measured by Proportion in NPR News Articles About COVID-19

occur nationally; "National Event," for names of events located in the United States; and "Enforcement," for a subclass of national or international events where authoritative powers have extended themselves to influence an outcome.

In both NPR and FOX News, the pandemic itself is referred to many times over the events belonging to other categories. This suggests that unlike as with organizations, the events associated with the pandemic are transitory and change with time. Compared with one-another, NPR shows a preference for mentioning events that are non-partisan and do not relate to legislation, whereas FOX News shows a preference for mentioning acts of legislation and enforcement.

Figure **??** shows the top 30 events mentioned in NPR, which make up 98.89% of the total events identified. The pandemic itself is given the most attention (e.g. Covid-19, Omicron, and Delta), making up 89.6% of the top 30 events mentioned, and 88.6% of the total events mentioned overall. After mentions of the pandemic itself, NPR showed a preference for reporting on international events such as holidays, memorials, war, and the Olympics. None of NPR's top events are part of the "Enforcement" category, which would include mandates or acts of law enforcement. While NPR does report on mandates, any single mandate receives

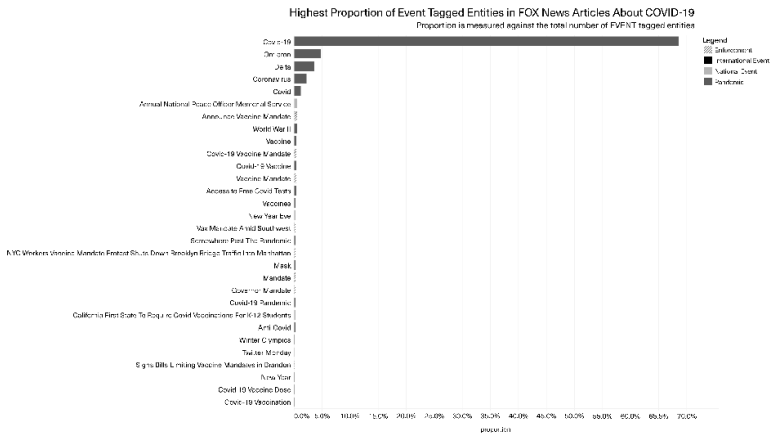


Figure 4.6: Top Events Measured by Proportion in FOX News Articles About COVID-19

too little attention to make its list of top events.

FOX News also gives the most attention to the pandemic. Figure **??** shows the top 30 events mentioned, which make up 88.24% of the total events identified. The pandemic itself (Covid-19, Omicron, Delta, Coronavirus, and Covid) makes up 80.53% of the total events mentioned, and 91.27% of the top 30 events. After the pandemic, FOX News focuses on entities in the "Enforcement" category such as vaccine and mask mandates. This orientation is different from NPR, which instead relates COVID-19 back to the threat of group spread by focusing on events with gatherings–such as the holidays or the Olympics. Readers are exposed to problems associated with travel and movement–to the shared spaces we inhabit and to the number of people we may encounter. FOX News, on the other hand, gives relatively more attention to moments where authoritative bodies exercised their perceived power. Compared to one-another, each outlet paints a disparate picture of the pandemic.

These differences also manifest in the way in which the increased availability of vaccines is depicted. NPR repeatedly mentions Biden's pledge to vaccinate 100 million Americans in 100 days (e.g. "Vaccinate 10%"), which was intended and received as a message of hope among many Americans waiting for the vaccine. FOX News on the other hand describes the increased accessibility of vaccines as co-occurring with enforcement, harkening back to the idea that governmental forces might exert their powers over unconsenting members of the public.

4.4 Contextualizing Named Entities Using Triples Analysis

Throughout different moments in the pandemic, news networks across the nation struggled to characterize the nature of COVID-19. The purpose of this section isn't to show that an outlet was more or less wrong in their descriptions of COVID-19, but to instead observe how actors related to the pandemic were described, and how these descriptions may contribute to community responses to the pandemic and its policies. This section does this by using triples extraction to analyze the actions performed by subjects, and then annotating the identified verbs with their guessed sentiment. [40] Extracting and annotating triples shows that FOX News's characterizations of the actors who were given the most attention in COVID-19 articles has a particular, emotional charge that is absent in articles published by NPR.

It has often been suggested that the problem with opinionated sources like news media is how they act like echo chambers and information silos, repeating information and indoctrinating its followers. But just an exposure to repeated ideas may not be at the core of why divisive worldviews occur. Any brand attempts to repeat itself, using distinct patterns (or language patterns) to create a product. Instead, this study looks at the emotional intensity of the repeated content and makes the assumption that understanding the repeated emotions may be important for understanding the different community responses to COVID-19. This section shows that, in a sense, we are literally talking past one-another by assigning different emotional realities to national responses to COVID-19.

Within these articles, both NPR and FOX News use emotionally encoded verbs at about the same rate, where 12 of the total verbs mentioned are either positive or negative. The distinction between how these news outlets leverage emotion occur instead among different subgroups.

Considering just the top 30 person tagged entities, 7% of the verbs used in FOX News articles communicate negative emotions, whereas 2% of the top 30 person tagged entities in NPR articles are negative. This suggests that these outlets offer distinct, emotionally charged understandings of the actors related to COVID-19. In FOX News articles, Joe Biden is described as acting in negatively emotionally charged ways. He is described as having "touted," "accused," "punished," "undermined," "attached," "flunked," "argued," and

"violated." Almost no negatively emotionally charged words are assigned to Joe Biden by NPR.

Organizations are also described with a distinct emotional flavor. As a subject, FOX News assigns the word "government" authoritarian words like "bans," "blames," "issues," "tells," and "requires." NPR, on the other hand, uses "mandate," and many words that are less authoritarian such as "consider," "tighten," "not restrict (gatherings)," or "authorized" as in authorizing others or making room for others to make decisions instead of making decisions as a monolithic institution.

These differences may be seen in the way both news outlets create narratives and distinct identities that reinforce a party divide and favor different emotions. FOX News depicts the Republican party as performing actions that express anger and confrontation, suggesting that these verbs are essential to having a position of power. Republicans are described as having "exploded," "demanded," "pressed," "expressed," "pushed," and "exerted (power)," to name just a few. When performing a confrontation, Republicans "strode," "reported," and in fostering a polarized community, Republicans "sided." The verbs associated with the subject Democrat(s) include "declare," "loose," "lack," and "spend."

These descriptions of Republicans and Democrats are different from NPR's, which assigns no emotionally charged verbs to Democrat(s) and few to Republican(s). The words assigned to Republicans include "grown (hostile)," "dominate (legislature)," "parlayed" (as in making bets), and "dying."

Both outlets describe Republicans as expressing anger and domination. A major difference is whether this expression is good or bad, an expression of power and control or of recklessness.

4.5 Implications for the Future

This study validates the claim that COVID-19 is not only a public health crisis, but also a crisis of communication. While this study only considered articles from NPR and FOX News, it does suggest how an "infodemic" can arise through the amount of attention a news outlet gives to different organizations, events, and people. Taken together, these findings illuminate the language trends within the news coverage that has been shown to

influence trust in medical science. [37] [41] Republicans are most likely to hold misperceptions about vaccination, and vaccine misinformation is associated with lower vaccination rates and higher vaccine resistance. [42] A unified message to the public about COVID-19 prevention is important because the extent to which experts can deliver a trustworthy message about the virus yields substantial differences in health outcomes.

Chapter 5

Pedagogy:Text Mining for Historical Analysis

5.1 Book Proposal and Manuscript

5.2 Rationale

The field of digital history, where computational tools are used to make discoveries about historical time, has recently had a number of breakthroughs: the first monograph-length study to use text mining (Luke Blaxill's War of Words), the first monograph-length study to use digital mapping (Cameron Blevins' Paper Trails), the first book-length meditation on the challenges of web-borne archives (Ian Milligan, History in the Age of Abundance). Indeed, one of us recently delivered the final manuscript for what we believe is the first book-length meditation on the challenges of adapting statistics and machine learning to the problems of the historical method (Jo Guldi, The Dangerous Art of Text Mining). Unfortunately, most students of history face an enormous barrier to entry when they read about and try to get started work of this kind: none of the major textbooks on digital history gives a hands-on, how-to guide to using code to analyzing text from the past. Our proposed book, Text Mining for Historical Analysis, will fill this gap.

As The Dangerous Art of Text Mining has made clear, the tools of historical analysis are crucial to making sense of how language changes over time. Without this crucial tool for analysis, many otherwise lucid examinations of culture lack an understanding of the events and moments responsible for shaping modern people and institutions. We wrote this book because we believe that a data science of textual analysis has little to contribute to our understanding of politics, the economy, society, gender, race, or class unless textual analysis can profit from the tools for understanding historical change – a process that our textbook is unique in offering.

Our textbook has a pragmatic orientation to critical thinking about how texts can be

modeled to show change over time, one that pairs approaches from the humanities alongside approaches from statistics and machine learning. In each chapter of the book that follows, researchers will be led through a small sample of tools available to computer programmers – whether the basic tool of counting words (Chapter 1-2) or more sophisticated approaches. In Chapters 3-4, the textbook offers an introduction to some basic tools from twentieth-century statistics, putting the reader in the position to grow curious about the consequences of choosing one statistical equation for representing difference over time over another equation that models similar purposes in a different way. In Chapters 5, the textbook offers an introduction to two important machine learning approaches, that is, programs that effectively identify grammatical structures within language as well as major patterns of language use. Each chapter offers opportunities for reflection on the historical insights to be gained from analyzing text with quantitative approaches. Each chapter also highlights opportunities for critical reflection on the algorithms and models commonly used in a digital history process and the biases or limits of each algorithm and reasons why multiple approaches may be necessary to achieve confidence in the meaning of one's insights.

Text Mining for Historical Analysis was developed as the textbook for Jo Guldi's undergraduate course in text mining, which counts for credit in three programs at Southern Methodist University: History, Data Science, and Computer Science. A series of 5 chapters are designed to take the novice student of code through a learning experience, from installing the programming language R on a computer, to loading appropriate software, to instructions on "running" lines of code for first-time use, to creating a first visualization of word count, to plotting word count over time, to chapters on making intelligent choices around the selection of algorithms and statistics. Following the format of most code-based textbooks, Text Mining for Historical Analysis offers a series of exercises in code accompanied by markup language describing the purpose of the code. Like many of the existing textbooks that serve digital humanists and students of data science, Text Mining for Historical Analysis is written in the R language, which is currently the dominant language taught at Stanford's department of Data Science, the major language for courses in the Digital Humanities taught by Andrew Piper and Lauren Tilton, and the chief language used in most departments of Statistics, Economics, or Sociology.

Most coding textbooks offer a preamble for students new to the programming language, meaning that the textbook will work for both novices, unused to programming at any level, as well as more experienced coders, who can turn to later chapters to gain insight on specific subjects. Coding textbooks typically mingle "snippets" of code, usually in a courier-type font offset from the main text, with "markup" text, usually in a Times-type font, which explains the purpose of each chapter, introduces concepts in code (for instance, in an introductory lesson, the concepts "function" or "variable", which are loosely speaking the verbs and nouns of code), and any special instructions or pedagogical activities the text recommends (for example, changing a parameter of a bar-chart visualization so that the computer is instructed to display the counts of the most common 20 words instead of just 10). We plan to deliver our book manuscript in "markdown" format, a standardized R export format that allows scholars to deliver a manuscript of English-language prose wherein code snippets, headings, tables, and illustrations are packaged in a standardized format that can be exported as a Word Document, HTML, or pdf file. For samples of code textbooks produced in this easy-to-read and extremely standardized format, see https://bookdown.org/.

Text Mining for Historical Analysis introduces important aspects of data science that will be of interest to analysts from many fields: the problems associated with tracking language over time. Serious discussions of the best practices, techniques, and problems associated with historical analysis has been missing in most textbooks associated with the digital humanities (Jockers, Silge and Robinson, and Arnold and Tilton) and they are only available in their most basic lineaments in the standard texts from Computer Science (Loper, Sarkar). The crucial element of the study of change over time – which is the basis for meaningful findings in the discipline of History – is otherwise missing from existing textbooks in the field of the digital humanities. The absence of an approach for understanding how words' usage changes poses one important barrier to more historians and social scientists using the methodologies described in The Dangerous Art of Text Mining. By explaining, in simple exercises, approaches to measuring word count over time – including the use of proportion rather than raw count and the treatment of date-type data in R – the textbook puts historical analysis at last within the grasp of a novice coder. Our textbook's method is ideally suited to instruction in a humanities or social science framework that emphasizes the value

of skepticism regarding "black box" approaches to modeling human societies. By emphasizing more than simple wordcount – especially the importance of tracing back algorithmic analyses to the original instance of words in their context for interpretive purposes – Text Mining for Historical Analysis exemplifies the tenets of "critical search," Guldi's term for the paradigm of analysis that insists on performing strategic reading and implementing multiple methods of measurement. In critical search, the analyst's attention is called to the fact that any search through text illuminates various dimensions of a corpus while obfuscating others, and critical searching can enhance the analyst's sense of the text measured by considering how certain vantage points share different information. The critical paradigm of analysis encourages analysts to use quantitative measurements side-by-side with close reading techniques. In our textbook, each exercise encourages the student of code to pause and reflect on the significance or lack thereof of their results, the bias of the data, and any bias attached to the algorithms consulted. Contextual, historical knowledge should guide assessment of the linguistic trends belonging to social structures like governmental or political ideology. The algorithms and statistics investigated by the book follow closely to the text of The Dangerous Art of Text Mining, offering an ideal pairing for a course on data science in the social sciences. An appendix gives a sample syllabus that outlines a short course of nine weeks where chapters of The Dangerous Art of Text Mining are accompanied by an exercise in code from Text Mining for Historical Analysis.

We have ensured that our textbook's examples remain accessible and relevant by following the standard procedures for a software package in R. To accompany this book, we have developed two software packages in R: hansardr (https://github.com/stephbuon/hansardr) and dhmeasures (https://github.com/stephbuon/dhmeasures). The package hansardr provides an easy way to access the Hansard debates within the R environment. After its installation, users can simply import sections of the corpus using the familiar 'data()' command. We developed this package as a way to make accessible the parsed debates from "The Hansard 19th-Century British Parliamentary Debates with Improved Speaker Names," which is an analysis ready version of the 19th-century Hansard corpus, and the only known version with disambiguated speakers. The process of collecting and cleaning the data, a process that includes identifying debates that are marked as "missing" from UK Parliament's version of

Hansard, and disambiguating 85% of speaker names is outlined in two forthcoming articles. The package dhmeasures provides functions that have been designed for wrangling and measuring change over time for historical analysis. This package, therefore, offers practical and adaptable code that can be used in other contexts in which historical analysis might be insightful. Because of our previous investment in software package infrastructure designed to be compatible with future system upgrades to the R language, we can offer coding exercises with the full confidence that the data and tools we teach will remain viable strategies for researchers and teachers for years to come.

5.3 Comparable Titles

How-to textbooks that teach the code behind scholarship are commonplace in fields such as computer science and economics, where code is the major method for opening new doors on analysis. Many code-based textbooks go into new editions every year and have high rates of sales, sometimes for longer than a decade; Edward Loper et al.'s Natural Language Processing with Python (2009) is still regarded by some as a perfect introductory text. In digital literary study, examples of code textbooks include Matt Jockers' Text Analysis with R for Students of Literature (2014). Wide-ranging surveys of code-based practices in the digital humanities include Taylor Arnold and Lauren Tilton, Exploring Humanities Data in R: Exploring Networks, Geospatial Data, Images and Texts (Springer 2015). The field of historical analysis, however, lacks such a code-based textbook. Text Mining for Historical Analysis was written to fill this gap, with the aim of accompanying Guldi's theoretical and descriptive investigation into the use of text mining to investigate historical change, The Dangerous Art of Text Mining (forthcoming from Cambridge). Some digital humanists have taken the approach of offering web-based how-to guides that are published in non-book rather than book form, and readers of this proposal may be curious about why we deem this manuscript appropriate to produce as a published book. From our own experience and from the precedent of other fields, we judge that well-organized textbooks offer an important genre supporting soup-to-nuts learning in a simple how-to guide. We also believe that the form of the book – available in both paper and online simultaneously – is essential to making learning routine and accessible. A person learning to code often benefits from having a

hard-copy version of the textbook on the desk to follow along with through a lesson, often accompanied by a (with O'Reilly, typically open-source; with Springer, sometimes as an ebook) online transcript of the text, which makes keyword searching a line of code easy. At the same time, many students of computer science find that it is easy to get lost between a window showing the student's work and a browser. We infer that paper copies of coding textbooks are necessary even when an open-source copy of the textbook is available on the web, a fact that explains the high rates of sales associated with many coding textbooks that have gone on to widespread popular uptake.

We can offer our own experience with Julia Silge and David Robinson's Tidy Text Mining with R (O'Reilly) as a specific illustration of how useful it is to have a hard-copy textbook while studying code. As Jo Guldi revisited coding for the first time since her childhood in 2018-19, she worked through a series of exercises from Silge and Robinson with the hard copy of the book next to her screen, inputting each line of code according to the instructions in the textbook. As her coding abilities developed, she put aside the paper copy but returned to the online textbook, hosted for free at O'Reilly's site, for reference. Silge and Robinson became the standard textbook for her course when she began teaching text mining in R, at least until she began to develop her own materials, more carefully tailored to problems of historical analysis – the result of which is the proposed manuscript. A list of other code-based textbooks for the Digital Humanities and Digital Social Sciences include:

- Title: Text Analysis with R: For Students of Literature
- Author(s): Matthew L Jockers and Rosamond Thalken
- Publisher: Springer
- Year: 2014
- [43] provides an introduction to R and focuses on its application to popular literary works, like Hamlet. Jockers offers chapters on subjects such as counting word frequency, clustering using Euclidean distance, topic modeling, and support vector machines (which are a set of supervised machine-learning methods used for classification, regression, and outliers detection). Many of these methods are considered "black

box" (in a sense that users cannot see the operations of the algorithm) and could be difficult to understand and implement critically for a student just learning to code. Additionally, Jockers does not focus on methods that are useful for showing change over time. As such, the book has limited application to historical analysis. Our book fills this gap by focusing on just divergence measurements as a way to make sense from time-series analysis.

- Title: Exploring Humanities Data in R: Exploring Networks, Geospatial Data, Images and Texts
- Author(s): Taylor Arnold and Lauren Tilton
- Publisher: Springer
- Year: 2015
- [44] provide an introduction to the R language. They offer chapters on subjects such as selecting and processing data in R, to chapters on statistical visualization tools, such as histograms, quantiles, and binning. Their methodologies focus on multivariate analysis and correlation. Many of their examples for processing text focus on generic text–such as lists of fruit or matrices of arbitrary numbers–instead of domain specific text, like a literary or historical corpus. This can leave readers with an incomplete understanding of how these methods can be directly applied to domain text. Their methods do not focus on showing change over time. Our book fills these gaps by showing practical examples for using R to process domain text, and by focusing on how divergence measures can be used to make sense from time-series analysis.

- Title: TidyText Mining with R
- Author(s): Julia Silge and David Robinson

- Publisher: O'Reilly
- Year: 2017
- [45] provide an introduction to using the R language specifically for text mining. Their book provides chapters on data processing, sentiment analysis, tf-idf (a type of divergence method), correlation, and topic modeling. The diverse corpuses they focus on range from 19th-century literary texts to NASA metadata. Each chapter ends by showing how a method of measure can be interpreted using a visualization strategy. Their book was released with a supplementary R package–janeaustenr–which gives users a pre-cleaned corpus for wrangling and analyzing Jane Austin's texts for comparative analyses. Our book builds off theirs by demonstrating how we can use contemporary R conventions to perform comparative analyses to show change over time. We also expand the knowledge offered by this book by focusing on divergence methods to show change over time, and by introducing natural language processing techniques for analyzing the linguistic features of a corpus.

Multiple prose-based books have also been released over the past few years that offer insight into how the fields of text mining and history have become interrelated. These books offer a theoretical overview of text mining practices, but do not provide lessons on the actual code behind the techniques they describe.

- Title: Technology and the Historian
- Author(s): Adam Crymble
- Publisher: University of Illinois Press
- Year: 2021
- [46] provides a history of technology's impact on the study of history and shows how text mining techniques and historical analysis have become intertwined. To make this history, Crymble begins by describing the origins and myths of computing in historical research. He describes the way that 19th-century educators have shaped the modern

classroom and how multiple "waves" of computational innovation (the 1960's the late 1990's with the rise of Silicon Valley, to our present technologies) have contributed to "digitizing" the history classroom. His book insists that history should "reclaim" technology by focusing on technology's influence on the history classroom and on historical research. While this book is insightful for understanding the bridge that is forming between historical research and text mining, this book does not provide instruction on code itself. Our book fills this gap by providing code-based demonstrations.

- Title: Text Mining: A Guidebook for the Social Sciences
- Author(s): Gabe Ignatow and Rada Mihalcea
- Publisher: Sage
- Year: 2017
- [47] guidebook gives a conceptual overview of the methods used for data analysis in the social sciences. They include chapters on strategies for research design, text mining fundamentals (in this case, web crawling and scraping, text processing, and supervised learning), qualitative analysis of data (by analyzing narratives, themes, and metaphors), and quantitative analysis of data (by generating topic models and classifying text). While this guidebook offers an insightful overview of many guiding principles of text mining, it does not provide any code examples, or any demonstrations of how code can be written or applied to data for the results they describe. Our book fills this gap by providing code examples along with a conceptual overview of data analysis in the field of history.

- Title: Knowledge Discovery in the Social Sciences A Data Mining Approach
- Author(s): Xiao-lin Shu
- Publisher: University of California Press

- Year: 2020

- Researchers in the social sciences are presented with more data than ever before, and in more forms than ever before (e.g. political speeches, government documents, emails, text messages, credit card transactions, parking meters, wearable sensors). [48] book shines light on big questions social scientists face today when working with such large, heterogeneous data: who owns what data? How might algorithmic bias affect access to resources? Who has access to the computational power needed to parse large amounts of data, and who has the technical knowledge and support to carry out this type of research? Shu's work begins to bridge the gap between computational methods and the research traditionally done within the social sciences by introducing methods in a few programming languages including R, Python, and SPSS. The code in this book, however, is not presented in such a way as to serve as an instructional lesson that can be applied for independent research. Instead Shu gives an introduction behind some of the language's concepts, as the primary goal of this book is to understand the process of knowledge discovery in the social sciences using new technologies in data mining. Our book fills this gap by focusing on teaching the basics of programming using examples in history. Readers of our book will be able to apply the code in this book to other problems or time-series analyses.

Instructional books that teach code for the analysis of language are popular. Many publications have computer scientists as their target audience. As a result, these books are usually in the Python programming language, not R, despite R's rising popularity in several fields that draw on corpus analytics, such as history, sociology, political science, and economics.

- Title: Text Analytics with Python: A Practitioner's Guide to Natural Language Processing

- Author(s): Dipanjan Sarkar

- Publisher: Apres / Springer

- Year: 2019

- [49] provides an introduction to natural language processing using Python. He offers chapters that describe concepts as well as introduce code for discovering language patterns within data. These include chapters on the philosophy of language, grammar, and text analytics; chapters on cleaning and structuring data; chapters introducing measurements like TF-IDF (a divergence measurement); and chapters on machine learning technologies and statistical models (including classification models, word embedding models, and regression analysis). While Sarkar provides a comprehensive introduction to the different technologies that make up natural language processing as a discipline, and provides an easy-to-follow path for beginner programmers, his book does not focus on the nuances of applying these techniques for time-series data, like historical data. Our book fills this gap by focusing on methods that are meaningful for analyzing historical events.

- Title: Natural Language Processing: A Quick Introduction to NLP with Python and NLTK

- Author(s): Samuel Burns

- Publisher: Amazon KDP

- Year: 2019

- [50] provides an introduction to text mining and natural language processing in Python. He focuses on using the Natural Language Toolkit (NLTK), a popular software module developed by computational linguistics that provides code and data for the instruction of text mining. Burns provides several chapters on data cleaning, and a few chapters on topics such as text classification and sentiment analysis. His book does not, however, mention popular natural language processing libraries like spaCy, nor does it show how these methods can be used to understand change over time for the analysis of history. Our book fills this gap by introducing a popular natural language

processing library in R, spacyr, and by using it in demonstrations for how language changes over time.

- Title: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit
- Author(s): Steven Bird, Ewan Klein, and Edward Loper
- Publisher: O'Reilly
- Year: 2009
- [51] provide an introduction to text mining and natural language processing using Python. They offer chapters on subjects such as counting word frequency, and build to more sophisticated–but still introductory–statistical analyses, like using word frequencies for analyzing conditional frequency distributions. They also provide introductions to multiple popular natural language processing modules such as Natural Language Toolkit and Wordnet. Set on providing beginners everything they need to start writing code and doing their own analyses, they also cover basic programming concepts such as how to reusing code by writing a function. While this popular book covers many useful topics, it does not include corpus analytic techniques for analyzing change over time, and their book was published before the release of popular natural language processing modules such as spaCy.

- Title: Natural Language Processing with Python and spaCy: A Practical Introduction
- Author(s): Yuli Vasiliev
- Publisher: No Starch Press
- Year: 2021

- [52] gives beginner Python programmers an under-the-hood look at spaCy–a popular natural language processing module–using Python. He includes chapters describing the components of a statistical model, how words and numbers are mapped with word embeddings, how we can use part-of-speech tags to find relevant verbs, and how users can employ spaCy's low-level data structures. His book also provides demonstrations on how to extract linguistic features. However, Vasiliev does not cover the ways in which natural language processing can be used for analysis of grammar, or how it can be used to study change over time. Our book fills this gap by using an R wrapper for the spaCy library, spacyr, and showing how part-of-speech analysis can reveal hidden language trends within a sentence or can be paired with divergence to show which language utterances were characteristic of one decade, but not another.

5.4 Intended Readership

The intended readership of Text Mining for Historical Analysis are analysts, scholars, and students of digital history and data science as well as self-taught people interested in learning about the methods used in digital history. As The Dangerous Art of Text Mining showed, the ability of data science to make sense of culture and society is enormously enhanced by adding to computational techniques historically-driven questions about what changed, when, and why. Because Text Mining for Historical Analysis offers a tour of just those data science approaches that help analysts of text to identify and interpret historical change, our book offers a crucial set of enhancements to text mining as taught in other disciplines. We expect that Text Mining for Historical Analysis will therefore offer important tools to digital social scientists and humanists as well as data scientists working on text, becoming an important textbook for courses in methodology in political science, sociology, history, art history, music, and literature, as well as for courses in Computer Science, Informatics, Data Science, and data science for the social good (which is itself a growing field in American universities).

Many of the coding exercises in Text Mining for Historical Analysis include material and explanation geared towards those who have never programmed before, or those who have some programming experience but are new to R or to its application for analyzing historical documents.

Our book will offer an ideal textbook for introductory courses in programming for Digital History, either at the undergraduate or graduate level. An appendix lays out a sample syllabus for pairing Text Mining for Historical Analysis with Dr. Guldi's previous volume from Cambridge, The Dangerous Art of Text Mining, in an undergraduate lecture course of nine weeks. Because our book begins with an introduction designed for total novices laying out all required software installations, it is suitable for classroom use even by faculty who have relatively little experience of code themselves. The book would also be useful in standard classes on Natural Language Processing in Computer Science, where students examine the abstract implications of textual artifacts for the design of new software, or an introductory or advanced class in Text in a Data Science curriculum, where students use existing software to practice real-world analysis.

In both Data Science and Computer Science departments, many instructors seek hands-on examples of how to apply code to practical questions of research from other disciplines. Our prepackaged datasets and software and the chapter-by-chapter examples of applied questions in digital history make Text Mining for Historical Analysis an easy choice for instructors who want a "classroom-ready" textbook.

5.5 Chapter Outline

5.5.0.1 Introduction: Getting Started With R and RStudio

The introduction reviews the distinctiveness of our approach to digital history, which is characterized by an approach we call "critical search." In critical search, the tools of humanistic knowledge – that is, historical context, appreciation of bias, and theory-driven approaches – are paired with a variety of quantitative tools for comparing the characteristics of different episodes of time. The result, we believe, is an approach that supports critical thinking about what happened over time, giving the student of digital history tools to answer that question with a variety of mutually-reflective approaches, testing how the answers change with a slight adjustment of algorithm or vocabulary.

The introduction introduces the data sets used by this book: the 19th-century British Parliamentary Debates of Great Britain, also known as Hansard, and the 20th-century transcripts of the debates of the Congress of the United States. We choose these data sets

because they makes available to researchers new historical insights into themes such as the Industrial Revolution, the abolition of slavery, the spread of empire, and the crisis over the environment. And while they include only the voices of a tiny minority of historical actors, these datasets come with robust information about the dates on which speeches were given and the names of the individuals who delivered speakers. We believe that because (rather than in spite) of their limitations, these two datasets are an ideal training dataset for demonstrating the bias of historical texts on issues such as gender, class, and race, and how that bias evolved over time.

Finally, the introduction offers an accessible start to coding with R, aimed at those with little to no programming experience. It walks readers through the process of installing the R programming language as well as RStudio, a development environment for R.

5.5.0.2 Chapter 1: Introduction to Text Mining for Historical Analysis Using R

Chapter 1 introduces three basic concepts for data processing in R: a) loading a decade of the Hansard data from the hansardr library; b) filtering that decade for keywords of interest; and c) counting words.

Readers will use the "filter" command to search for the text around the keyword "coal," a major subject of debate during the nineteenth century. They will use the command "count" to count the frequencies of words that form the context for the keyword. The textbook will give readers the code to render this information about the context of "coal" as a bar chart.

The chapter will introduce students to some first opportunities for critical thinking about how the study of change over time provides an opportunity for insights into texts and the clues they provide about culture. As readers are introduced to code for visualizing the data, readers will first create a "bad" plot, in this case, a plot of "top words per year" for the 1830s. As the narrative prose will explain, this graph is misleading because it contains stop words and does not normalize the data for the increasing number of debates throughout the 19th-century. Later, readers will correct this "bad" plot by making a more informative one that visualizes proportion instead of raw count.

Readers will also be introduced to basic concepts for visualizing data for digital history. Following the provided code, readers will create a plot that shows change in word frequency

over time for keywords such as coal for the decades 1830 and 1870. Visualizing these plots side-by-side, readers will have the beginning tools for a comparative analysis.

5.5.0.3 Chapter 2: Text Mining History Using a Controlled Vocabulary

Chapter 2 will introduce readers to several of the basic functions of the programming language R which are used to manipulate datasets to produce analysis. Two commands – "join" and "groupby" – provide a student of R with the flexibility to identify and examine abstract categories in the dataset, for instance, allowing the analyst to study the words that pertain to "fields" of the data such as the decade or year in which a speech was given or the speaker who gave the speech, or to extract special words and phrases of interest for further analysis.

In this chapter, students meet the "join" command, which allows an analyst to fuse together two datasets to produce a third dataset with new principles, and the "groupby" command, which allows the faceting of data around a common data field. Using the "groupby" command, readers will be able to re-organize existing data around a "field" (or property of the data) such as decade, year, or speaker.

Our code will lead readers through the process of asking the question: Which speakers spoke the most in a given decade? By "grouping" the data by speaker and using the command "aggregate" to add per-speaker word counts, the reader will produce a new dataset of how much each speaker spoke during the 1830s and another for the 1860s, two key decades for democracy and representation in Britain. The code will lead them through the process of creating two bar charts, one showing the top speakers of the 1830s and another for the 1860s.

A second exercise in this chapter will invite readers to investigate the idea that representations of memory change over time, using the approach known in data as a "controlled vocabulary," where the analyst searches for a set group of phrases. In our case, the controlled vocabulary will be a list of events from British history, for example the "Magna Carta" and the "Glorious Revolution." Having such a list of vocabulary will allow the user to measure how parliamentary speakers' references to the past changed over time.

Using the "join" command, readers will have the opportunity to weld together decade-data about key words and phrases and the list of events Using the aggregate data produced

by this join, readers will count the top mentions of discussed in each year of the 1860s, a decade of rapid change to the workings of British democracy. Using another set of joins, this one with a list of overall counts of words per year, will allow the reader to graph the events mentioned as a proportion of overall words in the decade.

5.5.0.4 Chapter 3: The Distinctiveness of Certain Eras

This chapter introduces readers to "distinctiveness," a statistical principle for identifying the key differences between documents, speakers, years, or other facets of an archive. Readers will be introduced to the idea that we can mathematically measure how "distinctive" two aspects of data are of each other, a question that allows the analyst to answer: is the word "coal" particularly distinctive of Gladstone, or do other members also use it? Is the word "coal" particularly distinctive of the 1830s, or was it also talked about in different time periods?

The chapter introduces the idea that vector-based representation of word counts can be used as the basis for comparing documents and that there are a variety of mathematical approaches for identifying which differences between documents are the most salient. The text then introduces one popular algorithm for identifying document distinctiveness, term frequency inverse document frequency (TF-IDF). We will share the equations and code for producing tf-idf measurements and contingency matrices of words and their relationship for particular speakers.

Turning from theory to practical application, readers will be led through the process of measuring the words that are most distinctive of two prolific members of Parliament, Benjamin Disraeli and William Gladstone. Following the code provided, readers will have the opportunity to represent the words most distinctive of each speaker as a bar chart.

We will also share code for applying distinctiveness as a measure to find the vocabulary that are most distinctive of each twenty year period throughout the 19th-century. To produce these measurements, readers will join different subsets of the hansardr corpus. They will produce two different representations of this information – first, a bar chart per decade of the most distinctive terms, and second a timeline of the century where the most distinctive words are represented as columns.

5.5.0.5 Chapter 4: The Many Ways to Measure Distinctiveness

This chapter introduces an important opportunity for critical thinking in the research process: the idea that each algorithm has a bias of its own. Small differences in what is measured and how can produce major differences in result. Inspecting the implications of adopting different algorithms is one approach for adopting critical thinking in the process of humanistic work with data.

This chapter gives students a hands-on experience of applying two statistical tools for comparing temporal change over time: JSD Divergence and tf-idf. It introduces each equation, discusses the history of the mathematical theory behind the measure, walks the student through the process of applying the equation to the data, and then helps the student to compare visual analyses based on the equations. We hope that readers will become comfortable with the idea that algorithmic approaches to a general problem such as distinctiveness are varied, and that the researcher can learn from comparing the results of working with different algorithms from different mathematical families.

The chapter begins by introducing the equation and code for a common measure for distinctiveness, Jenson Shannon-Divergence (JSD). Using JSD, readers will be walked through the process of measuring which words are more likely in the first half of the 19th-century Hansard debates (1803-50) versus the second half of the debates (1851-1899). We will provide code for outputting the most distinctive words of each half of the century.

The chapter will also lead readers through the process of comparing the results of using different measures of distinctiveness. Readers will recreate the results of the comparison between Disraeli and Gladstone from the previous chapter using JS instead of tf-idf. The results will demonstrate how tiny changes in how difference is measured create slightly different results. We will encourage readers to meditate on these differences, using them as a way of expanding their insight into the nature of "difference," the use of comparable algorithms as different tools in a process of analysis, and the beginning of a process of critically thinking about data, algorithm, and historical truth.

5.5.0.6 Grammatical Analysis of Natural Langauge Speech

In Chapter 5, the narrative takes a turn from studies in baseline statistics to studies that leverage the "machine learning" or "artificial intelligence" tools developed by computer scientists, computational linguists, and others over recent decades. These powerful tools allow researchers to study patterns in language change that happen not at the level of words or phrases, but at the level of sentence grammar. Using tools of this kind, a researcher can ask important questions about the imagination of agency in the past.

The reader will be walked through the process of parsing text and extracting co-occurring words that share a grammatical relationship. First readers will extract adjectives that complement the noun "woman" from the 1870 Hansard debates. We will count the most frequent adjectives and graph the results. A discussion will underscore how a naive interpretation of this graph might suggest that women in the 1870s were particularly "ignorant" or "wanton," some of the major words in the graph. The discussion will emphasize that historical interpretation requires the analyst to separate the representation of the past as it comes down in the documents that survived from the unknowable reality of the past as it was lived. We will argue that the best interpretation of such a diagram is as a tool for understanding the bias of the members of parliament who were investigating the conditions of working women in the era of the Contagious Disease Acts.

In the process of learning about grammatical relationships, the reader will be introduced to tools from spaCy NLP, a popular natural language processing library that can be used to parse text and tag words with their part-of-speech (such as "verb" or a "noun") and their syntactic dependencies (denoting which words grammatically depend on each other, such as which adjectives modify which nouns). One advantage of using spaCy is that we can also "lemmatize" words, meaning that we can extract with accuracy from each word its morphological base, that is, without verb tense or plural form ("rents" becomes "rent"), thus improving word counts.

5.6 Manuscript Draft

First Time Set Up

This book introduces text mining for historical analysis using the R programming language. Many of the exercises in this book are based on code samples from Jo Guldi's *The Dangerous Art of Text Mining* (2023) that have been rewritten with the purpose of offering an practical point of entry into the practice of digital history for those with no previous acquaintance with the basic methods of programming.

The intended readership of *Text Mining for Historical Analysis* are analysts, scholars, or students interested in learning about the methods used in digital history. Many of the coding exercises include material and explanation geared towards those who have never programmed before, or who have some programming experience but are new to R and its historical application. The first two chapters offer guided approaches to the programming concepts that are built on throughout this book.

The experiments in this book all concern one dataset, commonly known as "Hansard," which contains the official record of the debates of Britain's House of Lords and House of Commons, 1803-1899. It is a highly interesting data set because the debates cover the transformations of the Industrial Revolution, the abolition of the trans-Atlantic slave trade by the British navy, the struggle for women's rights, and many other debates of general interest to readers of modern history.

As we will explore in the chapters that follow, Hansard is in many ways an imperfect data set. The Hansard data dates from a moment in history before the advent of modern technologies of transcription, including shorthand and electronic recording. Instead, it was compiled from press

reports and notes, highly edited for what journalists assumed would be of interest to contemporary readers, and sometimes paraphrased to keep pace with the speaker. Nonetheless, even in the early versions of the reports, a great deal of data about the social, economic, and political realities facing parliament remain preserved. Though far from a verbatim transcript of the records of parliament, it is nevertheless an important historical source that has served generations of historians, many of them working with concerns social and cultural as well as political and intellectual.

Part of what the data set contains is a record of the bias of an earlier age: an age where the vote was limited to a tiny portion of the aristocracy (before 1832) and the middle class (to 1867). The record is also almost exclusively male, as women could not vote in Britain before 1918. Britain of course controlled a global empire, and many of the speakers expressed disdainful attitudes towards their Roman Catholics, Indians, Africans, and Irish subjects; indeed, historians have reasoned that it was the British willingness to fabricate arguments about racial superiority and the necessity of violence that allowed the empire to exist. The techniques in this volume allow the student of history to examine British attitudes about gender, race, and class for themselves.

Many practitioners of text mining in the humanities and social sciences have expressed the importance of the role of critical thinking when working with data. Indeed, the scholar of text mining who takes a naive position on the data in Hansard is likely to conclude, as some of our students have, that the upward tick of the phrase "ignorant women" across the nineteenth century means that there were actually more ignorant women over time. In fact, the phrase indicates the formation of what historians call a "discourse," that is, an increasingly accepted representation of reality. A representation, however, is not the same as reality. In the nineteenth century, speakers in parliament found that talking about the ignorance of women was useful as a tool to justify the much of the legislation they were trying to pass, especially in the era of the Contagious Disease Acts (1864-69). Parliamentary speakers – again almost exclusively male – blamed women for the spread of venereal disease through the British navy. The habit of blaming women had profoundly negative consequences for contemporary women, who lost important liberties during this period. Because of political habits

of blaming women, the laws were written in such a way that it was women who were arrested in the attempt to stem the spread of disease.

The point of this example is that an intelligent, reflective process of text mining requires more than the skills of counting words. One must know a little about the history. One must also have the power of reflection on where language operates as a game where assertions are made that might not reflect the truth of reality, and where the spread of those assertions, or their formation into a discourse composed of many apparent facts, can have real-world consequences in history. Counting words and phrases gives us hints about how discourses changed, but it is not in itself sufficient to produce understanding. In one recent book on text mining for the social sciences, Brandon Stewart and his collaborators stress the importance of iterative engagements with the data, validating each conclusion with extensive reading of the original texts. We too advise that part of the skill of text mining is the power of navigating from big-picture overviews of how discourse changed back to the original speeches, and again to some knowledge of the time period from secondary sources, without which the student of text mining is navigating a cave with only a book of matches. Accordingly, the chapters in this book alternate between practical instruction in code – for instance, instructions on how to count two-word phrases – and exercises designed to model a reflective and critically-aware point of view. These exercises guide the reader through the process of "validation," or checking what the content of the model means, by moving from an over view to individual speeches. They also ask the student of text mining to reflect on the meaning of particular trends visible in the data given some knowledge of the history of the delimitation of voting in Britain, the politics of slavery, and the political use of claims about gender.

What if you aren't a historian of nineteenth-century British parliament? By the book's end, we will offer advice to those working on the Congressional debates of twentieth-century America. More importantly, however, we submit that the exercises will still be in principle interesting, as they are formed around the problem of counting words. The tools of word count and analysis from this book can be applied to different data sets as soon as they are learned. New data sets are constantly

coming into use, and scholars of practically any moment in history may find data to which the following lessons can be applied with relatively minor adjustments such as matching the names of columns where speaker, date, and speech data are held.

More than just a code cookbook, however, *Text Mining for Historical Analysis* exemplifies the tenets of "critical search," a framework for analysis that insists on performing strategic reading and implementing multiple methods of measurement. Any search through text illuminates various dimensions of a corpus while obfuscating others, and critical searching can enhance the analyst's sense of the text measured by considering how certain vantage points share different information.

This framework for analysis also encourages analysts to use quantitative measurements side-by-side with close reading techniques. Contextual, historical knowledge should inform our assessment of the linguistic trends belonging to historical structures of meaning such as discourses and contested concepts.

It is our hope that *Text Mining for Historical Analysis* supplies its readers with the tools and perspective that incite a discerning sense of discovery and that enable readings of history that share the insight uniquely offered by computational analysis.

First Time Set Up for R and RStudio

Analysts of historical data often collaborate with skilled code practitioners or use out-of-the-box search interfaces to conduct their work, but working hands-on with a programming language gives the analyst direct control over every aspect of the data quality, its analysis and visualization. This introduction is designed to introduce you to the language R with no prior experience required.

Analysts of historical data sometimes use other languages than R, although at present, the languages Python and R are the most frequently consulted for text mining. In general, R is more frequently taught in economics and statistics departments, while Python is more frequently used in Computer Science. Both are legitimate interfaces for analyzing historical data. Certain text-mining processes

might be more suited to Python, for example named entity recognition, a process discussed in the **Dangerous Art of Text Mining** but not here, or the scraping of text from Twitter or websites. We urge those curious about these techniques to consult Melanie Klein's wonderful introduction to cultural analytics in Python.

In general, the learning curve on R is less steep for new comers, while giving users direct access to functional tools for dealing with language. R's packages are, at the moment of writing, more dependable for many kinds of textual analysis, thanks to the existence of certain software packages developed by the text mining community for the purpose of analyzing words and phrases in various configurations. A lively community of R users practices a high standard of peer review for software packages on R, which ensures that software updates are almost always kept in a state of compatibility with each other. This trustworthy community makes R an ideal choice for those who want to focus their primary efforts on analysis, as those learning R will not face many of the questions about version compatibility that face users of certain versions of Python.

The first step is to make sure that you have access to a computer that is set up to use the R programming language. The steps that follow explain how to install R on a Mac or PC (Windows machine).

We also recommend installing a desktop app called RStudio designed to make working with R as intuitive as possible. RStudio is a "code editor" that makes it easy to interact with the R language and your data as it undergoes transformations. A code editor is to the language R what Microsoft Word is to writing a document: there are ultimately many applications which one could use to write a document, but Microsoft word is a standard because it makes outlining, drafting, printing, and footnotes easy to manage. Just so, RStudio will be our preferred app for accessing R.

Afterwards, the R interface may alert the user of occasional updates to the software, which is important to keep up-to-date. When these updates are necessary, the R interface will provide its own instructions, and for the most part, it is sufficient to simply type whatever is indicated on the

computer screen, perhaps restarting the computer if the user encounters difficulty.

If you have already installed R and RStudio (or a comparable development environment of your choosing), you should skip to section "Loading Libraries" below.

System Requirements

Using a programming language and processing data requires some amount of computer resources. We suggest the following minimum system requirements to run the code examples in this book:

- Operating System: Windows 7 or macOS 10.12
- Available Disk Space: 6GB
- Memory: 8G

Installing the R Language

If you are using Mac go to the R project at https://cran.r-project.org/bin/macosx/. You will have two options of the R language to choose from: a R.pkg file and a R-arm64.pkg file. The R.pkg file is for Macs with Intel processors. The arm64 version is for Macs with arm processors. If you do not know the processor of your Mac, go to the apple menu and click on "About This Mac." Your computer's processor will be listed.

If you are using Windows go to the R project at https://cran.r-project.org/bin/windows/base/. Click "Download R for Windows" at the top of the page.

If you are using Linux your steps for installation may be different depending on your distribution. If you have Ubuntu, one of the most popular distributions, you can start by follow the instructions here: https://cloud.r-project.org/bin/linux/ubuntu/fullREADME.html.

Installing RStudio

Go to https://posit.co/download/rstudio-desktop/#download. You will have the option to download the correct version of RStudio for your operating system.

Running Your First Line of Code

When you start RStudio for the first time you will be presented with a screen divided into four parts, each of which represents one portion of R's coding environment.

The first place to look is the lower left-hand quadrant of the RStudio interface, where you'll see the word "Console." The panel in question is known as the "Console Tab Panel." A "console" is a window that allows the computer's user to send code directly to the computer. You may be familiar with consoles from other places on the computer: PC users use the Command Line Prompt – another name for the Console. Mac users use a console when they open the "Terminal" app to run command-line commands. In RStudio, the console panel is one of two ways that allow you to write R code interactively.

To see RStudio's console in action, inside RStudio's console panel, position your cursor at the prompt (">"). Type this simple line of code: `print("Hello World")` and press Enter/Return. The console will return "Hello World." You have just run your first line of code.

In RStudio you have the option of typing all of your code into the Console. However, running code line-by-line at the console makes it difficult to save, edit, and revisit. This is why RStudio provides another panel specifically for editing R scripts, the "text editor." The text editor panel is directly above the Console Tab panel in the upper left-hand quadrant of Rstudio.

The Text Editor in RStudio is to R code what Microsoft Word is to text files: it offers a software environment that makes it easy to write, save, revise, and annotate code. There are many other coding programs that would work, and coders comfortable with these programs may adapt our instructions. This textbook, however, recommends that first-time programmers write their code in

RStudio by default.

To use the Text Editor in RStudio, you must first open a new file. Go to "File," "New File," and then "R Script." A new tab will open where you can draft multiple lines of code to be run later. For now, in your new R Script tab, type the same command as you typed into the Console: type `print("Hello World")`.

Next, you will want to "run" the line of code that you just typed. To run a line of code from the Text Editor, you will click the "Run" button at upper-right-hand corner of the Text Editor panel. You can also use your keyboard (try this too). Position your cursor anywhere in the line of code in the Text Editor, and hit the keys `CTRL + Enter` (on PCS) or 'COMMAND + Return' (on Macs). To see what you did, look at the Console again, which will continue to keep track of the computer's response to all lines of code sent to the computer. In the Console, R will return "Hello World."

Loading Libraries

The "datasets" library is a set of quantitative data sets that are frequently used to teach code, although because this book is about text, we will only consult them once for the purpose of teaching.

```
library(datasets)
```

When a software package is called by the command 'library', its data and commands are available for you to use with R. R keeps track of which packages have been installed and loaded in the 'Packages' tab, which is one of the tabs available in the lower right-hand quadrant of RStudio.

In the lower-right-hand quadrant of Rstudio, click on the "Packages" tab now, and you'll see a list of software packages that came pre-loaded with R. Notice that some of these packages are checked, meaning that they are already loaded and ready for use.

Scroll down until you see the package "datasets." It should have a check mark next to it – the result of your loading the package with the command above. You can uncheck it and re-check the box

next to datasets: the "Package" panel offers you another way to load a software library. Make sure that the box next to "datasets" is checked before you move on.

One thing to remember: if you've perfectly typed a command that you expect to run, and R gives you an error message, one thing to check is whether all the libraries needed for a particular exercise are checked.

It is important to note that packages only need to be installed once but libraries need to be reloaded every time you start a new RStudio session.

Loading Data

Because we've turned on the package "datasets", we can next call the dataset "iris".

The command "data" tells R to load a data set from one of the available software libraries. Back in the Text Editor panel, type in the following line of code, then run it using the "Run" button or CNTRL+Enter / COMMAND+Return:

```
data(iris)
```

To see what you've done, look in the upper right-hand quadrant of RStudio, known as the "Workplace Browser" panel. By default, you will be looking at the "Environment" tab.

In the Environment Tab, you will see that there is only one data set currently listed: "iris." The Environment Panel tells us what data and variables are loaded. It is useful for checking what data is available and what it looks like. The Environment Panel lists "iris" as having 150 "obs" or "observations" of "5 variables," which is another way of saying that there is data that could be displayed in a spreadsheet of five rows and 160 columns, or 150 rows and 5 columns.

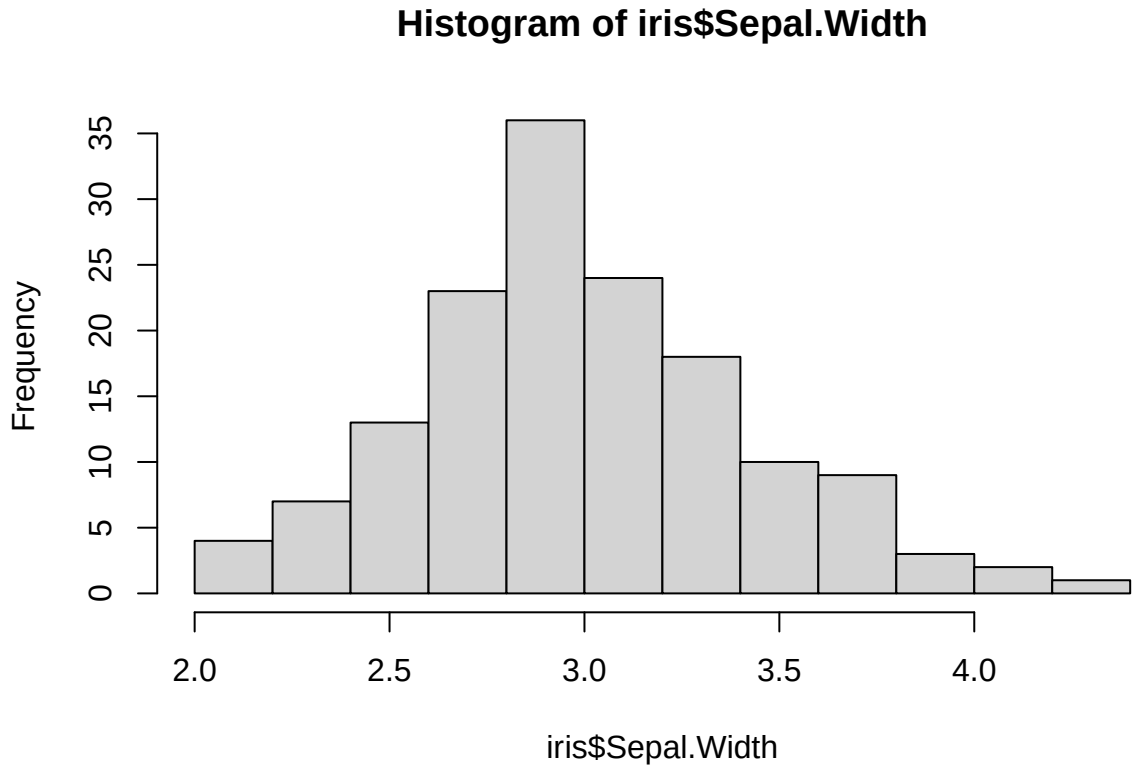
Click on the blue and white arrow button near iris to see a sample of the content of the data set.

When you run code, you may periodically want to check what your data looks like. Knowing how to inspect data in the "Environment" tab is a useful skill.

A Simple Visualization Next, we'll make a simple visualization. First, find the lower right-hand quadrant in Rstudio, where we formerly inspected software packages. Click on the "Plots" tab in this quadrant, which is where visualizations are displayed and temporarily stored. It should be blank.

Back in the Text Editor, type the following line of code and run it:

```
hist(iris$Sepal.Width)
```



The '$' marker tells R to select from the iris data set just the data in the column labeled "Sepal Width." If you're working with data that looks like a spreadsheet, for instance a data set with columns and rows, you can call the columns with the same marker. In R, these columns are known as "facets" of the data, and as we shall see, there are multiple ways of accessing them. We will

only rarely see the dollar-sign marker in the future, since we will be working with another set of commands known as "tidy" commands, which make it easy to work with data that takes the form of rows and columns.

Our first visualization command, "hist," is also not one that we'll see frequently in this book, but we use it here because it's fast. The "hist" command makes a "histogram" or bar chart of frequencies for each variable in a list. In this case, the "iris" data set has a list of the width of sepals of various kinds of irises. The histogram shows us that irises with the sepal width of 3 are by far the most common of all iris kinds. Notice that R has automatically attempted to make the resulting visualization as elegant and transparent as possible, even without our providing much by way of information. The visualization has a title and its axes are labeled. In this book, we will make many more complex visualizations which have custom axis labels and captions describing the data sets' content in detail.

The "Plots" tab temporarily stores all of the visualizations you make in an R session. To store these visualizations, you can pull down the "Export" menu (just below Plots/Packages/Help/Viewer). You'll see options to export the visualization as a pdf, to the clipboard, or to another format. Choose one and save the histogram somewhere on your computer.

In Chapter 1, we will meet the "Hansard" data set which includes a century of parliamentary debates, and we will begin counting words and visualizing them.

Working with Historical Data

The data samples chosen for this book are large for an instuctional textbook. One reason the samples are larger than typical is because we often work with full, decades-worth of data at a time, comparing them against one-another. The size of this data more closely resembles the kinds of data sets and computer requirements needed for serious historical research. One way we deal with the largness of our data is by designing steps that process the data in such a way as to make it

small enough to run on a computer with the above mentioned minimum specifications. Even still, we encourage the reader to clear the Global Environment (and remove all saved variables) before proceeding to a new chapter. To do this, click on the broom icon in the Global Environment panel and confirm you wish to remove all objects from the environment. Next, click on the disk icon in the Global Environment panel and select "Free Unused R Memory." You may also remove specific objects from the environment using `rm()`.

```
rm(iris)
```

The Errors We All Face And How to Learn

This book is designed to be accessible to first-time coders. It assumes that you have mastered the skills of installing packages, loading packages, inspecting data, and saving visualizations described in this introduction.

One route of learning is working through the code in the chapters that follow. The reader will type each line of console-type font in to the Text Editor panel of RStudio, running one line of code from beginning to end.

In the process of typing in code, first-time programmers are likely to commit errors, and indeed even advanced programmers make typos. In programming languages, a typo will produce an error message or cause the program to perform with unintended results. For an example, try typing the following line into the R console and pressing 'enter':

```
print("hello world)
```

This line will result in an error, because a close quotation mark is missing on the other side of the phrase "hello world."

As you learn to code, whatever your level of experience, you are almost certain to have typos and errors that stop the program from performing as you expect. We advise you to ask yourself what

went wrong when you inevitably get error messages or unexpected results. Through closely following the examples in the book and paying attention to any error messages that result, you will gradually learn to recognize common errors like missing parentheses or quotation marks.

What can you do as a first-time coder of the R language for analyzing historical text? The book's authors assume that you will proceed by following each line of instruction, typing in the given lines of code, and running them. If you follow these instructions, you will soon enough find yourself capable of adjusting the lines of code in certain ways, for instance, making a bigger visualization or showing a different sample of the data. At least one of us learned to code this way by working through the exercises in literary text mining from Silge and Robinson; together, we have guided many analysts with no background in code through the basics of text mining. We believe that by working through the recipes in this volume, inexperienced coders can become digital historians, capable of detecting many previously invisible patterns in the archive.

Can you teach this book to others if you've never coded before? Yes, you can, and rapidly, with some provisos. Our experience was learning R in a series of weekly, 3-hr private tutorials with a Statistics Department graduate student one summer. The graduate student was paid an hourly rate out of a research grant, and the mini-seminar was attended by Prof. Guldi and two of her humanities graduate students. In the following semester, the same Statistics graduate student was paid an hourly rate to help teach a small, pilot introductory course on Digital History. The professor remained responsible for discussions of how quantitative and text-analysis tools are being used in History, and for the grading of all assignments. The graduate student was available during class meeting times to help oversee the installation of RStudio, to troubleshoot troublesome code, and to add mini-lectures explaining the more technical dimensions of the code. In this way, one professor went from novice to teacher of code in half a year, given an institutional investment of well under five digits.

We also believe that it is possible to begin with the exercises here, working on the British parliamen-

tary debates, and move to other data sets from different historical periods or voices, even though that material is not covered in this book. Scholars of history usually come to the digital humanities with pre-formed commitments to particular times and places. If a data set exists from that time and place, it can be "swapped" for our data set in any of the exercises in this chapter.

Text books offer an excellent resource for scholars who have questions about programming beyond the practical introduction in this book. If you have never programmed before you might find *Hands-On Programming with R* by Grolemund (2014) or *R for Data Science* by Wickham & Grolemund (2016) to be a useful supplement to this book. If you are new to text mining and want to learn more you might look into *Text Mining in R* by Silge and Robinson (2017) for its Tidyverse R implementation, or Quanteda (https://quanteda.io/) for a comprehensive natural language processing (NLP) tool set.

Text Mining for Historical Analysis is a preliminary text book for practical text mining, and it was intended as a survey of the individual techniques we have found most useful in our own research. It is not a programming manual, nor is it a book of theory. In the course of using these exercises, some learners of text mining will want to explore the rules of code syntax. It was written with the idea of offering a practical companion to the theory of text mining for history offered in **The Dangerous Art of Text Mining**, and in the appendix we include a sample syllabus showing how chapters from the two books might form compatible assignments for an undergraduate-level course.

Chapter 1: Counting Words in the British Parliamentary Reports

The introduction to text mining in this book begins with tracking words' context. As the linguist John Rupert Firth said, "You shall know a word by the company it keeps." Tracking subtle variations in the context with which different words are used helps historians and other analysts to understand the cultural biases that structure our collective thinking. They may help us to understand how a culture understands men and women differently, how it talks about people of different nationalities, the difference between a concept like "the foreign" and its near synonyms like "outsider" that nevertheless are used important and different ways, or how these concepts are changing over time. In the course of this textbook, we will offer a taste of the subtle and refined approaches that scholars of texts and data scientists have used to unpack the context of different words.

In this chapter we introduce three basic techniques for processing text so as to foreground change over time through investigating the changing relative frequencies of a keyword. In this chapter, we will: a) access data for the British parliamentary debates of the House of Lords and House of Commons, 1806-1911 from the *hansardr* library, b) filter that decade for keywords of interest, and c) count the words that appear in the same context as our keywords in order to study the keywords' meaning in more detail.

Importantly, the techniques presented in this chapter have both promise and limits. As our companion volume, **The Dangerous Art of Text Mining**, makes clear, simple word counts are almost never sufficient on their own to support an in-depth analysis of culture. Nevertheless, knowing how to count words and compare the context with which different concepts are used is a first step to using text as data.

Loading the Hansard Data

The first step in working with text as data is to load the data we wish to analyze. Users of code routinely import software packages developed by other coders to expand the range of tools available to them.

The authors of this book developed a data package to go with this book: hansardr.

hansardr is a package for easily accessing our version of The Hansard 19th-Century British Parliamentary Debates with Improved Speaker Names within the R environment. This is a cleaned, analysis-ready corpus of the 19th-century British Parliamentary Debates (1803-1899), also known as "Hansard" in reference to Thomas Curzon Hansard, the publisher. It identifies debates whose records are missing from UK Parliament's corpus, and it also offers a field for disambiguated speakers. We believe these improvements will enable researchers to analyze the Hansard debates, including speaker discourse, in a way that has not been accessible before. Once installed, the data can be loaded with the familiar `data()` function.

To use a command from a software package, one must first install the package, something we learned to do in the introduction with the command `install.packages()`. After a software package has been installed the first time, we can load the software package into memory using the command `library()`. A student of code typically loads the software packages to be used at the beginning of each new coding session. Loading only relevant packages for each session helps to keep the computer running quickly, which allows us to process large-scale data quickly and efficiently. For teaching purposes we will regularly load packages throughout a chapter so the reader can see which lines of code evoke different packages.

hansardr can be installed from the Comprehensive R Archive Network (or "CRAN" for short). Use `install.packages()` to install a package from CRAN.

```
install.packages("hansardr")
```

If you have **devtools** installed, **hansardr** can also be installed from our GitHub repository:

```
library(devtools)
```

```
## Loading required package: usethis
```

```
install_github("stephbuon/hansardr")
```

```
## Skipping install of 'hansardr' from a github remote, the SHA1 (d9a77d4f) has not changed si
##   Use `force = TRUE` to force installation
```

As we explained in the introduction, packages only have to be installed once, but they need to be reloaded for each new R session using `library()`. `library()` tells R to load software programs that give us additional functionality.

```
library(hansardr)
```

hansardr and dhmeasures are now ready for use.

Navigating the hansardr data

To make working with the Hansard corpus quicker and easier–and to prevent our computers from being overloaded with too much data–we broke the century long corpus into decade sections. In the field of data analysis, it's common to call these sections "subsets." The first kind of dataset we will meet is a collection of text: this data covers the recorded words spoken in parliament, 18063-1899, broken into sentences. Later on, we will meet extra datasets that contain "metadata," or information about the speaker, year and date of the speech, and other important contextual information which is useful to historical analysis.

In this chapter, we will begin working with the text-based datasets, because they are easy to manipulate to count words. Counting words is the first step in most text mining for analysis of content.

Both text and metadata are divided into subsets named by decade, for instance, *hansard_1850.* We can use the command `data()` to load one of these decade-based subsets of the Hansard corpus.

```
data("hansard_1850")
```

A new variable named `hansard_1850` is now viewable in our Global Environment panel. If you are using RStudio, you can explore the data by clicking on its name in the Global Environment pane (the top right quadrant of RStudio). Clicking on the name of the dataset will open a new tab containing a view of the data, which you can search by keyword as you would in a spreadsheet. You can also double-click on any row in the data frame to select a row, then copy that data and paste it into a Word Document or other tool. Please try copy and pasting some data from `hansard_1850` now.

In the Global Environment Panel, you can also click on the blue ">" symbol to the left of each data frame to display the "fields" by which the dataset is structured, in this case, the names of the columns in the dataset, as well as a preview of the content of each of these columns. Please investigate the data now.

Throughout this book, we will routinely "inspect" the data by using commands like `head()` to see how the data is transformed between commands; this is a good habit to get into as you code. A few commands will give you the tools to look at any part of the dataset in detail.

Most of the time, when we load the data, we will want to quickly make sure that the data looks like what we expect; otherwise, errors may be produced. At the command line, you can also use the command `head()` to ask R to display the first six rows of the data. In some versions of R Studio you will see an arrow that allows you to move between two columns. The first column is `sentence_id`, which gives a unique identifier for each sentence, including information about which series of parliamentary proceedings we're looking at, which volume and page, and which sentence number on each page). The second column is `text`, and each row is a distinct sentence of text, as identified by computational tools that look for punctuation and line breaks.

```
head(hansard_1850)
```

```
##   sentence_id
## 1 S3V0108P0_0
## 2 S3V0108P0_1
## 3 S3V0108P0_2
## 4 S3V0108P0_3
## 5 S3V0108P0_4
## 6 S3V0108P0_5
##
## 1
## 2
## 3
## 4
## 5 Although the Act of Parliament directed that the proclamation should issue forthwith for
## 6
```

Tidy Text Mining and Loading Software Libraries

Most of the commands used in the first chapters of this book come from a software package called *tidyverse* which is actually a family of smaller software packages developed by Chief Scientist at Posit (formerly known as RStudio), Hadley Wickham.

The tidyverse commands were created to order data in "tidy" tables where data is structured as a data frame made up of columns and rows, and every observation has one row. In other words, the tidyverse is built to work with data like the "text" column in *hansard_1850*.

We have found that a tidy approach to text exploration and analysis allows us to teach code quickly and effectively, giving analysts ample control over the data at all times, and allowing them to

inspect the output of every form of analysis. This powerful library thus allows analysts to focus on analytical questions without getting weighed down by the nuances of different structural and stylistic coding problems.

We will also use the *tidytext* library, developed by David Robinson and Julia Silge to work with textual data. The tidytext library contains a variety of commands useful for processing text and treating text as data. Most lessons in this book will therefore begin by loading the two libraries tidyverse and tidytext.

```
library(tidyverse)
library(tidytext)
```

Preparing Data for Wordcount: The `unnest_tokens()` Function

In order to count words over time, we first want to put our data into a format where each word of the text is on a separate row. Only in this one-observation-per-row format can a computer easily count each word.

As we have seen, the data in *hansard_1850* is structured into sentences or sentence fragments, with one sentence per row in the column "text." To count the words spoken in parliament in the 1850s, we must begin by treating the speeches as bags of words that can be counted and analyzed.

The *tidytext* package provides several functions that are useful for processing text. Learning to use R typically proceeds by mastering one function at a time. "Functions," or commands, are the verbs of the computer world: they tell the computer what to do to the data, one step at a time.

Our first function, `unnest_tokens()`, tells the computer to break up strings of text – like the sentences in hansard_1850 – into individual tokens. "Tokens," in this case, refer to any unit of text that is meaningful for analysis, typically a word, a multi-word phrase (also called an n-gram), or a sentence. The process of splitting chunks of text into individual units is called "tokenization."

In the line of code below, we apply the function `unnest_tokens()` to the "text" column of *hansard_1850*. We assign the output to a variable named *tokenized_hansard*. Instead of a column named "text" the output has a column named "word" containing each individual word of the corpus.

Throughout the course of this book we will introduce multiple different kinds of symbols, called "operators." Certain operators connect the input, function, and output. The assignment operator, `<-`, sometimes translated as "gets," points to the output of a command and is followed by the name of an original dataset. The symbol `%>%` called the "pipe," and translated "next," is used to string together successive transformations of a dataset. This will make more sense when we cover more examples later in the book.

```
tokenized_hansard <- hansard_1850 %>%
  unnest_tokens(word, text)
```

If this is your first time using a programming language for analysis, it is useful to practice trying to read each line of code as a sentence of instructions which tells the computer what output to make, what input to use, and what processes to go through in a certain order.

We sometimes suggest that first time programmers translate code into a series of English sentences, using their knowledge of the `<-` and `%>%` operators, the names of variables, and their understanding of distinct commands. The line of code below could be read as follows: "Create a new variable called *tokenized_hansard*. Start with the dataset *hansard_1850*. Next, unnest the tokens in the *text* column of *hansard_1850*, unpacking those strings of text into individual words."

In the line of code above, you may also notice the use of the argument "word" in the line of code, `unnest_tokens(word, text)`. "Arguments" refer to the instructions given to a command in code. Commands like `unnest_tokens()` take a set number of arguments, typically in a precise order. The arguments gives the command precise instructions about what to do. The function `unnest_tokens()` takes two arguments: the first is the kind of token to identify, whether a word or a sentence. The second argument is the name of the column to work on, in this case, "text."

We have instructed `unnest_tokens()` to "unnest" the sentences in hansard_1850 into individual words – hence the "word" argument in the line of code `unnest_tokens(word, text)`.

As you inspect the arguments of each command, consider the fact that those arguments exist because they allow the coder to change the arguments to get different results. With the `unnest_tokens()` command, the user could hypothetically choose to "unnest" our sentences into tokens of different length, for instance, sentences, n-grams (that is, multi-word phrases), or other units of text. In later chapters we will explore some of these features of the command "unnest." Alternatively, if the `unnest_tokens()` command were being applied to a different dataset where the column had a different name than "text," we could change the column name so that the command would work. In later examples, you will see the same commands applied to different datasets with different column names. For now, we will mainly use `unnest_tokens()` with the arguments "word" and "text."

Next, let's inspect the results of our unnesting using `head()`, which returns just the first five rows of a data frame.

```
##   sentence_id         word
## 1 S3V0108P0_0        which
## 2 S3V0108P0_0          had
## 3 S3V0108P0_0         been
## 4 S3V0108P0_0    prorogued
## 5 S3V0108P0_0 successively
## 6 S3V0108P0_0         from
```

Notice that the words in `tokenized_hansard` look just like the words in the first sentence of the first speech in hansard_1850. Those speeches have been split into words, but the words are still in the same order.

Notice that the command `unnest_tokens()` has done a little extra work for us: it has transformed all the words to lower-case and removed punctuation marks such as commas and periods. This

transformation will make the words easier to count.

As we inspect the results of unnesting, you should notice that the input was "tidy" in format – that is, it had two columns, one for "sentence_id" and one for "text," the sentences and sentence fragments of the original parliamentary speeches.

After working with the command unnest_tokens(), the resulting output, `tokenized_hansard`, also is "tidy" in format. That is, it has two columns, one for `sentence_id` and one for `word`. In this tidy format, where each word is on its each row, the computer can easily count the words in the dataframe.

The tidyverse command "count" groups like entries with like entries, counting every word. The command "count" takes one argument, the name of the column to work on – in this case, "word."

```
tokenized_hansard_counts <- tokenized_hansard %>%
  count(word) %>%
  arrange(desc(n))
```

With the command `arrange(desc(n))`, we tell the computer to arrange the results from greatest to least.

```
##   word       n
## 1  the 2588864
## 2   of 1404831
## 3   to 1126226
## 4 that  726686
## 5  and  706117
## 6   in  597753
```

We now have a frequency count of every word recorded in the speeches of Hansard, 1850-59. The initial results are unimpressive. In our dataset, as with most datasets of text, the most common

words are also the least interesting. For this reason, linguists compile lists of the most common words, which they call "stop words." Analysts often remove stop words from their results before analysis.

The tidytext package comes with a prepared list of stop words. We can remove them from our results.

```
## Joining with `by = join_by(word)`
```

```
head(stopworded_hansard_counts)
```

```
##         word      n
## 1        hon 125465
## 2      house 118861
## 3 government  95417
## 4       bill  77322
## 5      noble  73180
## 6       lord  68942
```

These results seem sensible for parliament. Some of them demonstrate the way that members of parliament refer to each other, for example, as "my noble lord" or "the hon[ourable] member [in question]." Speakers in parliament also routinely refer to the business of parliament, especially referencing the "house [of commons or lords]," the "bill" or "question" under debate, the state of the "country," and how much "time" is available for debate.

General words, of course, give us little insight into the business of parliament. Later chapters will give us richer tools for examining how the content of speeches changed over time. For now, we have successfully achieved the first step of analysis, the processing of text into words, and the counting of words.

Searching for Keywords

The Tidyverse provides several commands that we will use throughout this book to track individual words. Two commands, `filter()` and `str_detect()`, are used together to match and search data.

The command "filter" allows us to find an exact match in the data. Suppose that we want to inspect the tenth sentence of the parliamentary speeches, series 3, volume 108, which has the id number "S3V0108P0_10."

We use the command `filter()`, which takes two arguments: first, the name of the column we will search for ("word"), and second the name of the strong to match ("india"). Another symbol, the double equals sign (==), is used to tell the computer that we want an exact match.

```
stopworded_hansard_counts %>%
  filter(word == "india")
```

```
##    word     n
## 1 india 20816
```

```
stopworded_hansard_counts %>%
  filter(word == "france")
```

```
##     word    n
## 1 france 7146
```

```
stopworded_hansard_counts %>%
  filter(word == "jamaica")
```

```
##      word   n
## 1 jamaica 475
```

Filter can also be used with standard R operators, such as `%in%`. In this example, we create a list of words using the command "c" or "concatenate." Then, we search for the list of words:

```
female_list = c("girl", "woman", "girls", "women", "wife", "wives", "widow", "widows", "sister
```

```
stopworded_hansard_counts %>%
  filter(word %in% female_list)
```

```
##           word    n
## 1         wife 2222
## 2       sister 1153
## 3        woman 1063
## 4        women 1062
## 5        wives  479
## 6        widow  315
## 7       widows  254
## 8      sisters  240
## 9      females  195
## 10       girls  180
## 11        girl   76
## 12        aunt   62
## 13 grandmother   18
## 14       aunts   10
```

The command "filter" also works on quantitative information with comparative and boolean operations.

```
stopworded_hansard_counts %>%
  filter(n > 474 & n < 476)
```

```
##           word   n
## 1       cities 475
```

```
## 2   depositors 475
## 3      jamaica 475
## 4        pause 475
## 5       shared 475
## 6 unfavourable 475
```

Our next command, `str_detect()`, detects the presence or absence of a pattern and returns the boolean values `TRUE` or `FALSE`.

The following example asks the computer to move through each word in the list "female_list" and to check whether the value contains the string "girl."

```
female_list %>%
  str_detect("girl")
```

```
##  [1]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE
```

In practice, textual analysis often requires that we use both commands, `filter()` and `str_detect()`, together.

The following example searches hansard_1850 for the keyword "coal." Notice that `str_detect()` is nested inside the command `filter()`. Together, these commands tell the computer to filter for only those rows of the column 'text' that contain the keyword "coal."

```
hansard_coal <- hansard_1850 %>%
  filter(str_detect(text, "\\bcoal\\b"))
```

The result of our search is variable, `hansard_coal`, that contains only sentences from the 1850 debates that contain the word "coal."

```
head(hansard_coal)
```

```
##       sentence_id
## 1 S3V0108P0_14705
## 2 S3V0108P0_14958
## 3 S3V0108P0_15631
## 4 S3V0108P0_15634
## 5 S3V0108P0_15678
## 6 S3V0108P0_16749
##
## 1
## 2
## 3
## 4
## 5 They were subjected to great loss by the theft of small articles of coal, and they submit
## 6
```

Inspecting the line of code above, you may notice that we placed special characters – `\\b` – on both sides of the keyword. Together with the word "coal," these special characters are make up part of a "regular expression" or "regex." The `\\b` indicates word boundaries. In other words, we told `str_detect()` that we only wish to return text that contain an exact match with the word "coal." If we didn't use word boundaries when telling the computer to look for "coal," the computer would also search for every word containing "coal," including "coalition."

Finding and Comparing Keywords' Context

Typically, we want to do more than simply search for a keyword. We may also want to search for collocates, the words that form the context of a given keyword. We already have everything we

need to make this happen: we can search for a keyword, we can break up text into words, and we can count the words. Let's use our tools to search for the context of several keywords and compare them in visual form.

```
coal_context <- hansard_coal %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  filter(!str_detect(word, "[:digit:]")) %>%
  count(word) %>%
  arrange(desc(n))
```

```
## Joining with `by = join_by(word)`
```

As you may have noticed, you can run these many of functions independently or chained together with the 'pipe' from the maggirtr package: `%>%`. Using the pipe allows us to use many functions sequentially without storing intermediary variables.

Notice the line of code that reads `filter(!str_detect(word, "[:digit:]"))`. Here, we are filtering out any entries that contain digits as part of the word.

```
head(coal_context)
```

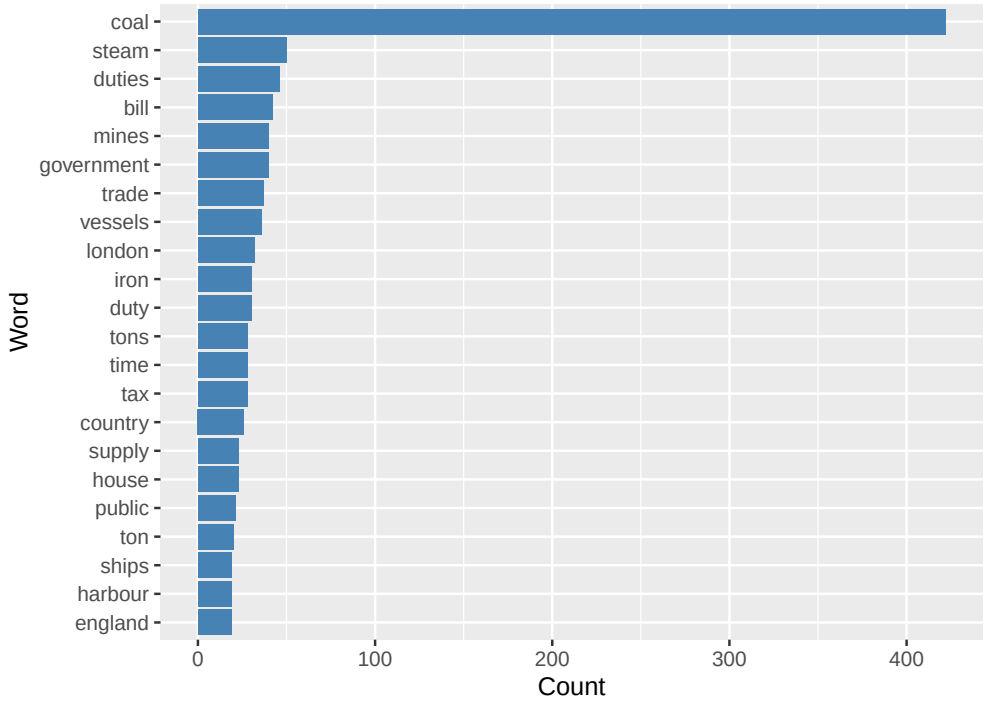
```
##         word   n
## 1       coal 422
## 2      steam  50
## 3     duties  46
## 4       bill  42
## 5 government  40
## 6      mines  40
```

```r
top_coal <- coal_context  %>%
  top_n(20)
```

```
## Selecting by n
```

```r
ggplot(data = top_coal) +
  geom_col(aes(x = reorder(word, n),
               y = n),
           fill = "steel blue") +
  coord_flip() +
  labs(title = "Figure 1.1: Top Words Occuring in Sentences Mentioning Coal",
       subtitle = "From the 1850 Hansard debates",
       x = "Word",
       y = "Count")
```

Figure 1.1: Top Words Occuring in Sentences Mentioning Coal
From the 1850 Hansard debates



Our analysis will gain nuance the more we compare near concepts. While coal was important to powering Britain's industrial revolution, perhaps the most controversial economic debates of the 1850s revolved around the issue of the Corn Laws, the system of taxation of wheat and other grains, which were originally introduced to protect British farmers. Working-class demands for cheap bread led to the repeal of the Corn Laws in 1846. Knowledge of these facts might lead us to ask: were coal and corn debated using the same language or different words?

```
corn_context <- hansard_1850 %>%
  filter(str_detect(text, "\\bcorn\\b")) %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  filter(!str_detect(word, "[:digit:]")) %>%
```

```
  count(word) %>%
  arrange(desc(n))
```

```
## Joining with `by = join_by(word)`
```

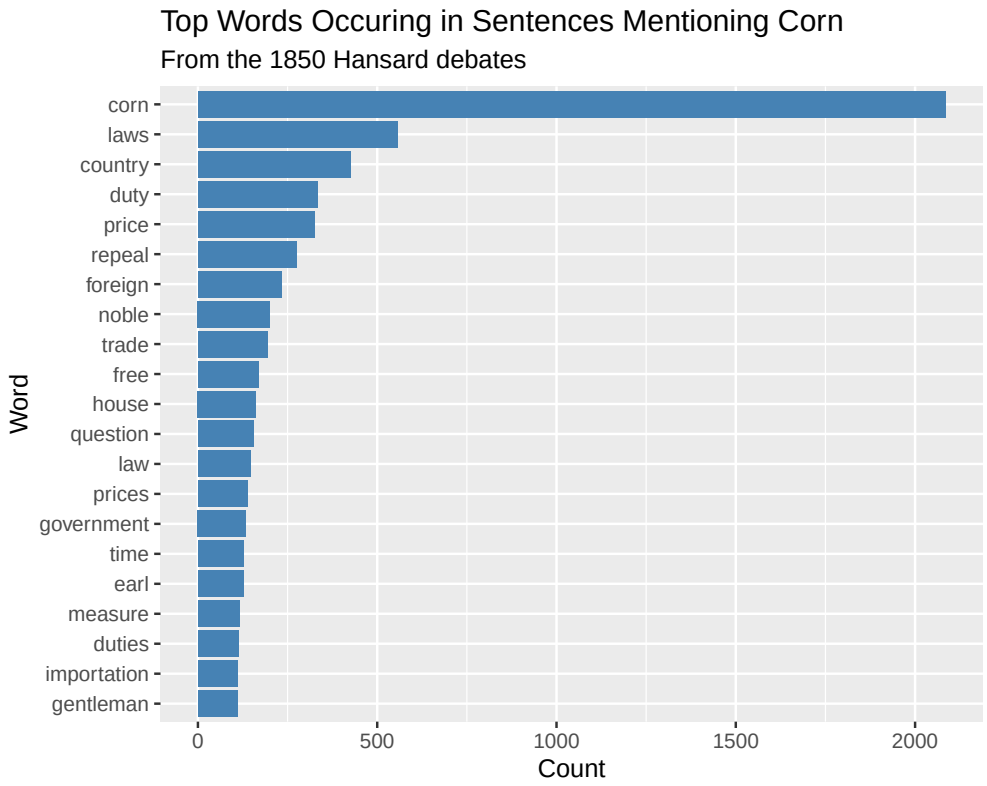
```
head(corn_context)
```

```
##      word    n
## 1    corn 2085
## 2    laws  556
## 3 country  426
## 4    duty  333
## 5   price  325
## 6  repeal  274
```

```
top_corn <- corn_context  %>%
  top_n(20)
```

```
## Selecting by n
```

```
ggplot(data = top_corn) +
  geom_col(aes(x = reorder(word, n),
               y = n),
           fill = "steel blue") +
  coord_flip() +
  labs(title = "Top Words Occuring in Sentences Mentioning Corn",
       subtitle = "From the 1850 Hansard debates",
       x = "Word",
       y = "Count")
```



Comparison between the words used in the context of coal and corn gives us a foothold for beginning to think about how taxation was discussed in different contexts. For instance, the context of both keywords reference the role of time, duties, taxes, and the government, but debates over corn are more linked to the question of Britain's dependence "foreign" powers, whereas debates over coal are more crucially related to the condition of "England" itself. We do not pretend that noticing this difference constitutes a major historical insight; further reading and research would be necessary for the analyst to decide whether comparing these two commodites as keywords is worthwhile, and which collocates offer insight.

Nevertheless, the student will have learned by this point the preliminary process with which most of the analysis in this book will start: loading data, inspecting data, breaking data into words, and counting those words. Later chapters will give us more tools – importantly the tools of counting data over time and understandign historical change – which can produce a richer form of analysis.

Finding a Word's Context using KWIC

Another way to explore a word's context is by using a tool termed "Keywords in Context" or "KWIC" for short. In this section we use `KWIC()` from the quanteda library and look at the sentence-level context for the word "corn" for February 1850.

To look at just the month of February, we need to load our decade subset that contains information about dates, called `debate_metadata_1850`, and join it with the data set containing the debates, `hansard_1850`. We will cover joins in greater detail in chapter 2.

```
data("debate_metadata_1850")
```

```
head(debate_metadata_1850)
```

```
##   sentence_id speechdate                        debate
## 1 S3V0108P0_0 1850-01-31       MEETING OF PARLIAMENT.
## 2 S3V0108P0_1 1850-01-31       MEETING OF PARLIAMENT.
## 3 S3V0108P0_2 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
## 4 S3V0108P0_3 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
## 5 S3V0108P0_4 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
## 6 S3V0108P0_5 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
```

```
hansard_1850_with_metadata <- left_join(hansard_1850, debate_metadata_1850)
```

```
## Joining with `by = join_by(sentence_id)`
```

```
head(hansard_1850_with_metadata)
```

```
##   sentence_id
## 1 S3V0108P0_0
## 2 S3V0108P0_1
## 3 S3V0108P0_2
```

```
## 4 S3V0108P0_3
## 5 S3V0108P0_4
## 6 S3V0108P0_5
##
## 1
## 2
## 3
## 4
## 5 Although the Act of Parliament directed that the proclamation should issue forthwith for
## 6
##   speechdate                        debate
## 1 1850-01-31       MEETING OF PARLIAMENT.
## 2 1850-01-31       MEETING OF PARLIAMENT.
## 3 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
## 4 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
## 5 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
## 6 1850-01-31 SCOTCH REPRESENTATIVE PEERS.
```

In the folllowing code we filter for speech dates that are on or after (greater than or equal to `>=`) February 1, 1850, and on or before (less than or equal to `<=`) February 28, 1850.

```
top_month <- hansard_1850_with_metadata %>%
  filter(speechdate >= "1850-02-01",
         speechdate <= "1850-02-28")
```

KWIC returns a concordance-style text output with the words that come directly before and after the keyword in a sentence. Quanteda's version of `KWIC()` works quickest if we first transform the data from a dataframe to a "corpus" object using `corpus()`. We tell Quanteda that the column we

want to operate on is called “text.”

When using `KWIC()` we can specify the keyword, the maximum number of words to display before and after the keyword, and we can also tell `KWIC()` to do a case insensitive match so that we match with uppercase or lowercase instances of our keyword.

```
library(quanteda)
```

```
## Package version: 3.2.4
## Unicode version: 14.0
## ICU version: 70.1
## Parallel computing: 8 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
```

```
my_corpus <- corpus(top_month, text_field = "text")


corn_kwic <- kwic(tokens(my_corpus),
                  "corn",
                  window = 5,
                  case_insensitive = TRUE)
```

```
head(corn_kwic)
```

```
## Keyword-in-context with 6 matches.
##   [text2032, 7]             cultivation of land, as | corn |
##  [text2036, 11]           south of Ireland dealt in | corn |
##  [text2038, 13] , deterred persons from cultivating | corn |
##   [text2536, 2]                                 on | corn |
##  [text2568, 23]         laws governing the price of | corn |
```

```
##  [text2578, 13]                to the repeal of the | corn |
##
##  ground, being given up
##  , and were the pro-
##  .
##  , would never have been
##  .
##  laws, for those laws
```

Critical Thinking With Collocates

When is collocate analysis the right tool? Researchers of conceptual history frequently use collocates to understand the subtle differences between synonyms. For instance, historian Ruben Ros has investigated the way that Dutch newspapers in the nineteenth century began to use terms such as "foreign," "overseas," and "strange" in the process of constructing a narrative that increasingly linked the dangers of foreign influence and suspicion of ethnic minorities.

With collocate analysis alone, we cannot trace discourses over time – a crucial step in understanding the construction of political and cultural positions. But we can begin to tease apart the meanings of closely-related ideas at various points in the past. Let's recreate Ros's case study in miniature here, asking the question: how did British members of parliament in the 1850s talk about foreigners and colonial subjects?

The following code uses a "for loop" to cycle through a list of related words. We will look at for loops in greater detail later in the book, but the basic concept should be easy to grasp. For each of the words in the list "wordlist," the for loop performs the same set of operations: it searches hansard_1850 for each keyword, then it counts the words in context, and it graphs the top results. The for loop allows us to produce a series of related graphs in succession without writing out the same instructions multiple times.

```
wordlist <- c("foreign", "overseas", "colonial", "savage", "native", "barbarian")

for (word in wordlist) {

  filtered_hansard_1850 <- hansard_1850 %>%
    filter(str_detect(word, paste0("\\b", word, "\\b"))) %>%
    unnest_tokens(word, text) %>%
    anti_join(stop_words) %>%
    filter(!str_detect(word, "[:digit:]")) %>%
    count(word) %>%
    top_n(20)

  ggplot(filtered_hansard_1850) +
    geom_col(aes(x = reorder(word, n),
                 y = n),
             fill = "steel blue") +
  coord_flip() +
  labs(title = "Top Words Occuring in Sentences Mentioning ", toupper(word),
       subtitle = "From the 1850 Hansard debates",
       x = "Word",
       y = "Count") }
```

```
## Joining with `by = join_by(word)`
## Selecting by n
## Joining with `by = join_by(word)`
## Selecting by n
## Joining with `by = join_by(word)`
```

```
## Selecting by n
## Joining with `by = join_by(word)`
## Selecting by n
## Joining with `by = join_by(word)`
## Selecting by n
## Joining with `by = join_by(word)`
## Selecting by n
```

Producing many graphs in succession can be useful for the process of preparing material for careful comparison, which is one of the major skills that humanists use when considering the subtle structures of meaning and difference that define cultural and political conversations. A careful comparison between the charts above offers the beginnings of a nuanced analysis of overlapping terms used by British members of parliament to refer to people and forces from beyond Britain.

In the results, we see several closely related terms which compose a field of interrelated meanings. Together, these terms sketch out the diversity of contrasting ways that British members of parliament spoke about the world outside of Britain. Some of the attributes ascribed to the foreign were neutral in value – for instance, the "foreign" was mainly spoken about in terms of laws, duties, trade, and protection, a world in which different nations "competed," but generally not a space of moral inferiority and superiority. By contrast, terms such as "savage" and "barbarian" telegraphed intellectual and ethical judgments (for instance, "brutal") onto cultures outside of Britain, typically associating inferiority with racial identity. With these terms, a set of binaries is set up, dividing the world into "barbarian" cultures and "christian" "civilisations," a distinction that may have been analyzed in terms of "rights," "treaties," "war," "rents," "passions," "liberty," and "energies. The distinctions between the"foreign" and the "barbarian" are invoked in relationship to the same abstractions we saw typifying the "foreign," including "law," the "country," and "treaties." We also see the "barbarian" being invoked in discussions of many commodities, including "woolens," "forage," and "rent." Indeed, the frequent invocation of law, rights, and foreignness alongside discussions

of the barbarian and the savage suggests that the judgments ascribed to race actually bled into conversations about law, tariffs, trade, and protectionism, perhaps as justification for invoking British superiority.

Some readers may assume that we are performing contemporary prejudices about racism, but the point of this exercise is that it is grounded in a detailed and careful examination of the words counted by computational methods. The paragraph above is not concocted out of thin air; it is actually an objective description of the prejudices on display in a quantitative reading of how British members of parliament spoke about the world beyond their nation. We emphasize that this analysis of the related language of the "foreign" and the "barbarian" is not concocted on the basis of our readings of contemporary theories about racism, but is actually derived from a careful reading of language, as well as immersion in writing about the history of empire.

In general, descriptions of word counts and arguments made on the basis of word counts tend to be more persuasive when they take on every word in a list. Critical thinking about the meaning of proximate keywords is enhanced when the analyst slows down, taking the trouble to examine the distinctiveness of each individual keyword and its context. In a further analysis, the scholar might ask questions that require examining the results in more detail. They might ask: What biases does each term convey? Where do the terms overlap? What new information is encapsulated by some words but not others? A thorough examination would trace the keyword-context pairs back to their original context in sentences and speeches on the page, showing how speakers used these words to formulate actual arguments with consequences for the development of nations and their economies.

Examining collocates provides the basis for critical thinking about the changing and overlapping meaning of shared concepts in the human past. Skillful analysts can use approaches of this kind to examine the meaning of language about categories beyond those of gender, race, nationality, and commodities, asking questions about the intellectual categories that governed how members of parliament understood governance itself, economics, truth, science, virtue, economics, and reason

itself. In every case, the approach would be the same: to examine not merely one or two keywords, but many related keywords, drawing attention to the subtle overlap and differences between the terms in usage, and how those overlapping concepts together work to produce a field of meaning.

Critical Thinking About Data and its Limits

Another fruitful avenue for critical thinking is the question of which problems are suited to textual analysis and which are not. Skillful analysts are wary of asking the wrong question with the wrong tool. Among the techniques that we have learned in this chapter is the ability to load and inspect data. Inspecting the data allows the analyst to notice issues caused by computational processing of text that may cause issues down the line unless the analyst is aware of them.

Looking at the first lines of `hansard_1850`, you might ask yourself these questions: are all of the lines in the "text" field actually sentences? Are all the words in the same style with respect to capitalization?

Here is another command which gives us a little more control over what we see. The square brackets tell R that we want to look at a certain column and row or group of columns and rows. The structure of the syntax is this: [row, column]. The syntax "1:10" in the "row" field tells r to display rows 1 to 10. We put "2" in the "column" field to tell R to display

```
hansard_1850[1:10, 2]
```

```
##  [1] ", which had been prorogued successively from the 1st of August to the 9th of October,
##  [2] "The Parliament was opened by Commission, the LORDS COMMISSIONERS present being the LO
##  [3] "called attention to a great omission of their duty on the part of Ministers, with resp
##  [4] "At pre-sent there were two vacancies in the representative Peers of Scotland, in conse
##  [5] "Although the Act of Parliament directed that the proclamation should issue forthwith
##  [6] "The Peerage of Scotland was not therefore represented in Parliament at present as it
##  [7] "He wanted to know what his noble Friend the President of the Council had to urge in de
```

```
##  [8] "said, that the Government were not to blame for the omission, as they had not receive
##  [9] "said, the explanation of the noble Marquess was unsatisfactory."
## [10] "The clause in 5 & 6 Anne, c. 8, directed what should be done in the case of vacancies
```

Perhaps you have noticed that some of the rows in the "text" field appear not to be complete sentences. We can use square brackets to investigate further.

```
hansard_1850[1, 2]
```

```
## [1] ", which had been prorogued successively from the 1st of August to the 9th of October, f
```

Notice that your view of the data is cut off in this view. To view the whole text, double-click on the cell, copy and paste the text into a word processor. You should see this content:

", which had been prorogued successively from the 1st of August to the 9th of October, from thence to the 20th of November, thence to the 16th of January, and from thence to the 31st January, met this day for despatch [sic] of business."

The first row appears to be a description of the opening of parliament. Technically, it is a commentary by the printer, not a line of debate. The convention of discussing parliament's timetables was in place in 1850, but we don't know if the printer always printed this line through the entire century. We should be aware of this convention and curious about it if we find ourselves counting months or discussions of words such as "business."

Let's keep inspecting the data, this time turning to the third row in our dataset.

```
hansard_1850[3,2]
```

```
## [1] "called attention to a great omission of their duty on the part of Ministers, with respe
```

Here, we see another sentence fragment:

"called attention to a great omission of their duty on the part of Ministers, with respect to the privileges of their Lordships, which might and ought to have been avoided."

The sentence fragment has resulted from the computer attempting to generate data while working on another convention of printing: the fact that the publisher of Hansard for much of the century gave the name of the speaker followed by a description of the airing of the speech. On the page, we would be given the name of a speaker or their title, for example, "The Lord Chancellor" or "Mr. Gladstone." The passage of text would therefore read, [the speaker] "called attention to a great omission. . . " – in other words, it would not be a sentence fragment, so much as a journalistic description of a speaker. The actual speech probably began with a statement something like this: "It is a great omission of their duty on the part of Ministers[. . .]."

In other words, we are looking at an error that resulted from a mismatch between the nineteenth-century printed record and what we asked the computer to do with the text of that record. In compiling the data, we asked the computer to separate all of the speaker names into one "field" of data, which we stored in a separate dataset, to be accessed in Chapter 2. We have sorted all of the text into the "text" field, which is held in hansard_1850. The computer doesn't know how to process the journalistic description of a speech rather than a normal speech.

Looking at the data presents an opportunity for critical thinking. Analysts of text as data routinely need to inspect their data and make sure that the data's quality and density are sufficient to support claims that they might make.

Already, we can see that the data has certain features that might interfere with certain kinds of queries, and we should already be thinking about what these limits might be. Yet the convention of journalistic reporting on speeches can cause trouble for the analyst if the analysts aren't aware of what has changed. The conventions of describing speeches changed over the course of the century; at the beginning of the century, journalistic descriptions were more common. Later in the century, the conventions changed, and many more speeches were reported near verbatim without journalistic commentary. We should be aware of conventions of this kind if we count words which may be related to journalistic observations such as "attention," "spoke," "declared," and so on. Otherwise we may

be tempted to interpret changes to the publisher's conventions of printing speeches as evidence for changing ideas about politics.

Whether or not data quality will interfere with our ability to make inferences from the data depends on what we are asking. If we are counting mentions of the word "duty" or the rest of the words in the content of the speech, then our counts of the words will be correct. If we contrive to study the history of journalistic observation by tracking sentence fragments in the dataset, this data will support such a reading. But we should be wary making inferences about any kind of inquiry where there is a risk of overlap between substantive discussions and journalistic observation.

Exercises

Expand your grasp of the code and your powers of critical reflection by trying the following exercises:

1) Alter the code above to compare the context for the keywords "girl" and "woman." Do you expect the context to be similar for both? Does anything surprise you about the results?

2) Use your powers of inspecting data to move from the keyword context back to the individual sentences. Cut and paste twenty sentences from the context of each keyword into a word processor. What do you learn from these sentences that you did not learn from counting keywords? Does your analysis become more perusasive when quantitative graphs are accompanied with readings of particular speeches where the words are examined in their original context?

3) We have examined 'coal' and 'corn' in the 1850s. Now, search for the words 'coal' and 'corn' in dataset `hansard_1820`. Were they being used in the same way? What changed?

4) Above, we created the list of words called `female_list`. Use the operator `%in%` with the command `filter()` to create a list of the words in the context of discussions of women. Create another list called `male_list` and do the same for discussions of men. Create visualizations for the top terms used in discussions of men and women.

5) Reflect on your work so far – in which you have compared words for gender, the concept of the foreign, and the context of discussions of different commodities. You have made comparisons over time and comparisons of lists. Which of the comparisons do you find the most productive of insight about how certain words were being used in political debate? Are you persuaded by large numbers? Are you persuaded by subtle variations in the context of different words? Write a paragraph explaining for yourself wherein constitutes the most persuasive usage of comparisons between keyword in context.

6) For the visualizations of the terms associated with different terms for the "foreign," tell us: What biases does each term convey? Where do the terms overlap? What new information is encapsulated by some words but not others? What do we learn from this exercise about the prejudices of Britain's members of parliament in the 1850s?

Chapter 2: Investigating the Memory of Events Using Ngrams, Controlled Vocabulary and Joins

In this chapter, analysts will wield their skills to explore questions about how discourses, conceptualizations, events and the memory of the past shape current-day political change. It is common in political speeches to reference the events of the past – whether American politicians reference the Tea Party or British parliamentarians reference the events of the Second World War. Speakers in parliament also acknowledge contemporary public protests as a source of political legitimacy. But the number, intensity, and specific events referenced change over time. What can we learn about how political reforms were made in the past from reviewing how political actors in the past referenced events?

We will explore questions such as these: Which historical and contemporary events were referenced the most by members of parliament in the era of the Second Reform Act of 1867, when working-class people got the vote for the first time? In the lead-up to the Great Reform Act of 1832, when the middle class got the vote in Britain, which historical references were on the table? Which members of parliament contributed the most to the debate about the Abolition of Slavery in 1833, and of these speakers, whose language reflected the most on the lived experience of toil and cruelty in the system of slavery?

This chapter builds the readers' sense of how to work with text to create insight about history. It introduces counting for n-grams, or multi-word phrases. It also introduces the use a "controlled vocabulary," or expert-provided list of important words and phrases, to explore which events were

spoken about in parliament during the 1830s.

This chapter also builds upon the previous chapter to introduce readers who are novices at working with data to several of the basic data processing strategies in tidyverse R that are used to manipulate data for analysis. A common task in data analysis is to summarize characteristics of a data set and glean insight into basic features of the data. One basic concept in data processing is the concept of "joining" data sets. In performing a "join," an analyst welds two data sets into a combined data set, a process that will allow us to explore parliamentary texts together with metadata about speakers, dates, and significant events. This chapter will teach students of data to work with event-based data such as information about years, months, and dates.

As with the previous chapter, this chapter will emphasize the skills of moving from distant reading back to a close reading of historical speeches in context, arming the analyst with the tools of deciding for themselves which apparent patterns identified by text mining are supported by the reading of texts. It will emphasize that researchers work with text in an iterative process, moving from counting words and visualizing trends to reading, and then often repeating this deductive process. Reading leads to specifying historical questions, which may mean more counting, more visualization, and more guided reading in pursuit of a historical analysis.

While this chapter does not model any such process from start to finish, it models the iterative process of working with a guided vocabulary and reading words in context with the idea of teaching a historical method of inquiry about how and why events happen, alongside the basics of data science for historical research.

Working With Debate Metadata

We have previously seen how to load the text from a single decade of the Hansard debates. But what happens if we want to work with not only the text, but also important information like the name of the speaker who gave the speech and the date on which the speech was given?

Let's load data from the 1860s speeches, recalling that one of the columns, sentence_id, refers to the number of each sentence and its location in the nineteenth-century printed volumes of Hansard's parliamentary debates, referenced by series ("S"), volume ("V"), and page ("P"). Sentences are listed in the order in which they appeared on the page.

```
library(hansardr)


data("hansard_1860")
```

```
head(hansard_1860)
```

```
##   sentence_id
## 1 S3V0156P0_0
## 2 S3V0156P0_1
## 3 S3V0156P0_2
## 4 S3V0156P0_3
## 5 S3V0156P0_4
## 6 S3V0156P0_5
##
## 1
## 2
## 3
## 4 My Lords, it is with diffidence that I rise to address your Lordships for the first time,
## 5
## 6
```

The hansardr package includes detailed contextual information for each of these speeches, using the same system of references that we saw in the previous data set – series, volume, page, and sentence number. Here, however, there is no "text" column, but rather a series of columns that illuminate

the "fields" of each sentence: the "speechdate" on which the speech was given, the official title of the "debate" in which the sentence was spoken, the "speaker" of the sentence, and a special number for each speaker, "disambig_speaker," which allows the user to disambiguate speakers with very common names.

```
data("debate_metadata_1860")
```

```
head(debate_metadata_1860)
```

```
##   sentence_id speechdate                                    debate
## 1 S3V0156P0_0 1860-01-24                 MEETING OF THE PARLIAMENT.
## 2 S3V0156P0_1 1860-01-24                       THE QUEEN'S SPEECH.
## 3 S3V0156P0_2 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
## 4 S3V0156P0_3 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
## 5 S3V0156P0_4 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
## 6 S3V0156P0_5 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
```

In the hansardr package, we have chosen to store the text information ("hansard_1860") separately from the contextual information about each speech ("debate_metadata_1860"), as well as the information about each hard copy file containing the speeches. This is because the data in each dataset is quite large in terms of computer memory. Furthermore, many operations – for instance, counting words per decade – require just one dataset, not both.

As you work with the datasets, however, many kinds of analysis will require working with both sets of data together. We will use the function `left_join()` to pair the annotated information about speech context with each of the speeches. A left join returns every record from the left-hand data frame, and all the matched records from the data frame on the right.

Notice that the resulting dataset, `hansard_corpus_1860`, has information about speaker and date for each sentence of text.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
hansard_1860 <- left_join(hansard_1860, debate_metadata_1860)
```

```
## Joining with `by = join_by(sentence_id)`
```

```
head(hansard_1860)
```

```
##   sentence_id
## 1 S3V0156P0_0
## 2 S3V0156P0_1
## 3 S3V0156P0_2
## 4 S3V0156P0_3
## 5 S3V0156P0_4
## 6 S3V0156P0_5
##
## 1
## 2
## 3
## 4 My Lords, it is with diffidence that I rise to address your Lordships for the first time,
```

```
## 5
## 6
##   speechdate                                        debate
## 1 1860-01-24                    MEETING OF THE PARLIAMENT.
## 2 1860-01-24                           THE QUEEN'S SPEECH.
## 3 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
## 4 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
## 5 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
## 6 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY'S SPEECH.
```

In programming, "joins" are an important concept for manipulating data held in different formats. The process of "joining" data refers to the process that allows analysts to fuse together two or or more data sets into a new one.

In the foregoing example, both `hansard_1830` and `debate_metadata_1830` shared the sentence_id column (or "field") in common. Having a common, stable identifier allows the computer to match the speaker and date of each sentence with the text of the speech. The common, stable identifier is the basis on which the two data sets are "joined" into a single, new data set.

With the annotated data set, we can inspect the parliamentary speeches with greater nuance. For example, we can probe the words invoked in parliament in the months leading up to August 15, 1867, when most working-class men in Britain finally achieved the right to vote.

Multi-Word Phrases

Very frequently, the analyst will want to count not merely the words pronounced by members of parliament, but also multi-word phrases. Multi-word phrases contain formulations of concepts that had powerful political, social, and cultural meaning to users, for example, "the will of the people," a phrase that has been invoked by both radicals and conservatives to support extremely different

imaginations of the people. That is, speakers who invoked "the will of the people" might have be bolstering the importance of their speech by gesturing towards their status as a representative of the workers whose labor is the source of all political legitimacy. On the other hand, some speakers might have used the same phrase – "the will of the people" – to kindle an imagination of parliament as a zone a traditional folk whose national identity legitimizes a politics of excluding from citizenship or rights people with certain racial or immigration status. The question becomes: which version of the people was being represented in the discourse in question?

The analyst cannot anticipate in advance how such a phrase might have been used. The only possible way of gaining knowledge from a count of phrases is to "validate" one theory or the other by gaining more information.

We could, in theory, contrive computational approaches to help us discern why people invoked "the will of the people." But in text mining for historical analysis, we often want to move from identifying a data-driven phenomenon to gaining more information about historical context in a traditional way, that is, by reading the actual speeches out of which the phrase count is made. Only then can we be sure that we won't miss some of the rich nuances suggested by speakers in how they invoke a particular phrase – and even then, we must be sure to read with care and sensitivity.

Accordingly, in the sections below, we will begin by counting phrases, but that is not where we will end. We will follow our instruction in counting by showing the readers how to return to the original speeches to read more deeply. An exercise at the chapter's end will ask what the reader learned.

Finding Bigrams

Finding multi-word phrases begins with tokenization, a concept that we have seen before. This time instead of tokenizing the data into single words, we will tokenize the data into bigrams, or sequential, two-word phrases.

Here is some code for filtering the speeches of the 1860s for the first months of 1867 before the vote

on the Second Reform Act, which gave Britain's urban working-class men the right to vote.

The second block of code creates a variable called `bigrams_1867`. Note the line of code with the command `unnest_tokens()`, the same command that we previously used to break up text into individual words. Here, however, we have made a slight adjustment. The argument `token` is set as `ngrams`, and n is set as 2. Together, these two arguments tell R to search for two-word phrases, or "bigrams."

```
library(tidytext)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
hansard_1867 <- hansard_1860 %>%
  mutate(year = year(speechdate)) %>%
  filter(year == 1867,
         speechdate <= 1867-08-15)

bigrams_1867 <- hansard_1867 %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)
```

```
head(bigrams_1867)
```

```
##   sentence_id speechdate              debate year         bigram
## 1 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867   being seated
## 2 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867      seated on
```

```
## 3 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867         on the
## 4 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867     the throne
## 5 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867 throne adorned
## 6 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867   adorned with
```

Note that the bigrams extracted overlap, as if the same sentence had been subdivided several times. This is exactly how ngrams work.

Counting Bigrams Let's count the top bigrams and remove any bigrams containing stop words. Here we use Silge & Robinson's code to split a bigram into separate words, labeled "first" and "second," using the `separate()` function. After splitting up the bigram, we can use `anti_join()` to eliminate any bigrams with stopwords in either the first or second column. Finally, we filter to eliminate any bigrams with digits instead of letters.

```
bigrams_1867 <- bigrams_1867 %>%
  separate(bigram, into = c("first", "second"), sep = " ", remove = FALSE) %>%
  anti_join(stop_words, by = c("first" = "word")) %>%
  anti_join(stop_words, by = c("second" = "word")) %>%
  filter(str_detect(first, "[a-z]") &
           str_detect(second, "[a-z]"))
```

```
head(bigrams_1867)
```

```
##   sentence_id speechdate              debate year          bigram     first
## 1 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867  throne adorned    throne
## 2 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867 regal ornaments     regal
## 3 S3V0185P0_1 1867-02-05 THE QUEEN'S SPEECH. 1867   robes sitting     robes
## 4 S3V0185P0_2 1867-02-05 THE QUEEN'S SPEECH. 1867 robes commanded     robes
## 5 S3V0185P0_2 1867-02-05 THE QUEEN'S SPEECH. 1867 gentleman usher gentleman
```

```
## 6 S3V0185P0_2 1867-02-05 THE QUEEN'S SPEECH. 1867       black rod     black
##      second
## 1   adorned
## 2 ornaments
## 3   sitting
## 4 commanded
## 5     usher
## 6       rod
```

To count the resulting bigrams, we must paste the first and second words back together into a new column. The command `mutate()` creates a new field using the content of its argument, in this case, the instructions to glue together the information in the first column and second column, separated by a space.

```
top_bigrams_1867 <- bigrams_1867 %>%
  mutate(bigram = paste0(first, " ", second)) %>%
  count(bigram) %>%
  top_n(100) %>%
  arrange(desc(n))
```

```
## Selecting by n
```

```
head(top_bigrams_1867)
```

```
##                 bigram    n
## 1           noble lord 1808
## 2           noble earl 1544
## 3         noble friend 1030
## 4 majesty's government 1014
```

```
## 5       roman catholic  968
## 6          reform bill  742
```

In this list, we see several contemporary rhetorical figures that stem from the practice of referring to other speakers in parliament as "my noble friend" or "my learned friend." We also see allusions to the debates over representation and the question of who would have the vote, especially in the phrases "household suffrage," referencing historical argument for granting middle-class men the right to vote with the Reform Bill of 1832, and "roman catholic," a reference to the fact that Roman Catholics could already vote until the Reform Act of 1834. Only in 1867 did those working-class men get the vote in Britain, and these earlier precedents were frequently alluded to as a justification.

We also see a reference to "public meetings." Why did speakers in parliament invoke public meetings in the lead-up to the Second Reform Act? To answer this question, we must again read.

```
my_reading <- hansard_1867 %>%
  filter(str_detect(text, "public meetings"))
```

```
head(my_reading$text)
```

```
## [1] "I am aware that many of the public meetings have declared in favour of manhood suffrag
## [2] "They are often drawn up with considerable ability; but they bear the mark, I think, of
## [3] "Speeches at public meetings, and the discussions of the press, great as their influenc
## [4] "I know I shall be asked, from what I have seen in the press, and what has been said at
## [5] "Have you not heard or read what has been said at public meetings of every kind?"
## [6] "There have been more than 1, 000 public meetings; at every one, the doors were open, a
```

A cursory reading of the sentences suggests that members of parliament were well aware of massive public meetings where more than a thousand members of the public showed up support the vote for working-class people. We read of demonstrations around British Empire, even in Nova Scotia. We read of members of parliament challenging each other to meet the demands of the public: "Have you

not heard or read what has been said?" We have only skimmed a few sentences, but the material in these 126 rows gives us what we need to understand how members of parliament talked about public protests in these crucial months of 1867.

Counting Multi-Word Phrases Text mining also allows us to look for multi-word phrases. Let's look at 5-word phrases from the year 1867. We can also add a line to filter the ngrams for the presence of the word "people," looking for voices that refer to democracy, for instance, "the will of the people," an important concept in the years leading up to 1867, when working-class men in cities got the right to vote for the first time. The results will allow us to ask the question: What are the most frequently-invoked phrases involving "the people"?

We can use a slight adjustment of the code we used above to count bigrams to find ngrams of any length. Here is code for finding common five-word-phrases, filtering them, counting them, and finding the top 50 examples:

```
fivegrams_1867 <- hansard_1867 %>%
  unnest_tokens(ngram, text, token = "ngrams", n = 5)


people_ngrams <- fivegrams_1867 %>%
  filter(str_detect(ngram, "people")) %>%
  count(ngram) %>%
  top_n(50)
```

```
## Selecting by n
```

```
head(people_ngrams)
```

```
##                          ngram  n
## 1    feelings of the people of  8
## 2     great body of the people 32
```

```
## 3 great majority of the people 13
## 4     great mass of the people 15
## 5 majority of the irish people 10
## 6    majority of the people of 27
```

We see a great many political concepts invoked about what members of parliament believed they were working for – the people of Ireland, England, or the metropolis, their representation, condition, education, minds, interests, feelings, rights, and welfare. An entire vocabulary of social engagement and political will was being spelled out.

Reading further down this list might give us more hints about the language with which speakers urged the principle of representation. Again, the careful analyst would trace these words back to their original context, using the original text of the speeches to develop an argumnet about why a phrase like "the minds of the people" was used.

```
minds <- hansard_1867 %>%
  filter(str_detect(text, "minds of the people"))
```

```
head(minds$text)
```

```
## [1] "It is satisfactory to learn that these measures afforded great relief to the minds of t
## [2] "Its debates have been the principal means by which political wisdom, and the results ar
## [3] "Its debates command nothing like the same interest and attention, and exercise far less
## [4] "Gentleman the Question of which he had given notice, and was sure the Government would
## [5] "He believed, however, that the partial administration of justice in Ireland sometimes c
## [6] "The question is whether the impression is or is not to be conveyed to the minds of the
```

When speakers referenced the "minds of the people," they frequently depicted popular politics as a form of collective political achievement to which ordinary people contributed the rational evidence of their own lived experience. We find the phrase connected to ideas about moral "duty" and political

"harmony." We also see it invoked in reference to Ireland, India and Scotland.

These images of the peoples' collective wisdom contrast against the bias expressed by elites in another age, which warned against democracy as a form of potential despotism governed by ignorance. It is not surprising that the months leading into the Reform Act of 1867 would see eulogies to the wisdom of popular politics, but the phrase "the minds of the people" offers one entry-point for analysis.

Using Bigrams to Find Mentions of Events

For many kinds of analysis involving text, the analyst will want to bring in outside information not contained in the words, speakers, or dates of the provided datasets. For example, a historian working with data is aware of the names of famous events from British history. They may want to track how many times the Magna Carta, Spanish Inquisition, or Glorious Revolution are mentioned in parliament. In this case, the user will employ an "index." The hansardr package comes with one such index, a list of important events in the British past with the year of their historical occurrence.

This exercise invites readers to investigate the idea that representations of memory change over time, using the approach known in data analysis as a "controlled vocabulary," where the analyst searches for a set group of phrases.

In our case, the controlled vocabulary will be a list of events from British history. We will use a controlled vocabulary to measure Parliamentarians' references to past events in the 1860 debates. We are intentionally choosing words such as "riot" or "meeting" that might collect evidence of parliamentary speakers referencing public meetings like those we saw evidence of above. We are also curious about how those references compare with references to famines, wars, strikes, exhibitions, and other contemporary and historical events.

```
data("hansard_1860")
data("debate_metadata_1860")
```

```
hansard_1860 <- left_join(hansard_1860, debate_metadata_1860)
```

```
## Joining with `by = join_by(sentence_id)`
```

```
hansard_1867 <- hansard_1860 %>%
  mutate(year = year(speechdate)) %>%
  filter(year == 1867,
         speechdate <= 1867-08-15)
```

```
bigrams_1867 <- hansard_1867 %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)


bigrams_1867 <- bigrams_1867 %>%
  separate(bigram, into = c("first", "second"), sep = " ", remove = FALSE) %>%
  anti_join(stop_words, by = c("first" = "word")) %>%
  anti_join(stop_words, by = c("second" = "word")) %>%
  filter(str_detect(first, "[a-z]") &
           str_detect(second, "[a-z]"))
```

```
head(bigrams_1867)
```

```
##   sentence_id speechdate              debate year          bigram     first
## 1 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867  throne adorned    throne
## 2 S3V0185P0_0 1867-02-05 THE QUEEN'S SPEECH. 1867 regal ornaments     regal
## 3 S3V0185P0_1 1867-02-05 THE QUEEN'S SPEECH. 1867   robes sitting     robes
## 4 S3V0185P0_2 1867-02-05 THE QUEEN'S SPEECH. 1867 robes commanded     robes
## 5 S3V0185P0_2 1867-02-05 THE QUEEN'S SPEECH. 1867 gentleman usher gentleman
## 6 S3V0185P0_2 1867-02-05 THE QUEEN'S SPEECH. 1867       black rod     black
##      second
```

```
## 1   adorned
## 2 ornaments
## 3   sitting
## 4 commanded
## 5     usher
## 6       rod
```

```
pattern1 = c('riot', 'meeting', 'famine','revolt', 'exhibition', 'massacre', 'strike', 'war')


events_1867 <- bigrams_1867 %>%
  filter(second %in% pattern1) %>%
  mutate(bigram = paste0(first, ' ', second)) %>%
  select(bigram, speechdate)
```

```
head(events_1867)
```

```
##            bigram speechdate
## 1  sanguinary war 1867-02-05
## 2 dreadful famine 1867-02-05
## 3        late war 1867-02-05
## 4       civil war 1867-02-05
## 5       civil war 1867-02-05
## 6       china war 1867-02-08
```

If we look carefully at the events list so generated, we will recognize only a few of these references as actual events. Many of the bigrams including our controlled vocabulary are simply descriptions, for example, "disorderly riot." A few are references to actual events and meetings specified by the name of an event, for instance, the "Bristol Riot" or a "Yorkshire meeting." Far more frequent are general references to a "public meeting" or "recent meeting." All of this is interesting, of course,

because as we have seen members of parliament leaned on each other to acknowledge the public sentiment for expanding the vote.

How did references to meetings vary over time? By this time, you will have noticed that searching for a controlled vocabulary, visualization, guided reading, and deliberation about the meaning of the data is an iterative process that we engage over many rounds – not an automatic process where the researcher looks for a word like "riot" and finds simple and meaningful results. Let's search for our key phrases and visualize them. Note that the code below uses a new command for faceted counting – `group_by()` – which we will investigate in the next chapter.

```
pattern2 = c('yorkshire meeting', 'clontarf meeting', 'recent meeting', 'public meeting', 'par


top_events_1867 <- events_1867 %>%
  filter(bigram %in% pattern2) %>%
  group_by(bigram) %>%
  summarize(total = n()) %>%
  top_n(10)
```

```
## Selecting by total
```

```
head(top_events_1867)
```

```
## # A tibble: 6 x 2
##   bigram           total
##   <chr>            <int>
## 1 affghan war          3
## 2 bristol riot         1
## 3 civil war           31
## 4 clontarf meeting     1
```

```
## 5 crimean war         55
## 6 irish famine         5
```

```
top_events_1867_w_speech_metadata <- top_events_1867 %>%
  left_join(events_1867, by = "bigram") %>%
  mutate(month = month(speechdate)) %>%
  count(bigram, speechdate) %>%
  ungroup()
```

```
## Warning in left_join(., events_1867, by = "bigram"): Each row in `x` is expected to match a
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

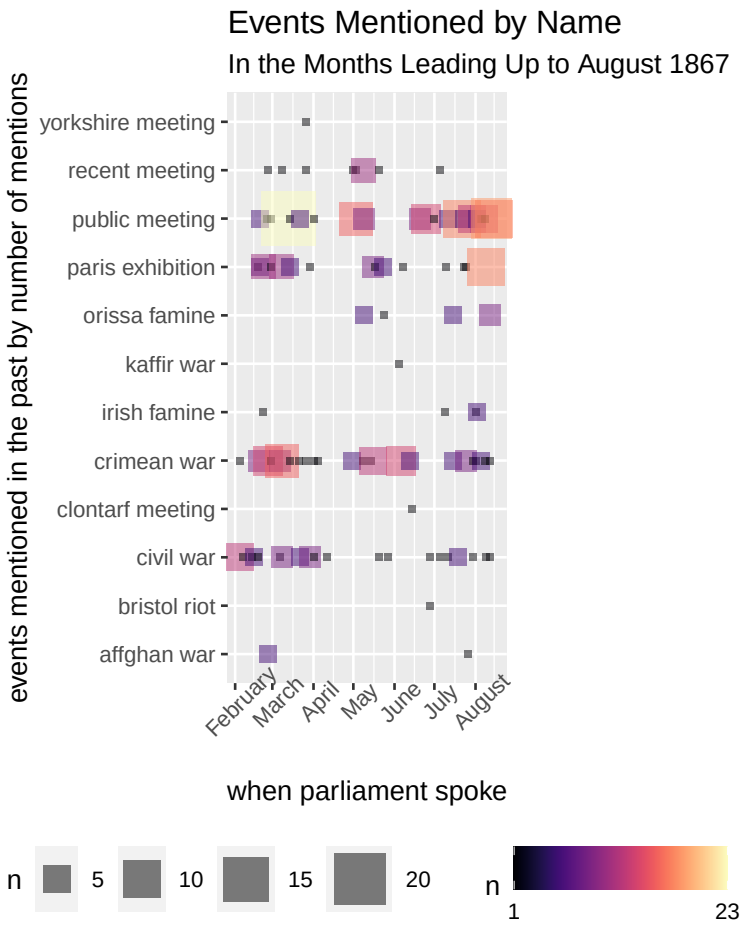
```
head(top_events_1867_w_speech_metadata)
```

```
## # A tibble: 6 x 3
##   bigram       speechdate     n
##   <chr>        <IDate>    <int>
## 1 affghan war  1867-02-26     2
## 2 affghan war  1867-07-26     1
## 3 bristol riot 1867-06-28     1
## 4 civil war    1867-02-05     5
## 5 civil war    1867-02-07     1
## 6 civil war    1867-02-14     1
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
ggplot(top_events_1867_w_speech_metadata,
       aes(x = speechdate,
           y = bigram,
           size = n,
           color = n)) +
  scale_color_viridis(breaks = round,
                      trans = "log",
                      option = "A",
                      discrete = F,
                      direction = 1) +
  scale_x_date(date_breaks = "1 month", date_labels = "%B") +
  scale_size_continuous(range = c(1, 10)) +
  geom_point(alpha = .5,
             shape = 15) +
  coord_cartesian(clip = 'off') +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 45),
        plot.margin = unit(c(1, 20, 1, 1), "lines")) +
  guides(shape = "none") +
  labs(x = "when parliament spoke",
       y = "events mentioned in the past by number of mentions",
       subtitle = "In the Months Leading Up to August 1867") +
  ggtitle("Events Mentioned by Name")
```



What do we make of this timeline? It suggests a swell of public meetings referenced in parliament in the months leading up to August 1867, which received more intense attention than either the Crimean War or the Paris Exhibition. Individual meetings such as those in Yorkshire or Clontarf received passing acknowledgment rather than deeper debate. Understanding more than this requires more cycles of research and reading.

Other research questions appear that the individual analyst, engaged in a process of research, would want to follow. Did references to public meetings change from January to August, while their mentions intensified? Were all of these public meetings about the vote? What was the meaning of

references to a "civil war" – the English civil war or recent events in America – and was this civil war invoked in reference to 1867? What about the references to the contemporary famine in Orissa (in India) or the (presumably historical) famine in Ireland?

One way to approach this line of inquiry would be by using the graph we have generated as a source of concrete questions about how members of parliament invoked current and historical events in the lead-up to the vote on the Second Reform Act. A skillful researcher would follow up on most of these questions, using their skills at investigating data to read individual sentences, speeches, and entire debates and to form these readings into a historical argument about the role of events in shaping political reform.

Finding Historical Events Using an Index

Next, let's return to the idea of finding events using what we have learned about joins. The authors of this book have created an index of event names which lists the date of famous events like the Magna Carta (1215).

As *The Dangerous Art of Text Mining* makes clear, the authors of this text understand that there is no universal definition of the most important events in world history. Any index has its merits and its omissions. It is in comparing the subtleties of different lists of events that the historian begins to probe the changing political and cultural nature of memory – for example, the fact that references to the Glorious Revolution began to disappear after the First Reform Act, gradually displaced by allusions to the Tudor past.

```
data("events")


eventslist <- events %>%
  distinct(event_name, scholar_assigned_date) %>%
  filter(!scholar_assigned_date > 1840) %>%
```

```
  mutate(event_name = tolower(event_name)) %>%
  select(event_name, scholar_assigned_date)
```

```
head(eventslist)
```

```
## # A tibble: 6 x 2
##   event_name          scholar_assigned_date
##   <chr>                               <dbl>
## 1 french revolution                    1789
## 2 magna carta                          1215
## 3 norman conquest                      1066
## 4 corn laws                            1815
## 5 battle of boyne                      1690
## 6 glorious revolution                  1688
```

Notice that the eventslist contains two columns: a list of events by name, and a list of the historical dates on which those dates occurred.

We will use the list of events to count events mentioned in parliamentary speeches, using an `inner_join()`. Like its cousin, `left_join()`, `inner_join()` works with two data sets to make a new, third data set. We use `inner_join()` when we want to concentrate on the information in the middle of a venn diagram – that is, shared areas of overlap between two data sets. In this case, we want to find the ngrams from annotated_1830 that are also listed as entities in the data set eventslist.

```
data("hansard_1830")
data("debate_metadata_1830")


hansard_1830 <- hansard_1830 %>%
```

```
  left_join(debate_metadata_1830)
```

```
## Joining with `by = join_by(sentence_id)`
```

```
bigrams_1830 <- hansard_1830 %>%
  unnest_tokens(ngram, text, token = "ngrams", n = 2)


events_1830 <- bigrams_1830 %>%
  inner_join(eventslist, by = c("ngram" = "event_name"))
```

```
head(events_1830)
```

```
##      sentence_id speechdate                                          debate
## 1  S2V0022P0_158 1830-02-04 ADDRESS ON THE LORDS COMMISSIONERS SPEECH. ]
## 2 S2V0022P0_1722 1830-02-18                                    POUTUGAL. ]
## 3 S2V0022P0_1729 1830-02-18                                    POUTUGAL. ]
## 4 S2V0022P0_1735 1830-02-18                                    POUTUGAL. ]
## 5 S2V0022P0_1833 1830-02-18                                    POUTUGAL. ]
## 6 S2V0022P0_1923 1830-02-18                                    POUTUGAL. ]
##                     ngram scholar_assigned_date
## 1         navigation laws                 1660
## 2 portuguese constitution                 1822
## 3 portuguese constitution                 1822
## 4 portuguese constitution                 1822
## 5 portuguese constitution                 1822
## 6 portuguese constitution                 1822
```

Notice that the resulting data set contains speaker and date columns from the parliamentary speeches, but the only bigrams listed are those that correspond to events from the eventslist data

set.

In the next blocks of code, we use our new list of events to generate a visualization. You should not expect to recognize every command in the following lines of code, but try to notice the ones that look familiar. Also note the use of comments, signaled by a hashtag ('#'), which tells the language R to stop looking for code. Comments are where the coder makes notes to the reader what a section of code is doing.

```
counted_events <- events_1830 %>%
  mutate(year = year(speechdate)) %>% # create a new column called "year" and assign it just t
  group_by(ngram, year, speechdate, scholar_assigned_date) %>%
  summarize(n = n()) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'ngram', 'year', 'speechdate'. You can
## override using the `.groups` argument.
```

```
head(counted_events)
```

```
## # A tibble: 6 x 5
##   ngram                  year speechdate scholar_assigned_date     n
##   <chr>                 <dbl> <IDate>                     <dbl> <int>
## 1 american constitution  1831 1831-10-05                   1776     1
## 2 american constitution  1831 1831-10-06                   1776     2
## 3 american constitution  1832 1832-04-13                   1776     2
## 4 belgian revolution     1831 1831-08-09                   1830     1
## 5 belgian revolution     1831 1831-08-18                   1830     1
## 6 belgian revolution     1832 1832-03-16                   1830    10
```

```
cleaned_events <- counted_events %>%
  group_by(ngram, scholar_assigned_date) %>%
  summarize(total = sum(n)) %>%
  ungroup() %>%
  arrange(scholar_assigned_date)
```

```
## `summarise()` has grouped output by 'ngram'. You can override using the
## `.groups` argument.
```

```
head(cleaned_events)
```

```
## # A tibble: 6 x 3
##   ngram              scholar_assigned_date total
##   <chr>                              <dbl> <int>
## 1 norman conquest                     1066     5
## 2 twelfth century                     1100     3
## 3 thirteenth century                  1200     7
## 4 fourteenth century                  1300     2
## 5 game laws                           1389   112
## 6 fifteenth century                   1400     7
```

```
indexed_events <- cleaned_events %>%
  mutate(index = scholar_assigned_date %/% ((max(scholar_assigned_date) - min(scholar_assigned
  # group into chronological periods of equal size and select one per  tranche
  group_by(index) %>%
  filter(total == max(total)) %>%
  ungroup()  %>%
  left_join(counted_events, by = c("ngram", "scholar_assigned_date"))
```

```
## Warning in left_join(., counted_events, by = c("ngram", "scholar_assigned_date")): Each row
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

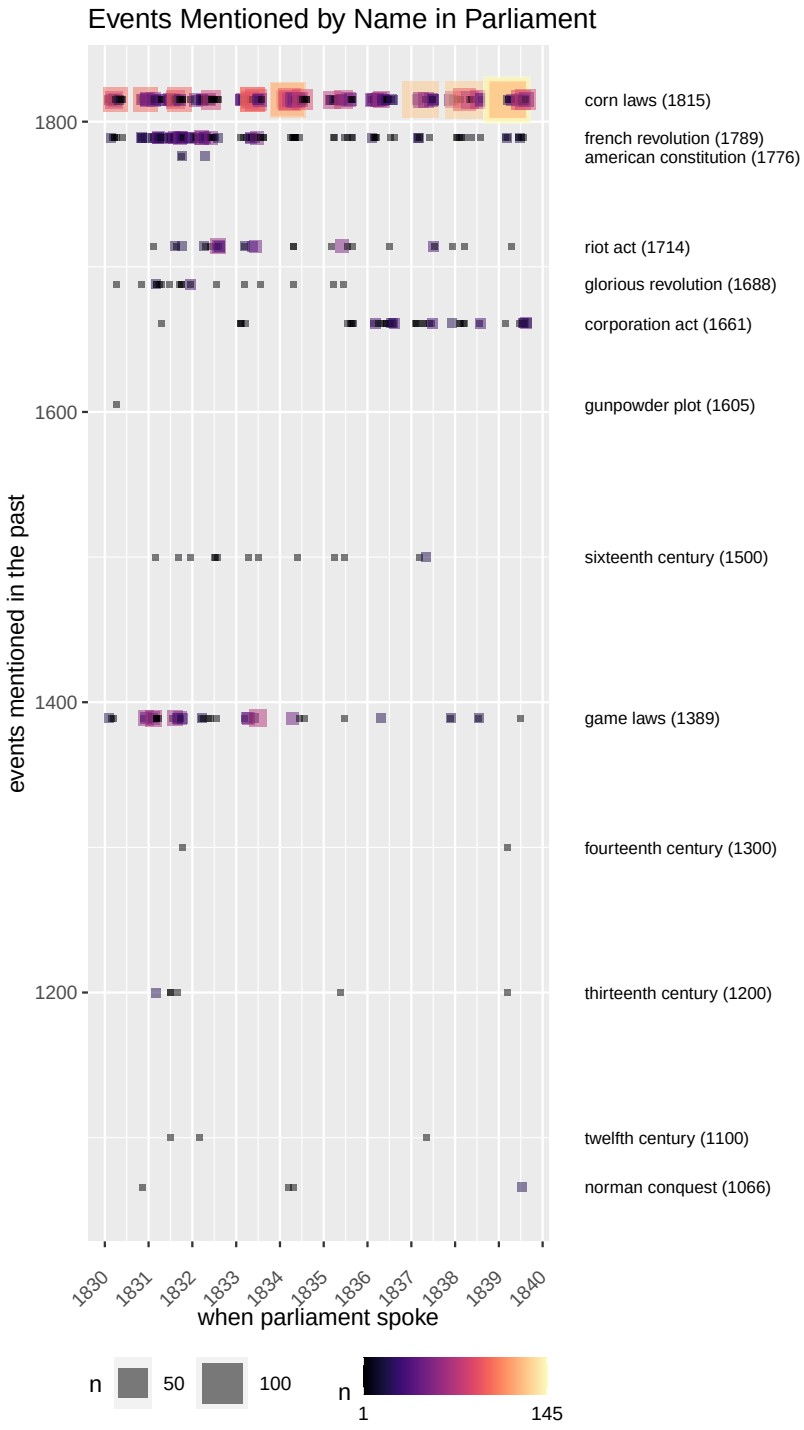
```
head(indexed_events)
```

```
## # A tibble: 6 x 7
##   ngram           scholar_assigned_date total index  year speechdate     n
##   <chr>                           <dbl> <int> <dbl> <dbl> <IDate>    <int>
## 1 norman conquest                  1066     5    41  1830 1830-11-15     1
## 2 norman conquest                  1066     5    41  1834 1834-03-14     1
## 3 norman conquest                  1066     5    41  1834 1834-04-25     1
## 4 norman conquest                  1066     5    41  1839 1839-07-12     2
## 5 twelfth century                  1100     3    43  1831 1831-07-06     1
## 6 twelfth century                  1100     3    43  1832 1832-02-27     1
```

```
left_range <- max(events_1830$speechdate) + 8
```

```
library(viridis)


ggplot(data = indexed_events,
       aes(x = speechdate,
           y = scholar_assigned_date,
           size = n,
           label = paste0('            ', ngram, ' (', scholar_assigned_date, ')'),
           color = n)) +
  scale_color_viridis(breaks = round,
                      trans = "log",
```

```
                    option = "A",
                    discrete = F,
                    direction = 1) +
scale_size_continuous(range = c(1, 10)) +
geom_point(alpha = .5,
           shape = 15) +
coord_cartesian(clip = 'off') +
scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
geom_text(data = indexed_events %>% group_by(ngram) %>% sample_n(1),
          show.legend = FALSE,
          aes(color = 1,
              x = left_range,
              y = scholar_assigned_date,
              hjust = 0,
              size = 8)) +
theme(legend.position = "bottom",
      plot.margin = unit(c(1,250,50,1), unit = "pt"),
      axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1),
      axis.title.x = element_text(vjust=-0.5)) +
guides(shape = "none") +
labs(x = "when parliament spoke", y = "events mentioned in the past") +
ggtitle("Events Mentioned by Name in Parliament")
```


Events Mentioned by Name in Parliament
events mentioned in the past
1800
1600
1400
1200
corn laws (1815)
french revolution (1789)
american constitution (1776)
riot act (1714)
glorious revolution (1688)
corporation act (1661)
gunpowder plot (1605)
sixteenth century (1500)
game laws (1389)
fourteenth century (1300)
thirteenth century (1200)
twelfth century (1100)
norman conquest (1066)
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
when parliament spoke
n
50
100
n
1
145

Note that the visualization here resembles the visualization of events we generated above, but in this case, we have some extra information: the actual date of the event mentioned, e.g. 1789 for the French Revolution. We have plotted this historical information on the y-axis. The resulting graph in fact shows *two* timelines rather than one – a timeline of parliamentary speech on the x-axis and a timeline of historical references on the y-axis.

How shall we interpret the graph? One way of proceeding from the representation to insight is to ask what the most surprising part of the visualization is to an area expert. As a British historian, I'm not surprised that events from the previous half century – like the French and American Revolutions – were named frequently in parliament. I'm not even that surprised that the Glorious Revolution was referenced a great deal in the early 1830s and less so after 1836. I'm certainly not surprised that the Riot Act was invoked on an annual basis. But perhaps the most surprising part is the vertical line of the deep past invoked around 1832, whhere the sixteenth century, thirteenth century, and twelfth century were invoked.

Why? I find myself asking. That verticality of temporal experience – the long perspective on the deep past – seems to be fairly distinct for 1830-32 of all the years in the decade. s Why do so many different, varied periods suddely become relevant all at once? Is this a general characteristic of moments of political reform – that the deep past is trawled for a multitidue of exmples, for and against?

We continue the explanation through reading.

```
hansard_1830 <- hansard_1830 %>%
  mutate(year = year(speechdate))


c15_mentions <- hansard_1830 %>%
  filter(year > 1831 & year > 1833) %>%
  filter(str_detect(text, "fifteenth century")) %>%
```

```
  select(text)
```

```
c15_mentions$text
```

```
## [1] "and learned Member would not defend the extravagant pretensions of the Pope in the fif
## [2] "From the conquest until nearly the middle of the fifteenth century, the exportation of
## [3] "All freeholders, to the extent of the fraction of a farthing, had, up to the middle of
## [4] "If calculations of this kind were to be those on which they were to proceed, they must
```

```
c14_mentions <- hansard_1830 %>%
  filter(year > 1831 & year > 1833) %>%
  filter(str_detect(text, "fourteenth century")) %>%
  select(text)
```

```
c14_mentions$text
```

```
## [1] "Scarcely a valuable discovery had been introduced; the spinning in some cases by hand,
```

```
c13_mentions <- hansard_1830 %>%
  filter(year > 1831 & year > 1833) %>%
  filter(str_detect(text, "thirteenth century")) %>%
  select(text)
```

```
c13_mentions$text
```

```
## [1] "Baronet who made such assertions, some nights ago, seemed not to have known, or to hav
## [2] "The account of this trade by Mr. Porter was as follows:- \"\"There is no doubt that br
```

```
c12_mentions <- hansard_1830 %>%
  filter(year > 1831 & year > 1833) %>%
  filter(str_detect(text, "twelfth century")) %>%
  select(text)
```

```
c12_mentions$text
```

```
## [1] "In the tenth century, when Canute came into the nation, a great passion for letters sp
```

What we see is a collection of historical references which serve two purposes: one is explaining why the institutions of the past differ from the needs of the present, and one is invoking the fact that parliamentary institutions have changed in the past to support the cause of reform in the present. The arguments work together, not against each other.

While we acknowledge that other historians have studied the discourse of parliamentary reform indetail, and even arrived at this same conclusion, what we find from a data-driven analysis of historical references to past events in parliament is the distinctiveness of 1832 as a moment of memory. Understanding how memory was used requires deep reading and a knowledge of the context of the debates over the vote; but understanding that historical memory was being leveraged in a particularly intensive way in 1832 is an insight that comes from treating text as data.

Exercises

1) In the code above, we searched for 5-word phrases that reference "the people." Use the code to find two-word phrases from 1830 and 1860 that reference "meeting." Write a paragraph, quoting at least 10 phrases from each decade. For each of the phrases mentioned more than five times, describe what prejudices, positive or negative, the phrase encapsulates. Use your evidence to answer the question: how did parliamentary references to meetings change between 1830 and 1860?

2) In the code above, we created two kinds of timelines, one for 1867 using a controlled vocabulary of words such as "meeting" and the other for the 1830s using the index of historical events, eventlist. Your job is to make two more timelines. Apply the controlled vocabulary to the 1830s, and search for the eventlist in the 1860s. Use the code above to graph the results. Keep

in mind that you will need to adjust your controlled vocabulary to what you find in the data, just as we do above.

3) Analyze the results of your timelines. Use iterative investigations of phrases, as modeled above, to conduct a distant reading of the parliamentary speeches. Using at least ten examples, compose a one-page essay investigating the question: how did parliamentary speakers use references to contemporary and historical events to make arguments for or against political change?

Chapter 3: Investigating Speakers and Change Over Time Using Grouped Data

In previous chapters, we investigated how breaking up strings of text into individual words or multi-word phrases can give us insight into a corpus. We did things like count the number of words within an entire data frame. But historians often want to profile not merely the collective use of language, but also the role of individual speakers and how their langauge changed over time. Or historians might want to count by other variables, like by the date or by the individual debate.

In this chapter, we will learn how to count by "grouping" the data, allowing us to take other fields, like speaker name, year, and debate title, into account. This chapter will use these groupings to gain insight into the disourse of speakers in Parliament. But in order to perform a critical analysis of speakers, we must first ask ourselves: why study the langauge of individual speakers in Parliament when 19th-century British Parliament was mostly a small group of elite white men and not people who reflected the British population?

One answer is that we can model the language of parliament to better understand the workings of institutions, rather than "great men" of history. We can better understand the dynamics of power that imposed itself on 19th-century British society. The language of Parliamentarians was not merely communication, but also an instrument of power through which these individuals imposed an imagined order upon Britian–an order that reflected the biases, fears, and desires of elite white men–many of whose fathers served in Parliament, and who themselves inhereted their place in Parliament through familial status.

Thus, when we analyze speakers we better understand which people were rewarded with power, and how this power was expressed through langauge. We get a glimps of how Parliament existed as a space where the power relations of a small, elite group were actualized as the legitimate order of British society–a power that extended outwards to the societies women and its poor.

Counting by Group to Find the Wordiest Speakers

Our first excersize is finding the most "wordy" Parliamentarians from 1830–that is, Parliamentarians who spoke the most words. Often, the Parliamentarians who spoke the most did so becuase fellow Parliamentarians trusted them to express the wants of their own social group.

To analyze speakers in Hansard, we will first need to import a new category of data from hansardr: "speaker_metadata."

```
library(hansardr)


data("speaker_metadata_1830")
```

```
head(speaker_metadata_1830)
```

```
##       sentence_id    speaker suggested_speaker ambiguous fuzzy_matched ignored
## 1 S3V0010P0_11508 Mr. Croker  john_croker_1539         0             0       0
## 2 S3V0010P0_11509 Mr. Croker  john_croker_1539         0             0       0
## 3 S3V0010P0_11510 Mr. Croker  john_croker_1539         0             0       0
## 4 S3V0010P0_11511 Mr. Croker  john_croker_1539         0             0       0
## 5 S3V0010P0_11512 Mr. Croker  john_croker_1539         0             0       0
## 6 S3V0010P0_11513 Mr. Croker  john_croker_1539         0             0       0
```

For now, we will just go over three relevant fields from `speaker_metadata`.

Each sentence from the Hansard corpus is assigned a unique ID, as represented by the "sentence_id"

field. The speaker who stated a sentence is assigned the same ID. This allows us to join the data from speaker_metadata to other data from hansardr, such as the debate text.

The other two important fields we will address are the "speaker" and the "suggested_speaker" fields.

The "speaker" field contains the speaker name as it was originally transcribed within the Hansard corpus. The resulting field contains inconsistencies and a lack of standardization that can make analysis of speakers difficult. A single speaker may have been recorded by different permutations of his first, middle and last name(s) (for example, "William Gladstone" may be transcribed as "William E. Gladstone", "W. Gladstone", or "Mr. Gladstone," to name just a few). In other cases, a speaker may have been called by a rotating office title (like "Prime Minister"). It is also the case that different speakers are recorded by the same name. For instance, two people named Sir Robert Peel served in Parliament during the 1820s and both are evoked by the same spelling of the name. To make analysis of speaker names more challenging yet, during the digitization processes, optical character recognition (OCR) errors were introduced into the speaker names. Common OCR error include interpreting a lower case "L" as an "i," or the lower case letter "a" as an "o."

To address the hurdles around analysis of speakers, we added a "suggested_speaker" field. This field contains our suggestion for the true identity of the speaker. The unique speaker within the corpus are assigned standardized name and number combinations. While multiple speakers during the same period shared the same name (like how "Mr. W. Gladstone" could refer to William Ewart Gladstone or his son William Henry Gladstone) only William Gladstone is assigned number 3104.

In the following code we will make our data set smaller and easier to work with by just selecting the "speaker," "suggested_speaker," and "text" fields before tokenizing the data into individual words.

```
library(tidytext)
library(tidyverse)
library(lubridate)
```

```
data("hansard_1830")


words_1830 <- hansard_1830 %>%
  left_join(speaker_metadata_1830) %>%
  select(speaker, suggested_speaker, text) %>%
  unnest_tokens(word, text)
```

```
head(words_1830)
```

```
##                 speaker suggested_speaker   word
## 1 The Duke of Buccleugh walter_scott_6566   rose
## 2 The Duke of Buccleugh walter_scott_6566     my
## 3 The Duke of Buccleugh walter_scott_6566  lords
## 4 The Duke of Buccleugh walter_scott_6566     in
## 5 The Duke of Buccleugh walter_scott_6566 rising
## 6 The Duke of Buccleugh walter_scott_6566     to
```

With a dataframe containing fields for "speaker", "suggested_speaker", and "word," we are now in a position to perform a new kind of analysis to see the number of words each speaker contributed to Parliament using two new functions: `group_by()` and `summarize()`. We will use `group_by()` to organize the data by group before using summarize for our count. By using `group_by()` we can count the number of times a word was stated by a speaker, or within a year, and so forth. `group_by()` can be passed multiple arguments that refer to fields in a data set. To group by speaker will pass "speaker" to the `group_by()` function. Another useful function along with `group_by()` is `summarize()`. The `summarize()` function can be used with any statistical transformation, for instance `mean()` or `max()`. Here, we will use `summarize()` with a function to count, which is `n()`.

In the following code the argument `words_spoken = n()` tells `summarize()` to create a new column

with the number of words that reflect each unique speaker

```
words_per_speaker_1830 <- words_1830 %>%
  group_by(speaker) %>%
  summarize(words_spoken = n()) %>%
  arrange(desc(words_spoken))
```

```
## # A tibble: 6 x 2
##   speaker                          words_spoken
##   <chr>                                   <int>
## 1 Mr. Hume                               856835
## 2 Mr. O'Connell                          823600
## 3 Sir Robert Peel                        822825
## 4 Lord John Russell                      782439
## 5 The Chancellor of the Exchequer        620830
## 6 Lord Brougham                          596096
```

Most students of British history will be familiar with the names of the speakers on this list, which include two prime ministers as well as one noted Irish orator and nationalist, Daniel O'Connell. The careful analyst will be a little concerned about the speaker as "The Chancellor of the Exchequer," however, which refers not to an individual but a title that was handed off to multiple individuals over the century.

The Chancellor of the Exchequer is the head financial officer of the United Kingdom. He oversees the work of the Treasury. In many cases, the Chancellor of the Exchequer was a preliminary step to becoming Prime Minister. We would expect most of the Chancellor of the Exchequer's speeches in parliament to reflect a predominant concern with taxation and spending. However, it is also possible that different individuals who held this post prioritized different causes. We can test our hypothesis on the 1830s by comparing the top words spoken by different individuals who occupied this post.

In the code below we create a custom stop words list to eliminate the most frequent words said by all members of parliament. We have already addressed the importance of removing stop words as a means to see words that are more meaningful for an analysis. In previous chapters we used an existing stop words list from TidyText. The hansardr library also provides its own stop words list. It may be the case, however, that an analyst wants to curate their own stop words list that caters to their specific data set or research question. In the following code we create a stop words list by assigning a list of words to a tibble (a tidyverse-style data frame).

```
custom_stop_words = tibble(word = c("hon", "speaker", "house", "question", "lord", "bill", "co
"proposition", "persons", "principles","service", "found", "propositions", "office", "matter",
"paid", "increase", "moved", "means", "considerable", "supply", "intention", "debt", "received
"sale", "times", "amounted", "deficiency", "laws", "til", "agreed", "alluded", "mentioned", "m
"sense", "arrangement", "address", "sufficient", "local", "clergy", "relief", "necessity", "ap
```

We can now filter and clean our data and explore the langauge of speakers called Chancellor of the Exchequer. In the following code we first remove instances of John Spencer, who spoke so few words as Chancellor of the Exchequer that he contributes little to our project. Then we eliminate numbers, stop words, and custom stop words before counting each speaker's words with `group_by()` and `summarize()`.

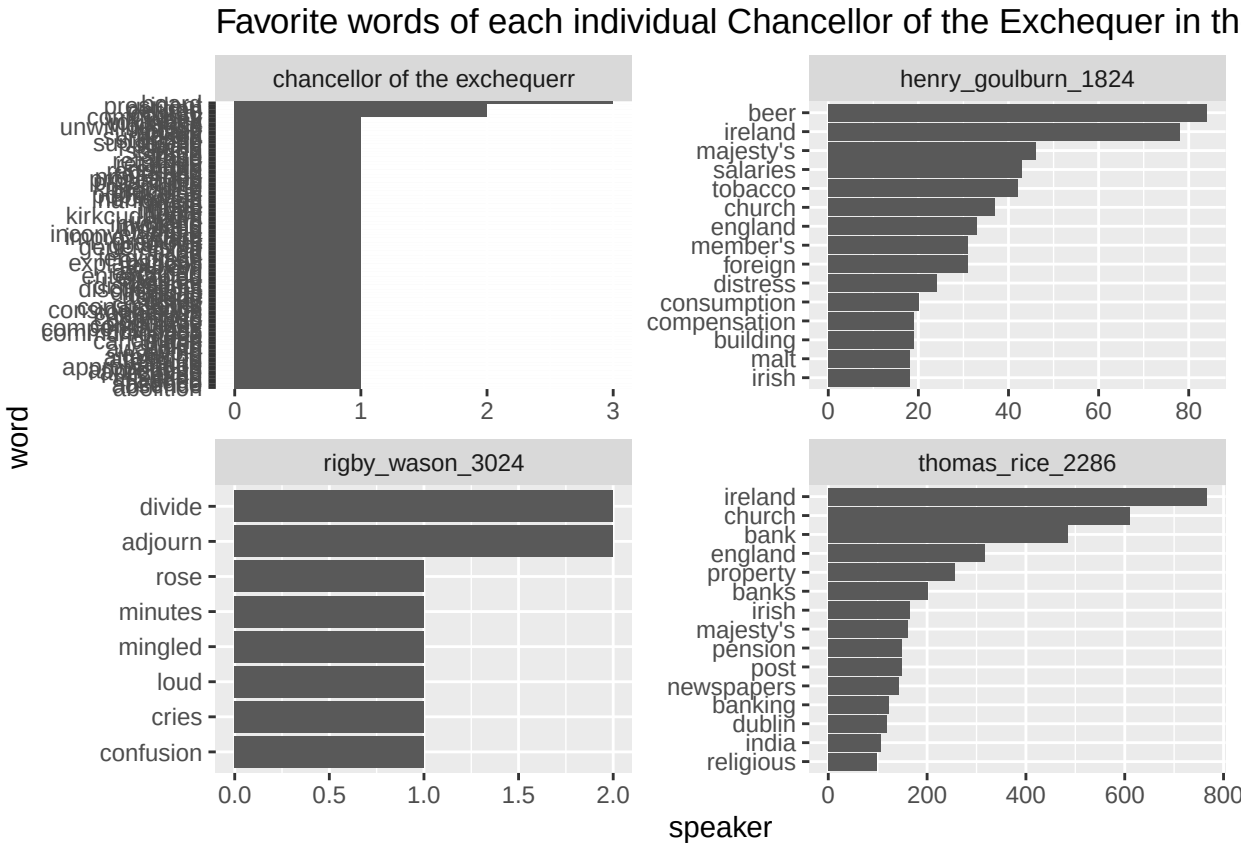
```
chancellor_of_the_exchequeur <- words_1830 %>%
  filter(str_detect(speaker, "Exchequer"),
         str_detect(word, "[a-z]"),
         suggested_speaker != "john_spencer_1234",
         suggested_speaker != "") %>%
   anti_join(stop_words) %>%
   anti_join(custom_stop_words) %>%
   group_by(suggested_speaker, word) %>%
```

```
  summarize(n = n()) %>%
  top_n(15) %>%
  ungroup()
```

```
head(chancellor_of_the_exchequeur)
```

```
## # A tibble: 6 x 3
##   suggested_speaker            word          n
##   <chr>                        <chr>     <int>
## 1 chancellor of the exchequerr abolition     1
## 2 chancellor of the exchequerr absence       1
## 3 chancellor of the exchequerr accede        1
## 4 chancellor of the exchequerr added         1
## 5 chancellor of the exchequerr affecting     1
## 6 chancellor of the exchequerr alien         1
```

```
chancellor_of_the_exchequeur <- chancellor_of_the_exchequeur %>%
  mutate(word = reorder_within(word, n, suggested_speaker))


ggplot(data = chancellor_of_the_exchequeur,
       aes(x = word, y = n)) +
  geom_col() +
  scale_x_reordered() +
  coord_flip() +
  facet_wrap(~suggested_speaker, scales = "free") +
  labs(x = "word", y = "speaker") +
  ggtitle("Favorite words of each individual Chancellor of the Exchequer in the 1830s")
```



The results show a division of interests. Goulburn is interested in talking about taxation on beer, tobacco, malt, and consumption; Peel is interested in addressing questions about how taxes relate to the church and its tithes, as well as political issues about how dissenters, Protestants and Catholics. Peel also thinks about commodities, but his commodities are different than Goulburn's; they include malt, land, and barley. Rice takes the issues in a different direction, reflecting on Dublin, India, banks, newspapers, and pensions. He shares with Peel a concern about property or land taxes.

How new is this information to readers of British history? We have long known that issues of what was taxed and how are a basic matter for politics of class, empire and nation; everyone wants someone else to pay for the government.

The research modeled in this textbook has a pedagogical function, and for that reason, we often

present in these chapters examples that could be improved, and the above chart is one of those. Analysts should note several problems that might complicate our analysis and that call for further work. One of these is that a frequent word does not necessarily equate to importance: the most frequently-mentioned terms are typically the terms most subject to debate. Robert Peel, for instance, does not name Ireland's church tithes because he was interested in protecting the church in Ireland; to the contrary, he wished to abolish the system of tithes. Nor does the word list give us material for interpreting the rest of Peel's politics; the word list does nothing to tell the uninformed analyst about the movement for the abolition of taxes on grain, which forms a background to interpreting Peel's mentions of barley. The word list tells us little about the presence in 1830s Ireland of two churches, Roman Catholic and the Church of England. Without further reading, we might never known that Goulburn was famous for arguing for reductions on taxation. The analyst who comes to this data without a background in British history would do well to read in secondary sources before attempting to interpret the graph.

After some basic background research, however, the word lists can help produce insight, cluing the reader into differences such as Goulburn's relative willingness to talk about "distress," Peel's interest in "education," or Rice's interest in the post office. With word lists of this sort, we retrieve important clues about the salient differences contributed to political debates by individual ministers. A researcher interested in the careers of Chancellors of the Exchequer might start here before reading more deeply.

Another issue is the custom stop words list. For some researchers, the list in `custom_stop_words` may unnecessarily eliminate words whose usage they might want to track. Some researchers will want to identify words for rhetorical statements such as "hoped" or "feeling"; others may be intrigued about how different speakers refer to "information" or "knowledge." In general, I have opted for a deep list which includes every general allusion to governance, institutions, discovery, and communication. The beauty of custom lists is that each researcher can easily adjust the words to match their interests Analysts should understand how carefully they must tailor a custom stop

words list to their project, and how much of a difference additions and subtractions make. Using a custom stop word list is very much an artisinal skill of research; it involves much the same skills of judgment, awareness, curiosity, and sensitivity to background issues from theory and secondary sources as does careful reading of primary sources.

One complication of lists is that to produce functional results, a custom stop words list must often be long; the hundreds of performative, rhetorical, and governmental words listed below is not a perfect list, and it would not work for every exercise. In later chapters, we will investigate approaches such as differential measurement which allow the reader to skip over custom stop words lists to obtain information about what distinguishes one speaker from another or one year from the next. No process is perfect, however, and analysts must often resort to some kind of custom filter to get closer to information that is useful for their project. There is no silver bullet for creating an absolutely accurate tool where text-mining research always produces useful information; rather, iterative inquiry, paired with background reading and in-depth reading of primary sources, is the basis upon which all insight is ultimately made.

Another issue has to do with the quality fo the data in the "speaker" field, which corresponds exactly to what is on the page in the nineteenth-century printed records of parliament. There is no guarantee that this method will produce a perfect analysis; we're going by who was lableled Chancellor of the Exchequer in Hansard, and there are no guarantees that the printed books label all speakers by their title. This is because the "speaker" field in the hansardr data set lists the names of the speakers as they are described in the published version of the debates. Wherever possible, we have annotated this information about speaker with static id's for most individual speakers, using the Parliamentary Identifiers used in parliament to this day. The field "suggested_speaker" uses these specific id numbers and gives a more accurate picture of the top speakers.

With more research, we could use the disambiguated speaker and date fields to revise the output above, creating a more accurate analysis that gives all the speeches by the individuals in question

while they held the post of Chancellor of the Exchequer. Most of the time, we will want to use the suggested_speaker field to find and track speakers.

Notice that the lines of code below differ from the lines of code above only in that the argument of the function `group_by()` has been changed from "speaker" to "suggested_speaker:"

```
words_per_speaker_1830 <- words_1830 %>%
  filter(suggested_speaker != "") %>%
  group_by(suggested_speaker) %>%
  summarize(words_spoken = n()) %>%
  arrange(desc(words_spoken))
```

```
## # A tibble: 6 x 2
##   suggested_speaker    words_spoken
##   <chr>                       <int>
## 1 robert_peel_1664          1224576
## 2 henry_brougham_1679       1180672
## 3 joseph_hume_1712           875951
## 4 daniel_oconnell_2552       808577
## 5 john_russell_1885          802418
## 6 thomas_rice_2286           647551
```

In this view, we see the first and last names of individual speakers as well as their ID number from parliament (that is, Robert Peel is id 1664; 1664 is not, in this case, a date).

Working With Dates

Knowing who spoke the most in parliament is useful. But very often insight comes not from examining how counts change over time. What if we want to know not merely who spoke the most, but who was the top speaker for each year in parliament? Our `hansard_1830` data frame lists a

date in the "speechdate" column, formatted as a four-digit year, two-digit month, and two digit day. To easily track speeches by year, we can add a new "field" – that is, a column – listing just the year of each speech.

Very frequently, when working with information about dates, we need to extract the month, day, or year from a data set. The lubridate package makes it easy to extract this information. We will use the `year()` function from lubridate with `mutate()` to create a new column that has just the extracted year from the `speechdate` field.

```
data("debate_metadata_1830")


words_1830 <- hansard_1830 %>%
  left_join(speaker_metadata_1830) %>%
  select(speaker, suggested_speaker, text) %>%
  unnest_tokens(word, text)

hansard_1830 <- hansard_1830 %>%
  left_join(debate_metadata_1830) %>%
  mutate(year = year(speechdate))


words_1830 <- hansard_1830 %>%
  left_join(speaker_metadata_1830) %>%
  select(speaker, suggested_speaker, text, year) %>%
  unnest_tokens(word, text)
```

```
##                 speaker suggested_speaker year   word
## 1 The Duke of Buccleugh walter_scott_6566 1830   rose
## 2 The Duke of Buccleugh walter_scott_6566 1830     my
```

```
## 3 The Duke of Buccleugh walter_scott_6566 1830  lords
## 4 The Duke of Buccleugh walter_scott_6566 1830     in
## 5 The Duke of Buccleugh walter_scott_6566 1830 rising
## 6 The Duke of Buccleugh walter_scott_6566 1830     to
```

Notice that the resulting dataset has a new field, "year."

Now that we have a field for the year of each speech, we can return to "grouping" data to execute a faceted count where we count words by speaker and year. When we used the command `group_by()` above, we used it with one argument. But `group_by()` can take multiple arguments, allowing the analyst to perform grouped counts that involve multiple dimensions of the dataset. Next, we will use `group_by()` on two data fields at the same time – the "speaker" and "year" columns of as the arguments of `group_by()`, as in `group_by(speaker, year)`.

```
words_per_speaker_1830 <- words_1830 %>%
  filter(suggested_speaker != "") %>%
  group_by(speaker, year) %>%
  summarize(words_spoken = n()) %>%
  arrange(desc(words_spoken))
```

```
## # A tibble: 6 x 3
## # Groups:   speaker [4]
##   speaker              year words_spoken
##   <chr>               <dbl>        <int>
## 1 Lord Brougham        1839       200436
## 2 Lord Althorp         1831       180573
## 3 Lord Brougham        1838       175070
## 4 The Lord Chancellor  1831       161729
## 5 Mr. O'Connell        1834       158617
```

```
## 6 Lord Althorp          1833       154948
```

Notice that the result of grouping and counting is a data set with three columns: "speaker," "year," and "words spoken." The last column is the count for each speaker per year – a field that makes it possible to analyze change over time.

When we use the `group_by()` command, the fields we use for grouping are preserved in the output. Two columns are the names of the facets we used for grouping – "speaker" and "year" – while the last column, "words_spoken," is the count of how many words are recorded for each speaker per year.

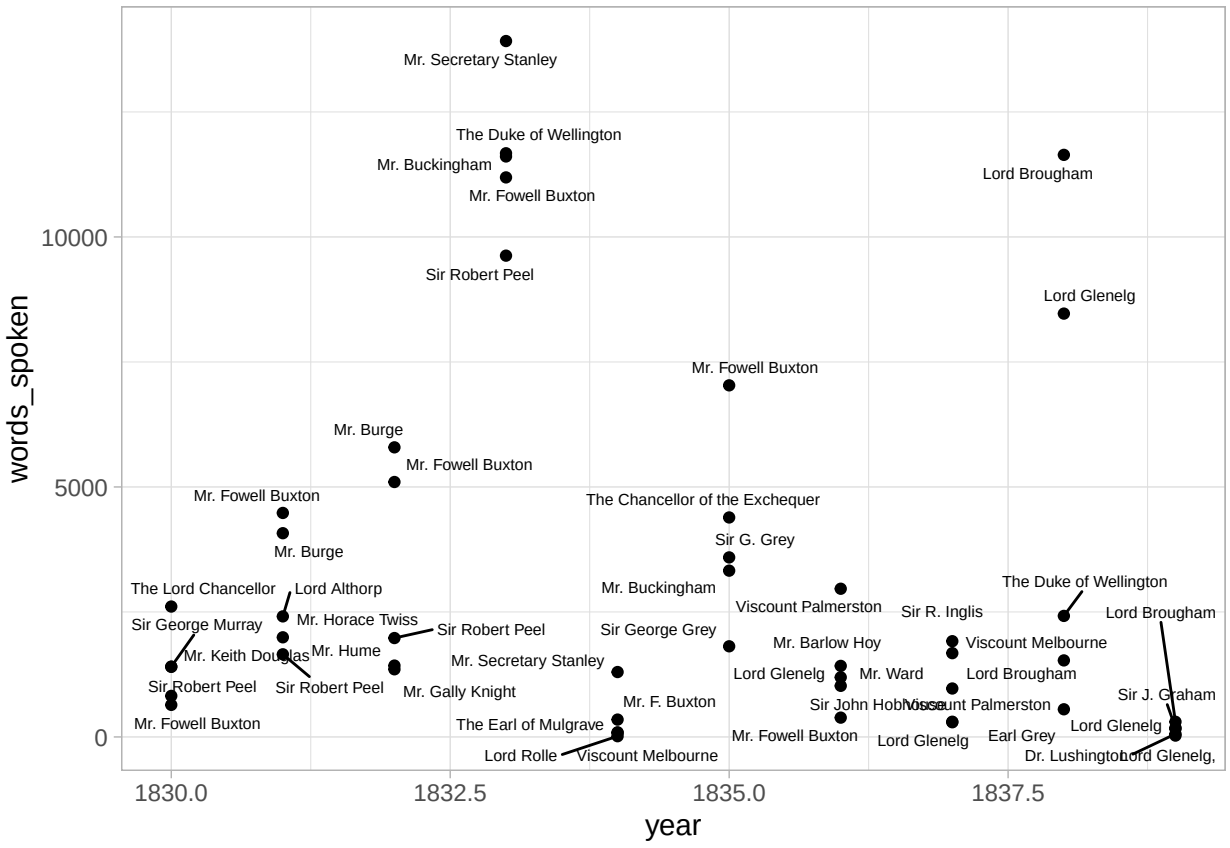
Top Speakers in Debates About Slavery

What if we want to find just those speakers who spoke the most in debates about slavery? We can use `str_detect()` to look for the word "slavery" in the titles of debate. Then we could the top speakers as before.

If you are new to code, take a moment to notice how changing only *one line* in the code below produces a totally different angle of analysis. The code below is otherwise exactly the same as the code in the sections above. One of the great pleasures of coding is the facility with which a few skills can quickly become powerful in the hands of an analyst who has many questions.

```
library(ggrepel)


slavery_1830 <- hansard_1830 %>%
  filter(str_detect(tolower(debate), "slavery")) %>%
  left_join(debate_metadata_1830) %>%
  mutate(year = year(speechdate))


slavery_words_1830 <- slavery_1830 %>%
```

```
  left_join(speaker_metadata_1830) %>%
  select(speaker, suggested_speaker, text, year) %>%
  unnest_tokens(word, text)

words_per_speaker_slavery_1830 <- slavery_words_1830 %>%
  filter(suggested_speaker != "") %>%
  group_by(speaker, year) %>%
  summarize(words_spoken = n()) %>%
  arrange(desc(words_spoken))

top_slavery_speakers <- words_per_speaker_slavery_1830 %>%
  group_by(year) %>%
  arrange(desc(words_spoken)) %>%
  slice(1:5)

ggplot(top_slavery_speakers,
       aes(x = year,
           y = words_spoken,
           label = speaker)) +
  geom_text_repel(size = 2) +
  geom_point() +
  theme_light()
```



You may notice that we have produced a different sort of visualization than the bar charts of the previous chapter. The ggplot library of graphics is a highly flexible language for producing many visualizations from the same data. Rather than the command for a bar chart – `geom_col()` – we have used a command for a dot plot – `geom_point()`. With this plot we also introduce another library, "ggrepel," which gives us tools for enhancing the readability of our plot by making sure that labels do not overlap.

What does this chart of the top speakers about slavery tell us? For one thing, it dramatizes the fact of the debates around 1833 – when Britain voted to abolish the slave trade in British empire. This event was once lauded as proof of Britain's liberalizing mission around the globe, but more recently, historians of British empire have emphasized that the theoretical abolition of slavery in

British empire did not result in the abolition of slavery in mahy parts of India and Africa under British control, nor did it minimize the punitive system of indentured servitude, by which many subjects of British empire were sent into conditions of forced labor abroad. Were we interested in considering the contribution of the major speakers to this debate, we would have their names: Mr. Secretary Stanley, the Duke of Wellington, Mr. Buckingham, Mr. Fowell Burton, and the Earl of Ripon. We also see the evidence of a second wave of debates, when issues of indentured servitude came to parliament's attention in 1838. Then, Lord Brougham and Lord Gleneig were the main contenders. In the next section, we will explore the contributions of these actors.

Using Our Skills to Explore the Historical Debates About the Abolition of Slavery

Understanding how to group data by speaker and year gives us the possibility of closely studying the contribution of different speakers. For instance, suppose that we wanted to read all of the debates relating to Slavery Abolition Act of 1833, which outlawed slavery in many British colonies, including the Caribbean and South Africa, although not across much of India, where debt slavery would persist for decades, and where the caste system would exclude many people from participation in the economy through the twenty-first century.

There are some new commands below that are used for demonstration purposes to show how one might navigate our newly joined dataset. Don't feel worried if you can't read every line. You will be learning how to aggregate data by speaker very soon.

```
## # A tibble: 20 x 2
##    speaker                    n
##    <chr>                  <int>
##  1 Mr. Buckingham         11610
##  2 Mr. Secretary Stanley  11368
##  3 The Duke of Wellington 10691
##  4 Sir Robert Peel         9626
```

```
##  5 The Earl of Ripon       9523
##  6 Mr. Fowell Buxton       9150
##  7 Mr. Godson              8734
##  8 Mr. O'Connell           6946
##  9 Mr. Hume                5843
## 10 The Lord Chancellor     5795
## 11 Dr. Lushington          5415
## 12 Earl Grey               4805
## 13 Lord Suffield           4467
## 14 Mr. Stanley             4319
## 15 Lord Ellenborough       4002
## 16 Lord Sandon             3846
## 17 Lord Althorp            3833
## 18 Mr. Macaulay            3252
## 19 Lord Howick             2995
## 20 Mr. Baring              2971
```

Counting how many words were spoken in the 6,000 some sentences about the abolition of British slavery is our first reminder of the bias of this data set: the speakers are white men, many of them of a class invested in slavery. This collection of words is not suitable for investigating narratives that depict lived experiences of dehumanization, recorded from the words of enslaved people.

Nevertheless, distant reading allows us to investigate the dynamics of politics in the British parliamentary system that made the official abolition of slavery possible. We can take a snapshot at the speeches of the most active speakers – Buckingham, Stanley, Wellington, and Peel – by comparing their most-invoked words in these debates.

In the following code, we provide a list of speakers and use the `%in%` operator to filter for any of

these speakers.

```
pattern1 = c("Mr. Buckingham", "Mr. Stanley", "The Duke of Wellington", "Sir Robert Peel")


abolition_speakers <- slavery_debates %>%
  left_join(speaker_metadata_1830) %>%
  filter(speaker %in% pattern1)
```

```
head(abolition_speakers)
```

```
##     sentence_id
## 1 S3V0017P0_725
## 2 S3V0017P0_726
## 3 S3V0017P0_727
## 4 S3V0017P0_728
## 5 S3V0017P0_729
## 6 S3V0017P0_730
##
## 1                                          presented a Petition from Magistrates, Bankers, M
## 2
## 3 The latter was signed by 2, 468 persons, and was well entitled to attention, no less on a
## 4
## 5
## 6                                                                    The petitioners referre
##   speechdate                                debate year                speaker
## 1 1833-05-02 ABOLITION OF SLAVERY. -PETITIONS. ] 1833 The Duke of Wellington
## 2 1833-05-02 ABOLITION OF SLAVERY. -PETITIONS. ] 1833 The Duke of Wellington
## 3 1833-05-02 ABOLITION OF SLAVERY. -PETITIONS. ] 1833 The Duke of Wellington
```

```
## 4 1833-05-02 ABOLITION OF SLAVERY. -PETITIONS. ] 1833 The Duke of Wellington
## 5 1833-05-02 ABOLITION OF SLAVERY. -PETITIONS. ] 1833 The Duke of Wellington
## 6 1833-05-02 ABOLITION OF SLAVERY. -PETITIONS. ] 1833 The Duke of Wellington
##       suggested_speaker ambiguous fuzzy_matched ignored
## 1 arthur_wellesley_1299         0             0       0
## 2 arthur_wellesley_1299         0             0       0
## 3 arthur_wellesley_1299         0             0       0
## 4 arthur_wellesley_1299         0             0       0
## 5 arthur_wellesley_1299         0             0       0
## 6 arthur_wellesley_1299         0             0       0
```

```
abolition_speakers <- abolition_speakers %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  group_by(speaker, word) %>%
  summarize(n = n())
```

```
head(abolition_speakers)
```

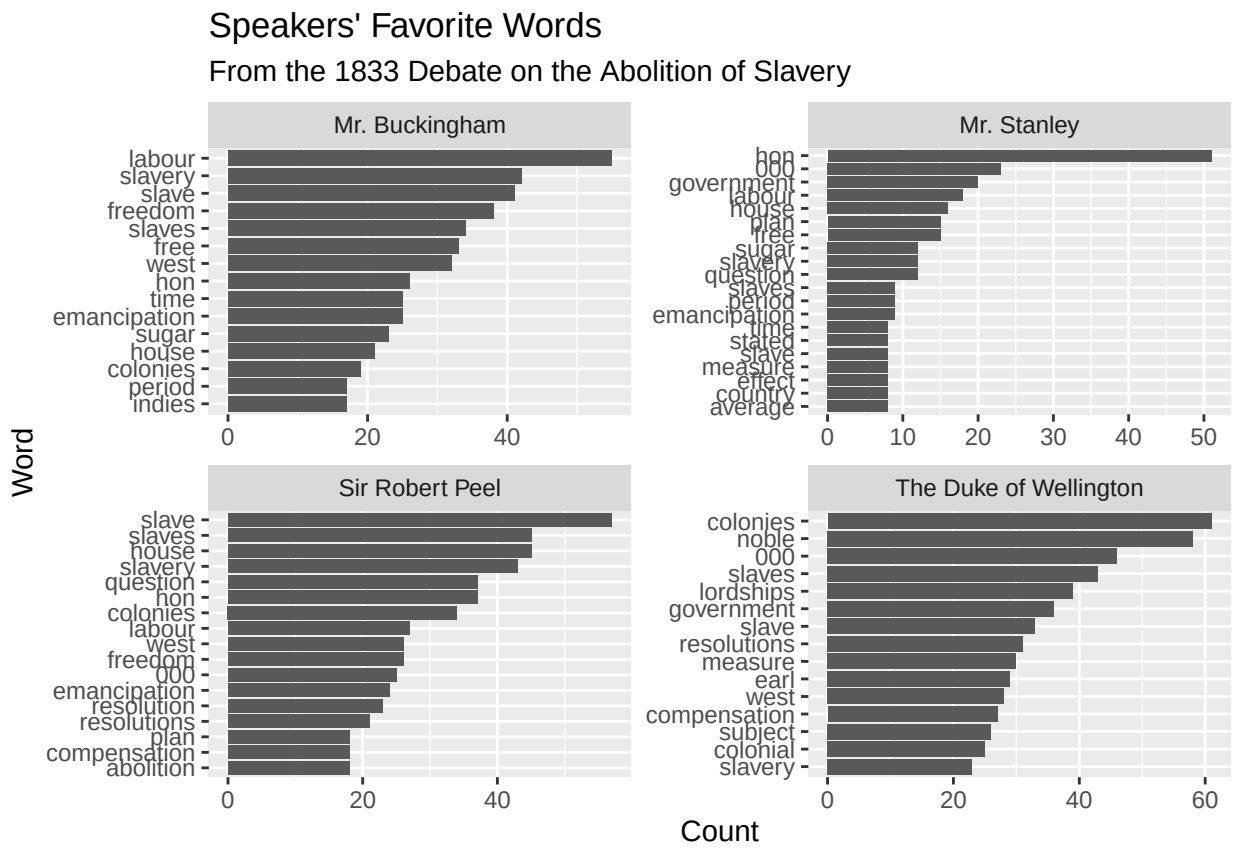
```
## # A tibble: 6 x 3
## # Groups:   speaker [1]
##   speaker        word      n
##   <chr>          <chr> <int>
## 1 Mr. Buckingham 000      15
## 2 Mr. Buckingham 125       1
## 3 Mr. Buckingham 14        1
## 4 Mr. Buckingham 1793      1
## 5 Mr. Buckingham 18        1
```

```
## 6 Mr. Buckingham 2         1
```

For our visualization, we can use just the top 15 owrds

```
abolition_speakers <- abolition_speakers %>%
  arrange(desc(n)) %>%
  top_n(15) %>%
  ungroup() %>%
  mutate(word = reorder_within(word, n, speaker))
```

Now visualize:



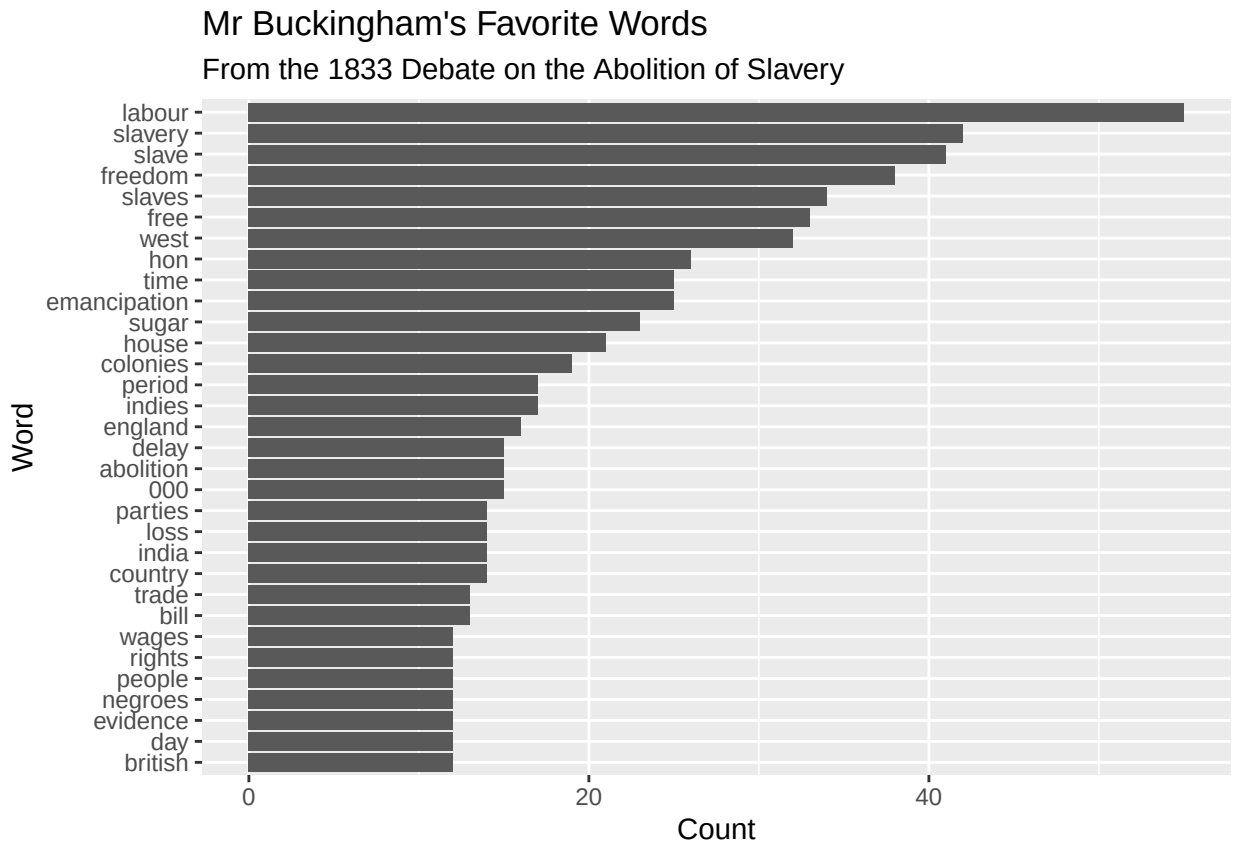
Unsurprisingly, many of the top speakers are less interested in the issue of slavery than in problems of governance. Even in a distant reading, we can detect that Robert Peel and the Duke of Wellington

were more interested in problems of "compensation" for the owners who benefited from the system of enslavement than in the details of life in the West Indies – an issue of political expediency, but also a reflection of the ethical orientation of parliament in the 1830s.

The one exception to this trend, in the four speakers examined, appears to be James Silk Buckingham, the reformer who toured sites of slavery round the world, collecting anecdotes from the slaves themselves about their experiences of cruelty.

We can look into Mr. Buckingham's language in greater detail.

```
mr_buckingham_slavery_debates <- slavery_debates %>%
  left_join(speaker_metadata_1830) %>%
  filter(speaker == "Mr. Buckingham") %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  group_by(speaker, word) %>%
  summarize(n = n()) %>%
  arrange(desc(n)) %>%
  top_n(30)


ggplot(data = mr_buckingham_slavery_debates,
       aes(x = reorder(word, n), y = n)) +
  geom_col() +
  coord_flip() +
  labs(title = "Mr Buckingham\'s Favorite Words",
       subtitle = "From the 1833 Debate on the Abolition of Slavery",
       x = "Word",
       y = "Count")
```



It is interesting that "labour" and "freedom" are used alongside "slavery" and "slave" as Buckingham"s favorite words. Also pronounced are his invocation of "rights," "wages," and "people," suggesting that Buckingham was interested in drawing equivalencies between the rights of working men in Britain and the rights of enslaved people int he West Indies.

We should not be too quick to leap to conclusions about how Buckingham used these words without reading directly from his speeches.

```
buckingham_sentences <- slavery_debates %>%
  left_join(speaker_metadata_1830) %>%
  filter(speaker == "Mr. Buckingham") %>%
  filter(str_detect(text, "labour"))
```

```
buckingham_sentences$text[1:10]
```

```
##  [1] "But, for the present, he would content himself with saying, that the two principal fea
##  [2] "This demand, however, was opposed by the Ministers, as well as by the West Indians, or
##  [3] "As to the indolence of the slaves, and their incapacity or unwillingness to labour for
##  [4] "All men disliked to labour more than was necessary to obtain for them the enjoyments o
##  [5] "The other work was one entitled \"\"Wages, of the Whip, \"\" drawn up by Mr. Conder, a
##  [6] "There was not the slightest ground, therefore, for the assumption, that, if liberated
##  [7] "Members be ignorant of the fact, that in our own immense empire of the East Indies, an
##  [8] "If, instead of the unjust preference which had hitherto been given to the produce of s
##  [9] "If loss should actually accrue for the first few years, from the change from a system
## [10] "member for Middlesex, that a much larger body of evidence than he seemed to contempla
```

We show here only the first ten sentences from Buckingham"s speeches on the Abolition of Slavery Bill. We can see from these sentences the many ways in which he invokes "labour." Not all of them are representations about his ideas; in the first several sentences he ventriloquizes the sentiments of slave-owners who believe that their production will grind to a halt unless they employ extreme cruelty. We see Buckingham arguing against that position and for the validity of and productivity of free labor.

If we were working on a paper analyzing these speeches, we would likely want to read Buckingham's speeches in their entirety. We can do this by loading "file_metadata" to get information on which speech a sentence belongs.

```
data("file_metadata_1830")
```

```
head(file_metadata_1830)
```

```
##   sentence_id speech_id debate_id src_file_id      src_image src_column
## 1 S2V0022P0_0     51883      6335   S2V0022P0 S2V0022P0I0030          4
```

```
## 2 S2V0022P0_1     51883      6335   S2V0022P0 S2V0022P0I0030          4
## 3 S2V0022P0_2     51883      6335   S2V0022P0 S2V0022P0I0030          4
## 4 S2V0022P0_3     51883      6335   S2V0022P0 S2V0022P0I0030          4
## 5 S2V0022P0_4     51883      6335   S2V0022P0 S2V0022P0I0030          4
## 6 S2V0022P0_5     51883      6335   S2V0022P0 S2V0022P0I0030          4
```

`file_metadata_1830` is a data frame with fields that either reference the original Hansard transcripts (such as `src_column`, for the column of the sentence) or were added to provide insight into the content and structure of the debates (such as `debate_id`, which tells us which sentences belonging to an entire debate).

It can be joined with `buckingham_sentences` to give us a way to easy access an entire speech by Mr. Buckingham, as well as an entire debate.

```
buckingham_sentences <- left_join(buckingham_sentences, file_metadata_1830)


buckingham_sentences <- buckingham_sentences %>%
  select(sentence_id, speech_id, debate_id, speechdate, debate, text)
```

```
head(buckingham_sentences)
```

```
##      sentence_id speech_id debate_id speechdate
## 1 S3V0018P0_6610     87635      9789 1833-05-31
## 2 S3V0018P0_6625     87635      9789 1833-05-31
## 3 S3V0018P0_6643     87635      9789 1833-05-31
## 4 S3V0018P0_6646     87635      9789 1833-05-31
## 5 S3V0018P0_6657     87635      9789 1833-05-31
## 6 S3V0018P0_6658     87635      9789 1833-05-31
##                                          debate
```

```
## 1 MINISTERIAL PLAN FOR THE ABOLITION OF SLAVERY. ]
## 2 MINISTERIAL PLAN FOR THE ABOLITION OF SLAVERY. ]
## 3 MINISTERIAL PLAN FOR THE ABOLITION OF SLAVERY. ]
## 4 MINISTERIAL PLAN FOR THE ABOLITION OF SLAVERY. ]
## 5 MINISTERIAL PLAN FOR THE ABOLITION OF SLAVERY. ]
## 6 MINISTERIAL PLAN FOR THE ABOLITION OF SLAVERY. ]
##
## 1
## 2
## 3
## 4
## 5
## 6 There was not the slightest ground, therefore, for the assumption, that, if liberated, the
```

Here we filter for a speech ID to view his sentences belonging to a single speech.

```
entire_speech <- buckingham_sentences %>%
  filter(speech_id == 89834) %>%
  select("text")
```

```
print(entire_speech$text)
```

```
## [1] "If the restraint were intended to prepare the slave for freedom, then he should say tha
## [2] "But for the artisans and artificers, who in large numbers maintained themselves by thei
## [3] "He was one of those who contended that no compensation should be paid till loss could b
## [4] "But for the cultivators, he believed no compensation would ever be required, as it had
```

We can do something similar to see all the speeches belonging to a specific debate based on the "debate_id" field.

```
entire_debate <- buckingham_sentences %>%
  filter(debate_id == 9789) %>%
  select("text")
```

```
print(entire_debate$text)
```

```
## [1] "But, for the present, he would content himself with saying, that the two principal fea
## [2] "This demand, however, was opposed by the Ministers, as well as by the West Indians, on
## [3] "As to the indolence of the slaves, and their incapacity or unwillingness to labour for
## [4] "All men disliked to labour more than was necessary to obtain for them the enjoyments o
## [5] "The other work was one entitled \"\"Wages, of the Whip, \"\" drawn up by Mr. Conder, a
## [6] "There was not the slightest ground, therefore, for the assumption, that, if liberated,
## [7] "Members be ignorant of the fact, that in our own immense empire of the East Indies, an
## [8] "If, instead of the unjust preference which had hitherto been given to the produce of s
## [9] "If loss should actually accrue for the first few years, from the change from a system
```

Reading a sentence within the context of an entire speech as a string of speeches gives us a much richer sense of the back-and-forth of parliamentary speeches, as well as the positions taken by each speaker. We recommend that the analyst of history alternate between "close reading" of these speeches in context alongside the "distant reading" of words and phrases charted over time.

Women in 19th-century Parliament

For the most part, an analysis of speakers in Parliament will take the researcher down the road of the many elite white men who assumed powerful positions or passed honorary titles to their sons. Only 12 named women spoke in Parliament throughout the 19th-century. One woman who spoke the most was Mrs. Walrand from the 1824 debate, "Motion Respecting the Trial and Condemnation of Missionary Smith at Demerara." Exploring her role in this debate can give us insight and speculation into the criteria that had to be set in place for a woman to appear as a speaker and instrument to

the power of Parliament.

"Motion Respecting the Trial and Condemnation of Missionary Smith at Demerara" reviews the trail, conviction, and death of John Smith, an Methodist missionary from England assigned to the British slave colony of Demerara in South America. Mr. Smith's trial accused him of assisting a slave rebellion that took place in Demerara in August of 1823. MP Brougham brought the topic of Mr. Smith's trial before Parliament to argue that the legal procedings convicting Mr. Smith were unjust and strayed from the guidance of Britain to the leaders of its Demerara colony. One could read his testimony as an early foundation for Britian's movement towards the Slavery Abolition Act of 1833, but his concerns–as well as the testimonies of other speakers–are nonetheless intertwined within a biased legal institution.

Mrs. Walrand spoke as a witness to the slave rebellion and served as a character witness condemning Mr. Smith. According to her testimony, she saw first hand the brutal violence of the rebells, having been a "defenceless lady" fired at by these "savages."

```
data("hansard_1820")
data("speaker_metadata_1820")
data("file_metadata_1820")


mrs_walrand_speaker_metadata <- speaker_metadata_1820 %>%
  filter(str_detect(speaker, regex('Mrs(.*)Walrand', ignore_case = T)))


mrs_walrand_sentences <- left_join(mrs_walrand_speaker_metadata, file_metadata_1820, by = "sen
  left_join(., hansard_1820, by = "sentence_id") %>%
  select(debate_id, text)
```

```
head(mrs_walrand_sentences)
```

```
##   debate_id
## 1      5142
## 2      5142
## 3      5142
## 4      5142
## 5      5142
## 6      5142
##
## 1
## 2
## 3 I entreated the guard, in the name of every principle of humanity, just to let me send to
## 4
## 5
## 6
```

```
entire_debate <- hansard_1820 %>%
  left_join(file_metadata_1820, hansard_1820, by = "sentence_id") %>%
  filter(debate_id == 5142) %>%
  select("text")
```

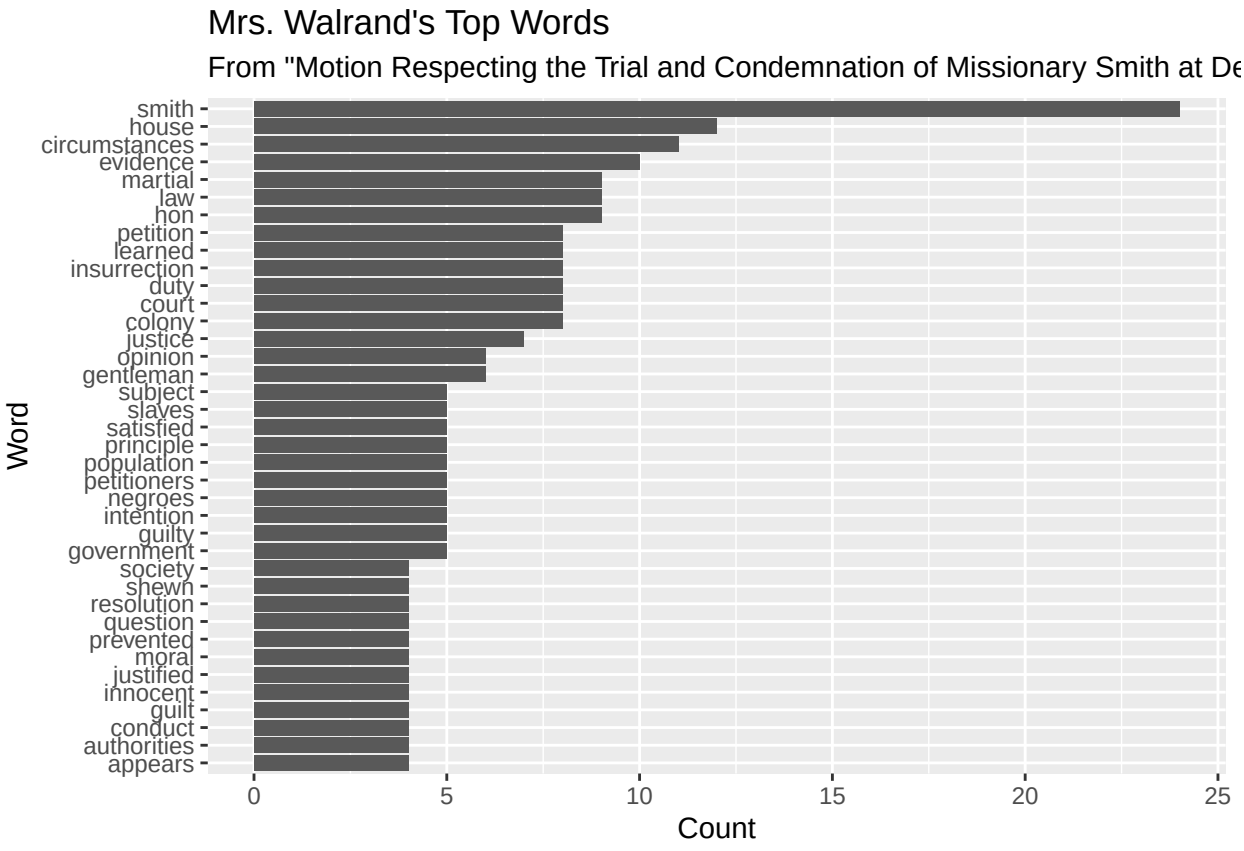
```
head(entire_debate$text, 20)
```

```
##  [1] "rose, and addressed the House to the following effect:-"
##  [2] "; I confess that, in bringing before this House the question on which I now rise to ac
##  [3] "I cannot conceal from myself, that, even in quarters where one would least have expect
##  [4] "Many persons who have, upon all other occasions, been remarkable for their manly hosti
```

```
##  [5] "Nay, they would fain fasten upon any excuse to get rid of the subject. \"\""
##  [6] "What signifies inquiring, \"\" say they, \"\"into a transaction which has occurred in
##  [7] "As if distance or climate made any difference in an outrage upon law or justice."
##  [8] "One would have rather expected that the very idea of that distance; the circumstance
##  [9] "Then, says another, too indolent to inquire, but prompt enough to decide, \"\"It is t
## [10] "* From the edition published by Hatchard and Son, with the sanction of the London Mis
## [11] "ject;"
## [12] "but then every body knows how those petitions are procured, by what descriptions of p
## [13] "And, after all, it is merely about a poor missionary!\"\""
## [14] "It is the first time that I have to learn that the weakness of the sufferer; his unpr
## [15] "But, it is not enough that he was a Missionary; to make the subject still more unpala
## [16] "I hasten to this objection, with a view at once to dispose of it."
## [17] "Suppose Mr. Smith had been a methodist; what then?"
## [18] "Does his connexion with that class of religious people, because, on some points essen
## [19] "Are British subjects to be treated more or less favourably in courts of law; are they
## [20] "Had he belonged to the society of the methodists, and been employed by the members of
```

```
mrs_walrand_top_words <- left_join(mrs_walrand_speaker_metadata, hansard_1820) %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  filter(is.na(as.numeric(word))) %>%
  group_by(speaker, word) %>%
  summarize(n = n()) %>%
  arrange(desc(n)) %>%
  top_n(30)

ggplot(data = mrs_walrand_top_words,
```

```
      aes(x = reorder(word, n), y = n)) +
geom_col() +
coord_flip() +
labs(title = "Mrs. Walrand\'s Top Words",
      subtitle = "From \"Motion Respecting the Trial and Condemnation of Missionary Smith at D
      x = "Word",
      y = "Count")
```



Exercises

1) Use the techniques shown above to move between a distant reading of the top words stated by Mrs. Walrand or Louisa Demont in all of her speeches to the full debate. (As a reminder, you

Name	Year	Name	Year
Mrs. (Mary Ann) Clark	1809	Mrs. Disraeli	1856
Mrs. Bridgeman	1809	Ms. Cunninghame Graham	1887
Miss Mary Ann Taylor	1809	Mrs. Dillon	1902
Louisa Demont	1820	Mrs. M'Govern	1902
Franchette Martigner	1820	Mrs. Mitchel-Thomson	1907
Mrs. Walrand	1824	Ms. Cave	1907

can view all of a speaker's speeches by filtering for just that speaker.) How does transitioning between the different modes of reading text provide insight into this specific debate? How does this illuminate the role of women in 19th-century Parliament?

2) As we discussed in the beginning of this chapter, analyzing the individual or collective speeches of men in Parliament can give insight into the workings of the Parliament as an institution of power. We can explore the roles of elite white men as they exerted their will–and the will of their peers–through the Parliamentary institution. We can also explore the roles of women in this space and examine how women differ in their contributions, as they were prohibited from becoming members of Parliament. Further, the woman's presence had to be mediated by one of the MPs, as they could not be there without being selected or approved. In this way, women were treated as instruments of power in Parliament.

We can further explore how the role of women manifested in Parliament. First, count the total number of words women spoke in Parliament. To do this, refer to the table below, which visualizes the women in Parliament and the year in which they spoke. Can you find a pattern across the kinds of topics women are brought to Parliament to speak on?

Note: The way in which their names are recorded in this table might not reflect the way in which their names appear in the debates. This is because the names have been consolidated into one representation

3) Adjust the above code to search for a different office title. What can you infer about the different MPs who held this title? Can we support our inferences through close readings of

their speeches?

To get you started, here are a few office positions: - Chancellor of the Exchequer - Prime Minister - Attorney General - Lord Chancellor

4) Visualization N shows the most verbose MPs for every year of the decade 1830. The counts are based on the "speaker" column–that is, the name of the speaker as it was originally transcribed in Hansard. Adjust the code so that instead you visualize the "suggested_speaker" column. This column contains the names of the speakers after they went through a "disambiguation" processes, that, where possible, assigned speakers with a standardized name and ID number. How are these two visualizations different? What are some pros and cons about visualizing the original speaker names and the disambiguated speaker names?

5) Above, we explored the contributions of four major speakers about slavery. But the names that we explored in Figure N don't match those that we saw were most important to the two peaks of debate in 1833 and 1838, shown in Figure M. Use the code to generate Figure N with an the list of names we extracted from Figure M: Mr. Secretary Stanley, the Duke of Wellington, Mr. Buckingham, Mr. Fowell Burton, the Earl of Ripon, Lord Brougham, and Lord Gleneig. What do you learn from this exercise? How are the results different than those in Figure N?

Chapter 5: Grammatical Analysis of Natural Language Speech

This chapter introduces ways to study patterns in language change that happen not at the level of words or phrases, but at the level of sentence grammar. It introduces a distinction between lexical analysis–that is, the analysis of words and their meaning–and syntactic analysis, or the analysis of the relationships between words. As this chapter will demonstrate, a syntactic analysis can lend insight into text that lexical studies alone cannot by making possible comparative and diachronic studies of how different entities were written about as subjects and objects.

To understand the usefulness of syntactic analysis for studying historical text, consider a sentence uttered by Mr. Johnstone, a speaker in Parliament from the year 1811:

```
"But it is urged that landlords may exercise a grievous oppression over tenants."
```

Among the relationships embedded in the sentence is a statement where the subject "landlords" is performing the verb "exercise," and the object of "exercise is"oppression." Word counts alone may not capture the landlord's action, or the tenant's received treatement, because these attributes are not encoded in any one word but instead within the relationships between the subject, verb, and object of the sentence.

In this chapter, we call this method of mining words based on their grammatical functions "part-of-speech extraction."

A Closer Look at Parts-of-Speech and Syntax

A word's role in a sentence is indicated by its part-of-speech, which include lexical categories such as verb, noun, adjective, and more. Syntax, on the other hand, describes how words relate to one-another and how they combine to form larger units such as sentences.

For the sake of extracting grammatical parts-of-speech, its important to understand that words in a sentence depend on one-another syntactically. Therefore, even if two words are both nouns, such as "landlords" and "tenants," they can play different syntactic roles. In this example, "landlords" is the subject, or the noun performing an action, and tenants is the object, or the noun being acted upon by the subject.

In previous chapters we extracted n-grams, or words that appear next to each-other in a sentence. N-grams can provide insight into a word's context, but they don't necessarily tell us other nuances expressed in a sentence, such as what kind of action is being performed, or how the different nouns are being described with which adjectives. Extracting words based on their grammatical functions allows us to extract words that share a meaningful relationship to one-another and express these nuances, but do not necessarily appear next to each other in a sentence.

In this chapter we will use part-of-speech extraction to mine for adjective-noun relationships and triples relationships. In the above sentence these would include the adjective-noun pairs "grevious oppression" and the triple "landlords excersize oppression." As we will show, the common problem that part-of-speech extraction aims to solve is a way for analysts to mine for the meaning in a sentence that is encoded in the relationship between units of text.

posextractr

We will use posextractr to extract part-of-speechs and analyze words and their grammatical relationships. posextractr uses spaCy NLP, a natural language processing library that provides tools for identifying the parts-of-speech and syntactic relationships within a sentence. spaCy does

this by parsing words with a pre-trained statistical model that guesses a word's grammatical role in a sentence. Calling spaCy "pre-trained" just means someone else has already put in the effort of teaching spaCy which words are likely to be nouns, verbs, subjects, objects, adjectives and so forth.

Using a pre-trained model is convenient, and it often means that users can move straight into analysis becuase they won't be spending large amounts of time teaching the model themselves. It's important to note, however, that pre-trained models are often trained on different data sets than the one used by the analyst, so these models might yield imperfect predictions when run on a historical corpus like the 19th-century Hansard corpus. It's the job of the analyst to make sure that the "black-box" model is appropriate and meaningful for a specific use-case. Imperfect predictions can be resolved through manual correction or by retraining the model on the desired corpus. Because the latter can be time intensive, manual correction is often a preferable option depending on the analyst's resources.

posextractr Installation

Like the other libraries introduced by this book, posextractr can be installed using `install.packages("posextractr")`. Unlike other libraries, however, posextractr requires additional set-up. For first time use, after loading the library, posextractr will need to be installed using `posextract_install()`.

```
## [1] "Initialized"
```

```
library(posextractr)


posextract_install()
posextract_initialize()
```

After its first time installation, posextractr will only need to be loaded with `library(posextractr)` and then initialize posextractr using `posextract_initialize()`. If you try to use posextractr

without initialization you will receive an error.

Syntactic Dependency Parsing

By introducing syntactic analysis, this chapter takes a turn from methods that use "white-box" statistics–that is, statistics whose steps can be traced by a human–to the "black-box" statistics of syntactic dependency parsing. Methods might be refered to as "black-box" when a human, alone, cannot reconstruct the steps used by the computer to arrive at an output. In the case of syntactic dependency parsing, finding the probability of a word's part-speech or syntactic dependencies in a sentence is considered black-box because of the complexity of the statistical model.

These methods can be powerful because they can leverage computational resources beyond the natural abilities of a human analyst. However, anytime a computer makes predictions without continuous human input, it is important to carefully explore the results to make sure the computer did not make problematic assumptions.

Dependency Grammar

posextractr leverages syntactic dependency parsing to understand the grammatical structure of a sentence. Dependency grammar represents words in a sentence as though they are connected to each other by a chain-line structure. The relationships between words are expressed by directed arcs. According to dependency grammar, every word has a "head" and one or more "children." The head is the word that is depended upon, whereas the children are the dependents.

To see what I mean, we can return to the sentence, "But it is urged that landlords may exercise a grievous oppression over tenants." The head of noun "landlords" is the verb "exercise," which has numerous children including: "may", "that," and "oppression." We can also see that the adjective "grevious" is the child of the noun "oppression."

posextractr performs part-of-speech extraction by identifying words that share a grammatical

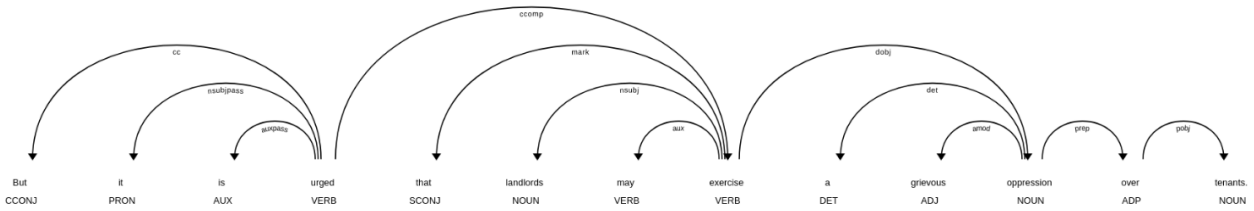


Figure 1: caption

relationship based on these tags and arcs.

In the following section we will focus our search to extract grammatical relationships from text.

Analyzing Grammatical Pairs

For our first exercise in syntactic analysis, we will extract adjective-noun pairs to better understand how women were characterized by Parliamentarians in 1870.

Preprocessing the Hansard Data

Parsing words with their parts-of-speech and their syntactic dependencies uses a lot of computer resources. We can preprocess the data to make it easier to run on a laptop.

We can start by filtering our data for just sentences with the words “woman” or “women.”

```
library(tidyverse)
library(hansardr)


data("hansard_1870")


hansard_woman_1870 <- hansard_1870 %>%
  filter(str_detect(text, regex("woman|women", ignore_case = T)))
```

```
head(hansard_woman_1870)
```

```
##      sentence_id
## 1 S3V0199P0_1775
## 2 S3V0199P0_1894
## 3 S3V0199P0_2429
## 4 S3V0199P0_2744
## 5 S3V0199P0_5203
## 6 S3V0199P0_5471
##
## 1
## 2
## 3
## 4
## 5
## 6 Then there are other questions which will have to be dealt with when the Bill is in Commi
```

We do this with the nested functions `filter(str_detect())`. By adding `regex()` we can also use the argument `ignore_case = T` to perform a case-insensitive match so both "Woman" (with a capital "W") and "woman" (with a lower-case "w") are returned.

We are going to analyze the adjective-noun pairs for the year 1879. To filter the 1870s data to just one year, we need to access the `speechdate` column. We can do this by joining `hansard_woman_1870` with `debate_metadata_1870`.

```
data("debate_metadata_1870")


debate_metadata_1870 <- debate_metadata_1870 %>%
  select(sentence_id, speechdate)
```

```
hansard_woman_1870 <- left_join(hansard_woman_1870, debate_metadata_1870)
```

```
head(hansard_woman_1870)
```

```
##      sentence_id
## 1 S3V0199P0_1775
## 2 S3V0199P0_1894
## 3 S3V0199P0_2429
## 4 S3V0199P0_2744
## 5 S3V0199P0_5203
## 6 S3V0199P0_5471
##
## 1
## 2
## 3
## 4
## 5
## 6 Then there are other questions which will have to be dealt with when the Bill is in Commi
##   speechdate
## 1 1870-02-24
## 2 1870-03-03
## 3 1870-03-07
## 4 1870-03-10
## 5 1870-02-14
## 6 1870-02-14
```

Because we joined `debate_metadata_1870` to `hansard_woman_1870` on the left side, we retain

metadata for just the sentences that have the word woman.

We can now filter for dates that start with just the year 1879 (as denoted by the carrot ^ symbol). For the sake of keeping our data set as small as possible, after we filter for our desired time frame we can select just the "text" column given that we do not need other metadata for our current analysis.

```
filtered_hansard_woman_1870 <- hansard_woman_1870 %>%
  select(text)
```

Now that we have our subset of the Hansard data, which is made up of sentences with just our keyword, we can more easily use posextractr to extract grammatical constructions from sentences.

Extracting Adjective-Noun Pairs from Sentences

Examining language at the level of sentence grammar provides a specific vantage point for exploring language trends. Instead of extracting words that occur next to one-another we can extract words that may exist in a different order but share a grammatical relationship.

Below we pass `extract_adjective_noun_pairs()` the vector from our data subset from which we wish to extract adjective-noun pairs. We also specify that we wish to lemmatize our results. During lemmatization, an inflected word is transformed to its base form. In the below example, plural "women" is transformed to "woman." We decided lemmatize our results to count words with the same meaning similarly.

```
adjective_noun_pairs <- extract_adj_noun_pairs(filtered_hansard_woman_1870$text, lemmatize = T
```

```
head(adjective_noun_pairs)
```

```
##   verb_neg neg_det     adjective    noun
## 1     <NA>    <NA>          only  object
## 2     <NA>    <NA>     excessive  labour
## 3     <NA>    <NA>       married   woman
```

```
## 4     <NA>    <NA>       married   woman
## 5     <NA>    <NA>       present     law
## 6     <NA>    <NA> revolutionary measure
```

The output of `extract_adj_noun_pairs` is a dataframe with columns containing the adjective and the noun. It also has columns for "verb_neg" and "neg_det," which words like "no," "not" and "never," if they modify the adjective-noun pair.

Visualize

For the sake of our visualization we can use `unite()` to combine the words in each column into a single item.

```
adjective_noun_pairs <- adjective_noun_pairs %>%
  unite(adj_noun_pair, 1:ncol(adjective_noun_pairs), sep = " ", na.rm = TRUE)
```

```
head(adjective_noun_pairs)
```

```
##           adj_noun_pair
## 1           only object
## 2      excessive labour
## 3         married woman
## 4         married woman
## 5           present law
## 6 revolutionary measure
```

And we can use `str_to_lower()` to transform the data to lowercase, making sure we don't count the word "Woman" (with capital "w") separately from the word "woman" (with a lower-case "w") when we visualize our results.

```
adjective_noun_pairs$adj_noun_pair <- str_to_lower(adjective_noun_pairs$adj_noun_pair)
```

We extracted adjective-noun pairs from just sentences with the words "woman" or "women," but there are still other adjective-noun pairs mentioned in these sentences. For our visualization, we can filter for just the rows that contain the object "woman" using the nested functions `filter(str_detect()`. By adding the `$` symbol to the end of the word "woman", we are telling our code to only keep rows that end with this word.

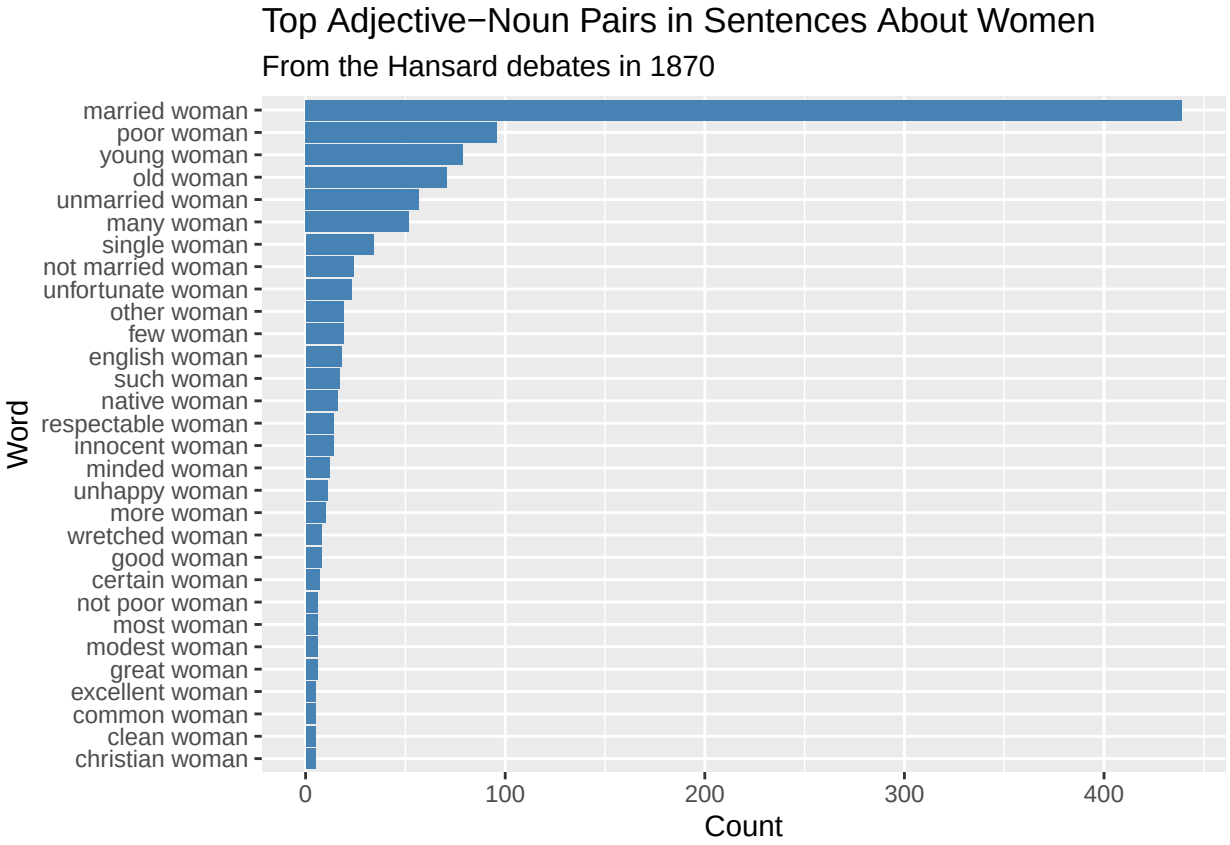
```
adjective_noun_pairs <- adjective_noun_pairs %>%
  filter(str_detect(adj_noun_pair, "woman$"))
```

Our bar chart visualizes the top adjective-noun pairs for the lemma woman in throughout the 1870s.

Without considering the historical context of the extracted words, our bar chart of top adjective-noun pairs might suggest that women in 1870 were particularly "poor" or "unmarried" or "niave," each of which are words found on our graph. However, historical interpretation encourages us to not interpret these results at face-value, but to consider how representations of the past come to us through documents that survived an unknowable reality of the past as it was lived. The best interpretation of this diagram is as a tool for understanding the beliefs of members of Parliament who were investigating the conditions of working women in the era of the Contagious Disease Acts.

```
top_adj_noun_pairs <- adjective_noun_pairs %>%
  count(adj_noun_pair) %>%
  arrange(desc(n)) %>%
  slice(1:30)


ggplot(data = top_adj_noun_pairs) +
  geom_col(aes(x = reorder(adj_noun_pair, n),
               y = n),
```

```
          fill = "steel blue") +
coord_flip() +
labs(title = "Top Adjective-Noun Pairs in Sentences About Women",
     subtitle = "From the Hansard debates in 1870",
     x = "Word",
     y = "Count")
```



Analyzing Grammatical Triples

posexractr gives us options to extract more grammatical patterns than just adjective-noun pairs. Like in our introductory example, posextractr can be used to extract grammatical triples, which are made up of the subject-verb-object and subject-verb-predicate adjective relationships found in a

sentence.

Let's return to our example sentence:

```
"But it is urged that landlords may exercise a grievous oppression over tenants."
```

Extracting just key parts-of-speech allows us to abstract the idea "landlords exercise oppression" even when it is stated with any number of additional words (such as articles or conjunctions) and regardless of what order these words appear in a sentence. In this way, by reducing the sentence to just the subject, verb, and object or adjective predicate, we can look for trends that might otherwise be buried within natural language speech.

Extracting Grammatical Triples

We can extract grammatical triples with `extract_triples()`. `extract_triples()` takes four arguments: the vector as well as three boolean values for `combine_adj`, `lemmatize`, and `add_aux`. The argument `combine_adj` gives the user the ability to include adjectives as part of the extracted triples. `lemmatize` gives the user the ability to return lemmatized triples. `add_aux` gives the user the ability to extract the auxiliary verbs often used to indicate future and past tenses, such as "will excersize" or "had excersized."

For our analysis we will just set `lemmatize` to true. We are using lemmatization to group like-triples so they can be analyzed as a single item. We can abstract the triple, "landlord exercise oppression" even when the verb is presented in different grammatical forms such as in its present-progressive tense, "landlords exercising," or its past tense, "landlords exercized." An analyst can thus accurately count the number of times the lemma "exercise" is applied to the object "oppression," and how many times the subject "landlord" is performing an action.

`extract_triples()` is a more computationally intensive function that will take longer to run.

```
triples <- extract_triples(hansard_woman_1870$text,
                           combine_adj = F,
                           lemmatize = T,
                           add_aux = F)
```

```
head(triples)
```

```
##   subject_negdet      subject neg_adverb aux_verb    verb  poa object_negdet
## 1           <NA>    education       <NA>     <NA>      be <NA>          <NA>
## 2           <NA>        which       <NA>     <NA>     aim <NA>          <NA>
## 3           <NA>        which       <NA>     <NA> protect <NA>          <NA>
## 4           <NA> Englishwoman       <NA>     <NA>  remain <NA>          <NA>
## 5           <NA> Englishwoman       <NA>     <NA>   marry <NA>          <NA>
## 6           <NA>         they       <NA>     <NA>  murder <NA>          <NA>
##   object_adjectives       object object_prep object_prep_noun     rule
## 1              <NA>       object        <NA>             <NA>  <rule2>
## 2              <NA>         also        <NA>             <NA>  <rule2>
## 3              <NA>        woman        <NA>             <NA>  <rule1>
## 4              <NA> Englishwoman        <NA>             <NA>  <rule2>
## 5              <NA>        alien        <NA>             <NA>  <rule3>
## 6              <NA>        woman        <NA>             <NA>  <rule2>
```

triples is a dataframe with several columns. Each column corresponds with a lexical category. Note how the extracted triples do not contain parts-of-speech that might be considered superficial for analysis, such as articles "the" or "an", which are not required, in many cases, to capture the main ideas in a sentence.

The following describes each category.

- subject_negdet: for whether the subject is modified by a negative determiner such as “no,” “not,” or “never.”
- subject: the noun in the sentence performing action.
- neg_adverb: for whether the verb in the sentence is negated, such as with “no,” “not,” or “never.”
- aux_verb: often used for communicating future (“will”) tense and past (“had” or “has”) tense.
- verb: the action performed in a sentence.
- poa: the preposition or agent in a sentence.
- object_negdet: for whether the object is modified by a negative determiner such as “no,” “not,” or “never.”
- object_adjectives: any adjectives modifying the object.
- object: the noun in a sentence receiving action.
- object_prep: a prepositional phrase that depends on the object.
- object_prep_noun: the noun of the prepositional phrase depending on the object.

For our analysis we are just interested in triples with the subject “woman.”

```
triples_with_woman_as_subject <- triples %>%
  filter(str_detect(subject, regex("woman", ignore_case = TRUE))) %>%
  select(!"rule")
```

```
head(triples_with_woman_as_subject)
```

```
##   subject_negdet      subject neg_adverb aux_verb      verb  poa object_negdet
## 1           <NA> Englishwoman       <NA>     <NA>    remain <NA>          <NA>
## 2           <NA> Englishwoman       <NA>     <NA>     marry <NA>          <NA>
## 3           <NA>        woman       <NA>     <NA> reinstate   in          <NA>
## 4           <NA>        woman       <NA>     <NA>   bargain with          <NA>
## 5           <NA>    forewoman       <NA>     <NA>     apply  for          <NA>
```

```
## 6           <NA>    forewoman       <NA>     <NA>     apply   in          <NA>
##   object_adjectives       object object_prep object_prep_noun
## 1              <NA> Englishwoman        <NA>             <NA>
## 2              <NA>        alien        <NA>             <NA>
## 3              <NA>        right        <NA>             <NA>
## 4              <NA>       cabman        <NA>             <NA>
## 5              <NA>       summon        <NA>             <NA>
## 6              <NA>  consequence        <NA>             <NA>
```

We can "unite" the parts-of-speech, making them easier to work with for analysis.

```
triples_with_woman_as_subject <- triples_with_woman_as_subject %>%
  unite(full_triple, 1:ncol(triples_with_woman_as_subject), sep = " ", na.rm = TRUE)
```

```
head(triples_with_woman_as_subject)
```

```
##                       full_triple
## 1 Englishwoman remain Englishwoman
## 2         Englishwoman marry alien
## 3         woman reinstate in right
## 4        woman bargain with cabman
## 5       forewoman apply for summon
## 6   forewoman apply in consequence
```

Exploring Our Results

Extracted triples allow the analyst to aggregate a great deal of information about the actions of women as described by speakers in Parliament: how often this triple uttered; who uses it; and what other language is used in a similar context. We can also perform different kinds of analyses with our extracted triples. We could compare times in which a woman is imaginged as exercising "franchise"

full_triple	n
woman have vote	48
woman have right	29
woman be able	16
woman be interested	15
woman be more	14
woman be so	14
woman exercise franchise	14
woman have property	14
woman take part	11
woman be subject	10

versus the times that a woman is instead imagined as being subjected.

One way to explore these different triples extractions is by visualizing them in a table.

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
top_10_triples_with_woman_as_subject <- triples_with_woman_as_subject %>%
  count(full_triple) %>%
  arrange(desc(n)) %>%
  slice(1:10)


top_10_triples_with_woman_as_subject %>%
  arrange(desc(n)) %>%
  kbl() %>%
  kable_material(c("striped", "hover"))
```

As shown in the above table, the top triple with the word woman as subject in the 1870's British Parliamentary debates is "woman have vote" followed by "woman have right." Many other top triples discuss women's agency, such as "woman exercise franchise" and "woman have property."

We can use KWIC to return to the original documents and contextualize our analysis. Provided with the word "vote," we can view the surrounding discussion.

```
library(quanteda)
```

```
## Package version: 3.2.4
## Unicode version: 14.0
## ICU version: 70.1
## Parallel computing: 8 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
```

```
hansard_woman_1870_corpus <- corpus(hansard_woman_1870, text_field = "text")


vote_kwic <- kwic(tokens(hansard_woman_1870_corpus),
                  "vote",
                  window = 8,
                  case_insensitive = TRUE)
```

```
vote_kwic
```

```
## Keyword-in-context with 418 matches.
##      [text7, 43]    possessed of the qualification established by Parliament to
##    [text140, 41]              description of property which now entitles men to
##    [text185, 39]                                    right, she has not a right to
##    [text186, 50]                           suffrage, women have as much right to
```

```
##     [text186, 69]                    woman having property now ought to have a
##     [text192, 53]                          now, because I think every man will
##     [text207, 19]                          this Bill we do not compel women to
##     [text209, 33]          Parliamentary elections the poor man retains his one
##     [text211, 32]                        , against it, women were admitted to
##     [text212, 11]                                up to the poll, they do not
##     [text225, 14]            to women having the money-qualification a right to
##     [text250, 12]                 not follow that women must necessarily have a
##     [text253, 11]                             the mere fact that a woman has a
##     [text253, 36]                       is no argument that she should have a
##     [text257, 12]                         be argued that she has the right to
##     [text266, 10]                       hope that this House will by a decisive
##     [text266, 39]                    a disability that women should not have a
##     [text272, 16]                          ) is that women should not have a
##     [text278, 11]                       that this Bill did not compel women to
##     [text288, 13]                  or unjust that married women should have a
##     [text296, 17]                       to deliberation than to action, for a
##     [text301, 75]                          ( Colonel Sykes ) reminded us - to
##     [text301, 94]                  - that disability is such that they cannot
##     [text303, 37]                        rights, on that very ground I should
##      [text325, 7]                                     If you give a woman a
##     [text382, 26]                   a form of government under which women could
##     [text383, 21]                  of the Constitution in extending the right to
##     [text385, 35]                     required as to the manner how women should
##      [text386, 8]                                  If they gave women a right to
##      [text402, 7]                               Friend that women should have a
```

```
##    [text439, 11]                    argued that, because women were admitted to
##    [text439, 22]                         elections, they ought to be allowed to
##    [text440, 54]                            country, that woman had a right to
##    [text443, 16]                 were to be universal suffrage then women should
##    [text446, 62]                               allow both a man and his wife to
##     [text451, 9]                         Last Session we gave women a right to
##     [text456, 6]                                      Women now have the local
##     [text459, 6]                                         Thus, while the local
##    [text459, 17]           comparatively small importance to women, the Imperial
##     [text514, 8]                           , married women would be entitled to
##    [text762, 11]                          a married woman would have a right to
##    [text777, 13]                                item of £ 500, included in the
##    [text861, 20]                          Lobby would ever regret that, by his
##    [text994, 11]                Government in its first Session enabled women to
##    [text994, 29]                                      him at the time -"" That
##     [text995, 7]                           Last Session women were enabled to
##   [text1000, 42]          who possessed the qualifications which entitled men to
##   [text1008, 36]                                  , than many men, when they did
##   [text1023, 37]                          , annoyed, and persecuted to give her
##   [text1033, 22]                   single town petitioned not to be allowed to
##   [text1033, 35]                amendment to the Constitution, allowing women to
##   [text1034, 10]                    whether it was proposed there to give the
##   [text1075, 17]                           in the parish; I apprehend she could
##   [text1088, 76]    definite manner in consequence of their disqualification to
##   [text1094, 11]                                  therefore, I am unable to give a
##   [text1099, 14]                  his argument that women might be permitted to
```

```
##   [text1103, 25]                    board elections, they should be allowed to
##   [text1121, 18]                         in the woman, and if the female
##   [text1125, 27]                   of political life - that of giving a
##   [text1152, 51]                     to be rated would be entitled to a
##   [text1156, 32]                of such property thereby obtain the right to
##   [text1172, 18]              enjoying the franchise, will give an unbiassed
##   [text1173, 31]                       , how can you refuse the right to
##   [text1181, 21]                    in the position which gives a man a
##   [text1197, 41]                     , upon which women would be qualified to
##   [text1242, 30]                  at the age of discretion were entitled to
##   [text1392, 38]                  at the age of discretion were entitled to
##    [text1481, 7]                   The question of giving the Parliamentary
##   [text1482, 15]                     in the abstract women have a right to
##   [text1510, 25]        time women advocated their claims to the Parliamentary
##   [text1520, 16]                to perform the humble function of giving a
##   [text1530, 17]             and heads of families possess already every local
##   [text1530, 30]               propose to give to this class the Parliamentary
##   [text1533, 72]                         in four or five years to cast a
##   [text1539, 16]                or not married women would be competent to
##   [text1545, 20]                    , her individuality, her property, her
##   [text1551, 27]                       to be rated would be entitled to a
##   [text1553, 55]                woman who had been entitled to the municipal
##   [text1553, 67]                 year just previous to an election lost that
##  [text1553, 104]               who would have been entitled to the municipal
##   [text1572, 21]                   400 women petitioned not to be allowed to
##   [text1572, 69]                satisfied women would be annoyed to have the
```

```
##   [text1584, 24]          had a property qualification, possessed an indirect
##   [text1588, 16]                      in every 6 possessed of the right to
##   [text1590, 19]                  a deaf and dumb man might interpret his
##   [text1590, 30]                       , and when he was dead might not
##   [text1620, 22]                    political duties, and if the right to
##    [text1654, 5]                                          If they gave a
##   [text1678, 54]                  any mode could be devised by which their
##   [text1687, 23]               and asking them to induce their husbands to
##   [text1688, 27]                is perhaps calculating how he can sell his
##   [text1700, 22]                     anything like all women, shall have a
##   [text1701, 71]                majority of women will not be entitled to
##   [text1728, 10]                           course, it will be said that a
##   [text1729, 15]                upon her marriage by being deprived of her
##   [text1752, 28]               women who had property should be allowed to
##   [text1754, 45]                  might be that women should be allowed to
##   [text1769, 11]            was virtually conceded when women were allowed to
##   [text1781, 15]                unfeminine for a woman to give a public
##    [text1787, 6]                                 Should women be allowed to
##    [text1795, 7]                                  Hence, some were urged to
##   [text1804, 65]             relevancy, unless it implied that the tenant's
##   [text1808, 29]                  of giving a woman a rent-charge a county
##   [text1823, 20]                       the question of fitness of women, I
##   [text1825, 11]                what is the objection to empowering women to
##    [text1827, 7]                                      To tell me you are to
##   [text1831, 18]               nature by allowing a woman having property to
##   [text1831, 36]                        face of nature to allow a woman to
```

```
##   [text1841, 25]                          Act, a married woman might have a
##   [text1842, 15]                        not wish to give a married woman a
##   [text1845, 29]                  occupied in this country, that he should
##   [text1849, 35]                             , he thought it his duty now to
##   [text1850, 30]                    no reason why he or the House should
##   [text1852, 22]          almost identical with those which gave the municipal
##   [text1852, 42]           woman married she no longer possessed the municipal
##    [text1929, 8]                                  At the present time, if a
##   [text2064, 16]                    in Italy, where it is understood they
##   [text2064, 50]                  into consideration; but if women were to
##   [text2064, 73]                                  far as I know, no one can
##   [text2064, 77]                           , no one can vote by proxy and
##   [text2069, 15]               why women should be entitled to a Parliamentary
##   [text2089, 23]              else who on some questions affecting women might
##   [text2095, 39]                in devising means to enable illiterate men to
##   [text2108, 27]                  , argued that to give women a Parliamentary
##   [text2110, 24]                       the claim of women to give a silent
##   [text2111, 33]                    attacked on the ground that it gives a
##   [text2111, 47]                         also, because it does not give a
##   [text2112, 37]                           - that women do not care for a
##   [text2113, 22]                whenever women have been allowed to exercise a
##   [text2114, 12]                 recent evidence that woman do care for the
##   [text2118, 21]                         say that women do not care for a
##   [text2120, 11]                       that women do not care to possess a
##   [text2124, 12]                         told that women do not care for a
##   [text2134, 34]                       that when a woman came to record her
```

```
##   [text2135, 28]             indefatigable, and intellectual woman was denied a
##   [text2159, 33]                          where they had now got the right to
##   [text2160, 11]             one of the greatest justifications for giving the
##    [text2161, 3]                                                   He should
##   [text2170, 25]       those whose education and independence qualified them to
##   [text2194, 28]                and have been authorized by the Legislature to
##   [text2208, 25]                 laws they have therefore an abstract right to
##   [text2209, 52]                   argue that therefore women have a right to
##   [text2228, 24]                           and that he must arm women with the
##   [text2232, 32] salutary and long-established demarcation between the right to
##   [text2233, 36]                    said that women might be well qualified to
##    [text2234, 7]                                  But to say that women might
##   [text2236, 19]                           men; but if giving women power to
##   [text2237, 38]                duly qualified by property were called upon to
##    [text2284, 7]                                  Unless a woman can obtain a
##    [text2286, 8]                    If this Bill contemplated making a woman
##   [text2286, 14]                           a woman vote who did not wish to
##   [text2306, 55]                  the Church that if they could exercise their
##   [text2311, 23]                        to me that to exclude women from the
##   [text2313, 14]                 agricultural labourer, and refuse to give a
##   [text2314, 16]                          , however ignorant, ought to have a
##    [text2317, 9]                 Gentleman can convince me that giving them a
##   [text2317, 27]                      or men less manly, I would immediately
##   [text2319, 28]                         also yet to learn that giving women a
##   [text2328, 36]                    conclusion that the woman ought to have a
##    [text2341, 9]              He thought that the question whether women would
```

```
##   [text2363, 24]                      and woman in the country should have a
##   [text2364, 45]                           they would, by a largo majority,
##   [text2705, 33]                             , then it is useless for us to
##   [text2774, 66]                   if they declined to interfere, he should
##   [text2822, 25]            the case of the proposed prohibitory liquor selling
##   [text2940, 12]                    , in consequence of women being able to
##   [text3045, 40]                      were suited, he must give his decided
##   [text3051, 31]                             a Scotch point of view; not to
##   [text3052, 18]                     education of women, he felt bound to
##   [text3128, 34]          of sufficient property qualifications in counties, to
##   [text3129, 30]                  opposed to the claim of married women to
##   [text3130, 29]                that section does not enable married women to
##   [text3131, 20]                  that married women should not be allowed to
##   [text3144, 56]                        could not give effect to them by her
##    [text3153, 8]                         He said that women ought not to
##   [text3154, 16]                           halt, and the blind ought not to
##   [text3154, 32]                  reasonable to say that women ought not to
##    [text3164, 9]                 It does not follow that because women could
##   [text3171, 14]                      hon. and learned Friend has promised to
##   [text3173, 21]                      perception of right and wrong, and will
##   [text3178, 37]                         confer - namely, the possession of a
##   [text3189, 16]                       Throne of the country, and that women
##   [text3190, 49]                        be adduced why she should not have a
##   [text3191, 15]                          sits on the Throne, and that women
##   [text3191, 29]                         boards, is a reason why they should
##   [text3193, 46]                       sooner or later, women should not only
```

```
##   [text3232, 18]                    the Queen's taxes, therefore they ought to
##   [text3247, 30]                  might be registered, though they should not
##   [text3263, 26]                          used was that women ought to have a
##   [text3266, 50]                          ; and the right of married women to
##   [text3285, 82]                         views on the subject, and might even
##   [text3291, 35]                         once in every three or four years to
##    [text3292, 9]                         Women have by the Common Law a local
##   [text3328, 16]                           are as well informed and as fit to
##    [text3334, 3]                                                      If you
##    [text3348, 5]                                    They allowed women to
##   [text3348, 15]                     Poor Law guardians; they allowed them to
##   [text3348, 25]               municipal councillors, and they allowed them to
##   [text3348, 39]                          ; yet they refused to allow them to
##   [text3391, 22]                          is to obtain for women the right to
##   [text3391, 35]                on the same conditions which entitle men to
##    [text3414, 4]                                          In claiming my
##   [text3414, 34]                     the women in my constituency asked me to
##   [text3421, 26]                 that the measure would enable married women to
##    [text3787, 5]                                        Beyond this, a
##    [text3802, 3]                                              Women do
##   [text3806, 39]                           right, she has not a right to
##   [text3807, 26]                       that they could not be trusted with a
##   [text3840, 41]                  that might follow when she had recorded her
##   [text3842, 19]                        , married women will claim a right to
##   [text3844, 14]                  those who think that a married woman should
##   [text3847, 29]                           , and that if women are allowed to
```

```
##   [text3850, 64]                                to the taxation, ought to have a
##   [text3884, 34]                                are taxed, they ought to have the
##   [text3886, 18]                  regards other women because she has not the
##   [text3886, 29]                     the simple reason that no woman has the
##   [text3887, 19]                     as regards men because she has not the
##   [text3887, 23]                                she has not the vote, for to
##    [text3932, 5]                                                I would give a
##   [text3933, 37]                               , and I would give that woman a
##   [text3934, 27]                     his character, shall have the right to
##    [text3978, 8]                         Unmarried women who were rated, by
##   [text3978, 30]                         in England - they had the right to
##   [text3978, 44]                           Poor Laws, they had the right to
##   [text3978, 58]                         affairs, and they had the right to
##    [text3991, 9]                       By this Bill the woman will lose her
##   [text3994, 30]                    for her separate use would be entitled to
##   [text3994, 60]                  by mutual arrangement, would be entitled to
##   [text3997, 15]                         would be much more fitted to give a
##   [text4023, 25]           possessed the qualifications for the exercise of a
##   [text4039, 37]                         say what will be the effect of the
##   [text4040, 33]              five women who would be enfranchised four would
##   [text4057, 15]                 rates, that they already enjoy the municipal
##   [text4057, 33]                    Overseers of the Poor, that they already
##    [text4064, 5]                                        Either you give a
##    [text4078, 8]                      Friend said everybody agreed to give the
##   [text4079, 14]                     in favour of conferring it upon women to
##   [text4097, 17]                               , and say that women may have a
```

```
##   [text4097, 28]                      councillors, but they shall not have a
##   [text4098, 32]               men in electing Members of Parliament who will
##   [text4099, 49]                       poor people, even though women do not
##   [text4104, 15]                  so strongly to women having the power to
##   [text4110, 33]                giving a woman having property the power to
##   [text4110, 56]                 of the Constitution by allowing a woman to
##   [text4111, 64]               the right and the privilege of recording his
##   [text4131, 26]                       in another house, why should not she
##   [text4132, 16]                  no harm to the country that women should
##   [text4142, 11]                          -"" Married women cannot claim to
##   [text4142, 26]                                  not they, as well as men,
##    [text4147, 5]                                        You have given a
##    [text4148, 9]                Having first granted this, that women shall
##   [text4148, 30]           Shall marriage disqualify, and if the unmarried
##   [text4151, 71]                whether married or unmarried, shall have a
##   [text4154, 34]                      where I live, whose Member or Members
##   [text4156, 13]              upon, that in Catholic Ireland every woman's
##   [text4156, 21]                       vote may be taken to be the priest's
##    [text4160, 3]                                          I should
##   [text4160, 22]                          the interests of the men; I shall
##    [text4163, 8]                       If the House believes that it cannot
##   [text4520, 95]                 householders and women, have a right to
##   [text4521, 11]             opinion, compound householders had a right to
##    [text4549, 3]                                      Members who
##   [text4601, 36]                        ability, and, therefore, he should
##   [text4632, 52]                 own Party had supported the Bill by their
```

```
##   [text4638, 72]                          classes, to whom alone the power to
##   [text4649, 12]                                   that a woman, if she had a
##   [text4649, 49]                       woman was to be content with giving a
##   [text4650, 77]                 of knowledge which would justify their giving a
##    [text4653, 9]                 There was another consideration - in giving the
##   [text4653, 44]                              exercise, in a manner, a double
##    [text4657, 8]                        It was argued that because women might
##   [text4658, 13]                      was every reason why women should have a
##   [text4665, 54]                         farms, because they could not give a
##   [text4672, 21]                    any demonstration of opinion, other than my
##   [text4684, 56]                    assert that that Bill itself would give the
##   [text4688, 31]                           that power, she should not have the
##   [text4691, 14]                                the Bill was, that to give the
##   [text4702, 34]                             , not one of whom was entitled to
##   [text4707, 14]                     argument as was now used against giving a
##   [text4708, 37]                  Legislature would pass a law enabling women to
##   [text4709, 23]                           a woman married, she would lose her
##   [text4710, 17]                      , having to choose between matrimony and a
##   [text4725, 39]                    been accorded to women they never ought to
##   [text4748, 15]           the intellectual capacity of women to exercise this
##   [text4755, 15]                     great influence, why cannot she give her
##   [text4803, 18]                        step further than to enable a woman to
##   [text4808, 37]                    infelicitous change, I would turn round and
##   [text4820, 57]                     thought that every woman was entitled to a
##    [text4822, 8]                           said that the admission of women to
##   [text4828, 25]                       they believed that women had the right to
```

```
##   [text4829, 21]                      women should think they had a right to
##    [text4832, 4]                                                 However, the
##   [text4851, 10]                            simply says, let a woman have a
##   [text4852, 15]              a woman in possession of the qualification to
##   [text4853, 30]                   in the possession of property may have a
##   [text4860, 16]                      obtained it that she shall not have a
##    [text4883, 8]                                   If a woman is to have a
##   [text5308, 15]            England women have been enabled to establish their
##    [text5358, 8]                                Now, a woman was entitled to
##   [text5486, 29]                                 Bill, did, I believe, once
##    [text5490, 7]                               We have admitted women to the
##    [text5491, 7]                               We have admitted women to the
##   [text5526, 25]                      rid of love because you give women a
##   [text5527, 14]                     because you give to women the power to
##   [text5535, 25]                           the matter, or wishes to have a
##   [text5536, 17]                          , no woman outside who asks for a
##   [text5537, 24]                    there was no objection in giving women a
##   [text5538, 11]          although gallantry originally prompted some of us to
##   [text5538, 37]                  the women themselves is to refuse them this
##   [text5547, 38]                     bring in a Bill which simply gives a
##   [text5550, 18]                      could deny that we should be giving a
##    [text5551, 6]                                   So far from giving a
##   [text5552, 27]                           , we cannot fail to see that a
##   [text5554, 28]                     of women and the flower of womanhood a
##   [text5555, 19]                    comes forward and says that women are to
##    [text5572, 6]                                   We are told that the
```

```
##   [text5574, 17]          largely interested where property is concerned have a
##    [text5585, 5]                                                 But, although a
##   [text5587, 18]                      whatever to do with the giving of a
##    [text5588, 6]                                              If we did give this
##   [text5596, 13]                         is that women go to the poll and
##    [text5598, 6]                                   Yet she was disqualified to
##   [text5612, 39]                     must not be permitted on any account to
##   [text5624, 51]                      with single women, must be allowed to
##   [text5628, 23]                       Bill does not attempt to give them a
##   [text5688, 41]                      think, state the grounds on which they
##   [text5704, 22]                       the suffrage, I should be disposed to
##   [text5719, 28]                            , and shall not be entitled to a
##   [text5727, 10]                          , again, women have been allowed to
##   [text5729, 21]                        because they are not fit to give a
##   [text5745, 17]                     language as to the result of giving the
##    [text5747, 6]                                              If the law denied a
##   [text5750, 17]                   world that at this moment gives women the
##   [text5751, 27]                           to know whether, if women had a
##   [text5752, 27]                       would allow a single woman to have a
##   [text5781, 12]                        of expediency do you give the man a
##    [text5782, 8]                       The general reasons why you give the
##    [text5786, 8]                              If you give a married woman a
##   [text5787, 16]                     hope that by giving a frivolous woman a
##   [text5790, 48]                              but that if they were, he would
##   [text5791, 13]                 that is a sufficient reason for withholding a
##   [text5791, 39]                       if you believe by giving a woman a
```

```
##   [text5794, 27]                 everywhere a large number of women desiring a
##   [text5795, 16]                                    it; but does every man use the
##    [text5889, 9]                                  If a householder had a right to a
##   [text5889, 22]                               every adult also claim a right to a
##   [text5889, 31]                                , and if they once admitted that a
##   [text5913, 43]                   the Country that women who are entitled to
##   [text5944, 20]                 being adult human creatures, were entitled to
##   [text5944, 42]                      an adult human creature, was entitled to
##   [text5960, 52]                  conferred on those women who are entitled to
##   [text6004, 31]                   the country that women who are entitled to
##   [text6010, 20]       very great distinction between giving them the municipal
##   [text6022, 12]                       told that women ought to be entitled to
##    [text6023, 2]                                                          The
##   [text6028, 84]                                go together, that I for one shall
##   [text6036, 61]                               go in to a ballot-box and record a
##   [text6040, 36]                   faculties such as may qualify anyone for a
##   [text6044, 26]                                - that woman ought not to have a
##   [text6059, 13]                        you admit women to the exercise of the
##   [text6074, 27]                         get such share in the Government as a
##   [text6088, 31]                               of the Realm, but in the supreme
##   [text6101, 22]                        guided by that confession; and when we
##   [text6102, 15]                  sentimental dislike to the idea of giving a
##   [text6109, 44]                               used to think that he had lost a
##   [text6123, 26]                        any possibility, can give the right to
##   [text6143, 82]                        have a large class of voters who could
##   [text6153, 29]                                , if women claim the right of a
```

```
##   [text6154, 42]                              - is that women do not desire a
##  [text6155, 100]                        to the present time with giving a silent
##   [text6160, 49]                      the election of whom women are entitled to
##   [text6172, 24]                  proprietors of property - women have aright to
##   [text6172, 62]                have the required qualification to enable her to
##   [text6172, 93]             associate themselves together gives her a right to
##   [text6174, 45]                          children - has a right not alone to
##   [text6176, 20]                     members of which women have a right to
##    [text6195, 8]                              It seems to be assumed that the
##   [text6228, 24]                       we speak of women not asking for the
##   [text6249, 11]                     elector would be more likely to give a
##   [text6271, 18]                  to society has resulted by allowing women to
##   [text6273, 29]                 can go in the polling-booth and register her
##   [text6275, 38]                  when she is called upon to exercise her
##   [text6277, 36]                     very limited class of women the right to
##   [text6279, 31]                     of married women would have the right to
##   [text6280, 60]                   under that Bill would have the right to
##   [text6281, 34]                        be such - is entitled to record her
##   [text6281, 55]                           Act of 1870, will be entitled to
##    [text6283, 9]                  The number of married women thus entitled to
##   [text6286, 64]                  , every married woman in this country might
##    [text6287, 7]                              When the right for women to
##   [text6287, 37]                     the same was said about the school board
##   [text6288, 54]                  woman under that Bill should be entitled to
##   [text6290, 22]                          is to obtain for women the right to
##   [text6290, 35]                  on the same conditions which entitle men to
```

```
##   [text6291, 19]                      will be if married women are allowed to
##    [text6315, 6]                                      He will not give the
##   [text6316, 25]                       you not only will not give her a
##   [text6316, 32]                            a vote, but you take away a
##   [text6316, 82]                        , you will be able to maintain your
##   [text6317, 30]                      success be yours, you will lose your
##   [text6318, 16]                        to say you are afraid to give the
##   [text6318, 53]                   her profession; whilst you would give the
##   [text6319, 52]                       women of this county are not fit to
##  [text6326, 117]                           it; and I undertook to give my
##   [text6358, 39]                     have thought for a moment of giving the
##   [text6359, 45]                            , for I am prepared to give a
##   [text6359, 66]                      same position as the man who has a
##   [text6361, 25]                         that if he was able to take the
##   [text6361, 38]                  England they would not do-sire to have the
##   [text6367, 11]                         on the whole it would be better a
##   [text6371, 18]                         would be to bring women in to the
##   [text6371, 26]                     vote, because they would always give a
##    [text6372, 8]                               I think if the women had to
##   [text6372, 14]                              had to vote, it would be a
##   [text6376, 37]                         women, do not wish to have the
##   [text6377, 54]                       the women who do not wish for the
##   [text6470, 68]                     Department, he wanted to know how a
##   [text6565, 69]                   begged leave to move the reduction of the
##
##  | vote | for Members of Parliament.
```

```
##  | vote | .
##  | vote | .""
##  | vote | as men; and, more than that
##  | vote | in a country, in which she may
##  | vote | according to what he takes to be the
##  | vote | .
##  | vote | , while the woman, who is in
##  | vote | in all our municipal elections.
##  | vote | with the quiet of the Ballot, but
##  | vote | .
##  | vote | .
##  | vote | with regard to the expenditure of the taxes
##  | vote | with regard to the large questions which are
##  | vote | as much as any other woman.
##  | vote | maintain the position we have hitherto maintained,
##  | vote | , but that it is rather a privilege
##  | vote | because of the turmoil of elections; because
##  | vote | ; but if a particular woman does not
##  | vote | , you have at present no intelligible basis
##  | vote | is but the expression of an opinion.
##  | vote | in the election of directors for the East
##  | vote | for Members of this House.
##  | vote | in favour of this Bill.
##  | vote | , are you prepared to make a woman
##  | vote | .
##  | vote | to women in the case of municipal elections
```

```
##  | vote | , and as to the consequences - whether
##  | vote | , how could they refuse them a seat
##  | vote | at Parliamentary elections is, that it would
##  | vote | at municipal elections, they ought to be
##  | vote | at Parliamentary elections.
##  | vote | in matters connected with the Poor Law,
##  | vote | .
##  | vote | in respect of property which is sufficiently valuable
##  | vote | in municipal elections; but we did not
##  | vote | universally, but it is f comparatively small
##  | vote | is of comparatively small importance to women,
##  | vote | is of great importance to them.
##  | vote | in respect of their property.
##  | vote | at meetings of the company in respect of
##  | Vote | for servants and charwomen.
##  | vote | to-day, he had helped to confer on
##  | vote | at municipal elections; an eminent Member observed
##  | vote | means the other,"" and the
##  | vote | for members of school boards, and to
##  | vote | , and whose claims had not been voluntarily
##  | vote | .
##  | vote | .
##  | vote | ; in Massachusetts an amendment to the Constitution
##  | vote | , has been rejected in the Legislature by
##  | vote | to married as well as to single women
##  | vote | and act in parochial affairs.
```

```
##  | vote | .
##  | vote | for a Bill with respect to which there
##  | vote | for such inferior bodies as Poor Law Guardians
##  | vote | for Members of that House.
##  | vote | made any noticeable difference in the character of
##  | vote | to a representative.
##  | vote | equally with the unmarried.
##  | vote | .
##  | vote | , the result of political conviction?
##  | vote | to those who are of her sex?
##  | vote | shall be entitled to exercise the franchise.
##  | vote | ; and therefore the right hon.
##  | vote | , it would be only a trust,
##  | vote | , it would be only a trust,
##  | vote | to women who are owners and occupiers of
##  | vote | .
##  | vote | , and at the present hour I could
##  | vote | at the poll, it should not be
##  | vote | , and he would propose to give to
##  | vote | .
##  | vote | at the poll.
##  | vote | if they had the qualifications, but my
##  | vote | .
##  | vote | equally with the un-married.""
##  | vote | for the whole year just previous to an
##  | vote | if she married before the election, and
```

```
##  | vote | had she been single, was not so
##  | vote | ; how in Massachusetts the Constitution would have
##  | vote | given them; and, lastly, how
##  | vote | .
##  | vote | , and 1 woman in every 60.
##  | vote | for him, and when he was dead
##  | vote | for herself?
##  | vote | now claimed for them were to be conferred
##  | vote | to women, he thought he had shown
##  | vote | could be taken without submitting them to the
##  | vote | .
##  | vote | to the best advantage, and the wife
##  | vote | ; he only asks that the minority shall
##  | vote | , simply because they are married, and
##  | vote | is given upon considerations of property, and
##  | vote | ?
##  | vote | ; and if he were not thoroughly persuaded
##  | vote | for Members of Parliament, and then seek
##  | vote | at the school board elections and municipal elections
##  | vote | should remember the unpleasant occupations to which women
##  | vote | on the question of a free breakfast table
##  | vote | for the Bill on the ground that women
##  | vote | was the property of the landlord.
##  | vote | could be created.
##  | vote | for this Bill for another reason.
##  | vote | ?
```

```
##  | vote | against this Bill because you do not think
##  | vote | for a Member of Parliament than flying in
##  | vote | for a town councillor or a member of
##  | vote | .
##  | vote | , why did not he say so in
##  | vote | against the second reading.
##  | vote | against it, and not to thrust a
##  | vote | for this ill-considered and incomplete Bill.
##  | vote | to women, and it had been decided
##  | vote | .
##  | vote | were taken on the subject, I venture
##  | vote | by proxy, and said if something of
##  | vote | by proxy they would lose the protection of
##  | vote | by proxy and vote in secret.
##  | vote | in secret.
##  | vote | .
##  | vote | entirely against their views.
##  | vote | not to continue to withhold the suffrage from
##  | vote | would be"" contrary to the experience
##  | vote | at the poll.
##  | vote | to married women and, also, because
##  | vote | to married women.
##  | vote | .
##  | vote | they have made use of the privilege.
##  | vote | , the House will perhaps allow me to
##  | vote | .
```

```
##  | vote | they ought at least to bear this in
##  | vote | I am reminded that two or three weeks
##  | vote | she was received by the working men with
##  | vote | simply because she was a woman, while
##  | vote | for the first time, they had voted
##  | vote | to any class was that that class would
##  | vote | for the second reading of the Bill in
##  | vote | , but the disqualification of women rested upon
##  | vote | for school boards in matters involving intellectual questions
##  | vote | .
##  | vote | is simply to ignore the whole career which
##  | vote | in self-defence and array them against the other
##  | vote | and the right to sit in that House
##  | vote | for those bodies, but not for so
##  | vote | for such a board and sit on it
##  | vote | unsexed them, the mischief had been done
##  | vote | every year for town councillors, and every
##  | vote | by property we do not wish to do
##  | vote | who did not wish to vote, it
##  | vote | , it would not find a more resolute
##  | vote | the Establishment of the Church would continue.
##  | vote | , simply because we think it would delay
##  | vote | to women, we shall be landed in
##  | vote | , no woman, however intellectual, ought
##  | vote | would make them in any respect less womanly
##  | vote | against the Bill.
```

```
##  | vote | , can in the slightest degree diminish the
##  | vote | .
##  | vote | for one political party or the other,
##  | vote | .
##  | vote | against this system, which breaks up so
##  | vote | from year to year upwards of £ 2
##  | vote | for the second reading of the Bill,
##  | vote | ?
##  | vote | , the constituency for a school board election
##  | vote | against the Bill.
##  | vote | as upon the higher education of women in
##  | vote | against the Bill on the grounds he had
##  | vote | at the election of Members of Parliament.
##  | vote | at Parliamentary Elections, and there is nothing
##  | vote | .
##  | vote | .
##  | vote | .
##  | vote | at Parliamentary Elections, because no woman had
##  | vote | , and it would be just as reasonable
##  | vote | , because they do not shoot partridges nor
##  | vote | , they would become active and ardent politicians
##  | vote | for a Bill, which he has described
##  | vote | for the right kind of representative to pass
##  | vote | .
##  | vote | for and sit upon school boards.
##  | vote | in the same way as a man.
```

```
##  | vote | for and sit upon school boards, is
##  | vote | , it is, of course, a
##  | vote | , but should be elected as Members of
##  | vote | , on the ground that representation and taxation
##  | vote | under the Bill.
##  | vote | because the sexes were equal
##  | vote | in contradistinction to their husbands was insisted upon
##  | vote | for the Bill.
##  | vote | by ballot at the election of a Member
##  | vote | , and of late years we have given
##  | vote | or to sit in this House, or
##  | vote | for this Bill in any shape you make
##  | vote | for Poor Law guardians; they allowed them
##  | vote | for municipal councillors, and they allowed them
##  | vote | for members of school boards; yet they
##  | vote | for the men who taxed their property as
##  | vote | for Members of Parliament on the same conditions
##  | vote | .
##  | vote | for the measure, he says that,
##  | vote | for it, I would do so.
##  | vote | .
##  | vote | may be exercised by a lunatic in a
##  | vote | now at municipal elections, and I believe
##  | vote | .
##  | vote | .
##  | vote | .
```

```
##  | vote | , and their claim must be conceded when
##  | vote | under any circumstances whatever is exceedingly small.
##  | vote | , they will ask to sit in the
##  | vote | in the election of the Representatives who frame
##  | vote | .
##  | vote | , for the simple reason that no woman
##  | vote | .
##  | vote | , for to vote and to rule have
##  | vote | and to rule have never been the prerogatives
##  | vote | to every householder in the Kingdom, man
##  | vote | .
##  | vote | and every woman shall be excluded.
##  | vote | had at that moment the right to deal
##  | vote | for the administration of the Poor Laws,
##  | vote | for the management of municipal affairs, and
##  | vote | for the regulation of the education of the
##  | vote | on her marriage, and the husband will
##  | vote | in a county; and that a married
##  | vote | in a town or borough, if qualified
##  | vote | for the good of the common weal than
##  | vote | that they should be excluded simply because they
##  | vote | of women in England, if they were
##  | vote | on this question diametrically opposite to the opinions
##  | vote | , that they may be elected as Guardians
##  | vote | for and sit upon school boards, that
##  | vote | to married women or you do not.
```

```
##  | vote | to women for members to sit upon the
##  | vote | for Members of this House.
##  | vote | for town councillors, but they shall not
##  | vote | for Members of Parliament.
##  | vote | for making men more sober and saving,
##  | vote | for Members of this House.
##  | vote | .
##  | vote | for a Member of Parliament, any more
##  | vote | for a town councillor or a member of
##  | vote | for one Party in the State or the
##  | vote | also?
##  | vote | , and I believe that is a thing
##  | vote | as householders, but why should not they
##  | vote | as lodgers?
##  | vote | to all young women who are unmarried who
##  | vote | , how can you answer any man who
##  | vote | , shall the married be disfranchised?
##  | vote | ?
##  | vote | for this Bill, at a recent municipal
##  | vote | may be taken to be the priest's vote
##  | vote | .""
##  | vote | for this measure, if I were voting
##  | vote | against it, I believe with perfect honesty
##  | vote | justly for our mothers, our sisters,
##  | vote | at such elections?
##  | vote | , but he had very considerable doubt whether
```

```
##  | vote | against giving women the franchise, and I
##  | vote | for it.
##  | vote | and their voice, and he was swayed
##  | vote | would be given.
##  | vote | , would not be obliged to take any
##  | vote | .
##  | vote | upon great and complex questions.
##  | vote | to a woman, who might not desire
##  | vote | .
##  | vote | at municipal and school-board elections, they ought
##  | vote | at school-board elections.
##  | vote | .
##  | vote | , with one exception, and that was
##  | vote | to a married woman.
##  | vote | .
##  | vote | to women would to a certain extent make
##  | vote | .
##  | vote | to women.
##  | vote | for Members of that House.
##  | vote | .
##  | vote | , preferred the latter.
##  | vote | , was not satisfactory.
##  | vote | .
##  | vote | , feeling, and influence upon many questions
##  | vote | .
##  | vote | for the repeal of the Bill.
```

```
##  | vote | as much as every man.
##  | vote | would not be followed by their admission to
##  | vote | if the law were fairly construed.
##  | vote | .
##  | vote | was given, and women availed themselves of
##  | vote | if she have the qualification.
##  | vote | , I should not ask her whether she
##  | vote | .
##  | vote | if she is a householder seems to me
##  | vote | - a meeting is called for the electors
##  | vote | in municipal affairs; and I feel certain
##  | vote | by proxy or by voting papers, without
##  | vote | for it, and I understand that he
##  | vote | in the municipalities.
##  | vote | in the election of school boards; and
##  | vote | is one that cannot be seriously entertained by
##  | vote | at Parliamentary Elections; and, as I
##  | vote | .
##  | vote | ; but I know there are a great
##  | vote | .
##  | vote | in favour of this measure, we now
##  | vote | , and put them out of the turmoil
##  | vote | to a certain number of well-to-do females.
##  | vote | to a class of women who are,
##  | vote | to the best class of women, you
##  | vote | given to women of this kind would work
```

```
##  | vote | , and gives it to those who are
##  | vote | , but are not to sit as representatives
##  | vote | has been already given to women in many
##  | vote | , and I do not know that they
##  | vote | is a small thing, the granting of
##  | vote | to women.
##  | vote | , we should open the door to all
##  | vote | without the least difficulty; and I never
##  | vote | simply because she happened to be a woman
##  | vote | , but those words have been deleted.
##  | vote | at elections for Members of this House of
##  | vote | .
##  | vote | , in order that they may not be
##  | vote | for it; for I think we may
##  | vote | ; but that any unmarried lady who drives
##  | vote | in the election of members of school boards
##  | vote | , or else it is that their admission
##  | vote | to women-householders.
##  | vote | to all but the possessors of £ 5
##  | vote | , and I read in The Women's Suffrage
##  | vote | , the House would in consequence provide work
##  | vote | and deprive her of it directly
##  | vote | and refuse it to the woman?
##  | vote | to a man must apply to a woman
##  | vote | , politics may become a subject of domestic
##  | vote | you will at once convert her into a
```

```
##  | vote | for it.
##  | vote | from those who desire it; and still
##  | vote | you can develop their interests in the well-being
##  | vote | .
##  | vote | he has?
##  | vote | , why should not every adult also claim
##  | vote | , and if they once admitted that a
##  | vote | was right, how could they prevent women
##  | vote | in municipal, parochial, and school board
##  | vote | , I should be obliged to admit that
##  | vote | ; but I do not base my argument
##  | vote | at municipal, parochial, and school-board elections
##  | vote | in certain elections should be disabled from voting
##  | vote | and giving them the Parliamentary franchise, because
##  | vote | in respect of their property, I say
##  | vote | of the landowner is swamped by the votes
##  | vote | against it.
##  | vote |
##  | vote | .
##  | vote | because she cannot become a policeman.
##  | vote | on municipal questions; why, then,
##  | vote | gives.
##  | vote | .
##  | vote | for it, let us know that it
##  | vote | to a woman, because she is a
##  | vote | by the disqualification of a particular farm,
```

```
##  | vote | to a married woman.""
##  | vote | , but could not - or, it
##  | vote | , they ought to claim the right to
##  | vote |
##  | vote | in support of the proposal; but as
##  | vote | - and in some cases, such as
##  | vote | ; and to show the equity of the
##  | vote | , she may associate herself with others in
##  | vote | .
##  | vote | at their meetings or take a part in
##  | vote | , whether they are married or unmarried,
##  | vote | is a remedy for"" All the
##  | vote | because they are not educated.
##  | vote | for the endowment of some particular sect,
##  | vote | .
##  | vote | just as quietly as if she were in
##  | vote | , as to upset the foundations of society
##  | vote | once every four or five years at the
##  | vote | .
##  | vote | .
##  | vote | , and every married woman who holds leasehold
##  | vote | .
##  | vote | would be limited, no doubt; but
##  | vote | .
##  | vote | in the municipal elections was asked, it
##  | vote | , and yet the whole Resolution now before
```

```
##  | vote | , and thereupon there was a demonstration made
##  | vote | for Members of Parliament on the same conditions
##  | vote | .
##  | vote | ?
##  | vote | to married women, but only to widows
##  | vote | , but you take away a vote from
##  | vote | from her, and you have to tell
##  | vote | , and you will have that position which
##  | vote | , because you will become so changed that
##  | vote | to the woman who has had the advantage
##  | vote | to the woman who has not had those
##  | vote | , when a woman has proved herself more
##  | vote | on the floor of this House in favour
##  | vote | , if I had not at the same
##  | vote | to every woman, married or unmarried,
##  | vote | .
##  | vote | of the women of England they would not
##  | vote | with all its burdens, with all its
##  | vote | should be given to women, and they
##  | vote | , because they would always give a vote
##  | vote | in favour of peace.
##  | vote | , it would be a vote in favour
##  | vote | in favour of war rather than peace.
##  | vote | , on the ground on which I agree
##  | vote | , if my hon.
##  | Vote | of this kind could possibly be maintained?
```

```
##  | Vote | by the amount of £ 100, the
```

Despite its frequent appearance, the triple "woman have vote" does suggest that in the 1870s women were granted this right. Women were explicitly banned from voting in Great Britain during the Reform Act 1832 and the Municipal Corporations Act 1835, and were not granted the right to vote on the same terms as men (over the age of 21) until the Representation of the People Act of 1928.

However, women's right to vote became a national movement in the Victorian era. A combination of triples analaysis and KWIC gives us insight into the conversations surrounding women.

It is easy to explore the top 20 or even top 100 triples instead. Moving between the types and number of triples displayed can give us a broader or more focused insight into our corpus.

```
top_20_triples_woman_as_subject <- triples_with_woman_as_subject %>%
  count(full_triple) %>%
  arrange(desc(n)) %>%
  slice(1:20)


top_20_triples_woman_as_subject %>%
  kbl() %>%
  kable_material(c("striped", "hover"))
```

```
top_100_triples_woman_as_subject <- triples_with_woman_as_subject %>%
  count(full_triple) %>%
  arrange(desc(n)) %>%
  slice(1:100)


top_100_triples_woman_as_subject %>%
  kbl() %>%
  kable_material(c("striped", "hover"))
```

full_triple	n
woman have vote	48
woman have right	29
woman be able	16
woman be interested	15
woman be more	14
woman be so	14
woman exercise franchise	14
woman have property	14
woman take part	11
woman be subject	10
woman be well	10
woman be in which	9
woman be now	9
woman have child	9
woman be equal	8
woman be qualified	8
woman be unfit	8
woman employ in which	8
woman have franchise	8
woman have interest	8

Exercises

By giving analysts insight into the relationship between words in a sentence, part-of-speech extraction has potentially meaningful implications for the study of history and our understanding of the social and legal imagination more generally. Extracted triples can show us the dynamics between subjects and objects–the insight needed to understand who or what is described as having agency or as being acted upon. Extracted subject-verb pairs give us a lens to focus on the actions of subjects, and extracted adjective-noun pairs can show us the way in which nouns are characterized. To strengthen your understanding of how part-of-speech extraction gives us insight into a corpus, do the following exercises:

1) Alter the code above to see the adjectives that modify the nouns "man" and "men" for the year 1879. Do you expect the adjectives to be similar? How might the different adjectives used to modify gendered words reflect the way in which Parliamentarians imagined social order?

2) We examined gendered adjective-noun pairs for the year 1879. Try changing the year to 1830 instead. How does the way in which Parliamentarians imagine men and women change over time?

3) Part-of-speech analysis can give us insight into more dimension of a corpus than gender. Filter the debate text for 1870 for the word "future." Can the analysis of the adjectives that modify the noun "future" give us insight into the way in which the future is imagined? Do this exercise again but for the decade 1850. Now explore how William Gladstone describes the future in 1850 versus Benjamin Disraeli.

4) In our triples analysis example we filter on the subject and objects. Try instead filtering for triples that contain the verb ENTER and visualize the results in a table.

Hint: We use the carrot ^ symbol to return items that start with a word and the $ symbol to return

full_triple	n
woman have vote	48
woman have right	29
woman be able	16
woman be interested	15
woman be more	14
woman be so	14
woman exercise franchise	14
woman have property	14
woman take part	11
woman be subject	10
woman be well	10
woman be in which	9
woman be now	9
woman have child	9
woman be equal	8
woman be qualified	8
woman be unfit	8
woman employ in which	8
woman have franchise	8
woman have interest	8
woman pay taxis	8
woman be in position	7
woman be liable	7
woman be ratepayer	7
woman be woman	7
woman employ in factory	7
woman have voice	7
woman pass examination	7
woman perform duty	7
woman take interest	7
woman vote at election	7
woman be even	6
woman be in favour	6
woman desire it	6
woman earn livelihood	6
woman not have vote	6
woman be as	5
woman be capable	5
woman be competent	5
woman be farmer	5
woman be ill	5
woman do work	5
woman entitle to vote	5
woman have now	5
woman have power	5
woman have share	5

items that end with a word. To search for triples that contain a verb you will need to search for the word as it exists between two hyphens.

5) In the previous chapter we introduced methods of measuring distinctiveness in a corpus using TF-IDF and JSD. Instead of counting top triples, use your knowledge of the application of distinctiveness measurements to determine which triples are distinctive of one year but not another, and vice-versa (e.g. which triples are distinctive of 1879 versus 1878). Then load the speaker metadata and explore which triples are distinctive across speakers (e.g. which triples are distinctive of William Gladstone versus Benjamin Disraeli).

BIBLIOGRAPHY

[1] T. R. Tangherlini, V. Roychowdhury, B. Glenn, C. M. Crespi, R. Bandari, A. Wadia, M. Falahi, E. Ebrahimzadeh, and R. Bastani, ""Mommy Blogs" and the Vaccination Exemption Narrative: Results From A Machine-Learning Approach for Story Aggregation on Parenting Social Media Sites," *JMIR Public Health and Surveillance*, vol. 2, no. 2, p. e166, Nov. 2016. [Online]. Available: http://publichealth.jmir.org/2016/2/e166/ 3, 9

[2] J. Guldi, "Critical Search: A Procedure for Guided Reading in Large-Scale Textual Corpora," p. 35. 3

[3] R. Cordell and D. Smith, "Viral texts: Mapping networks of reprinting in 19th-century newspapers and magazines," 2022. [Online]. Available: http://viraltexts.org 3

[4] D. Papadopoulos, N. Papadakis, and A. Litke, "A Methodology for Open Information Extraction and Representation from Large Scientific Corpora: The CORD-19 Data Exploration Use Case," *Applied Sciences*, vol. 10, no. 16, p. 5630, Aug. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/16/5630 5, 9, 10

[5] R. Glauber and D. Barreiro Claro, "A systematic mapping study on open information extraction," *Expert Systems with Applications*, vol. 112, pp. 372–387, Dec. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417418303932 9

[6] G. Angeli, M. J. Johnson Premkumar, and C. D. Manning, "Leveraging Linguistic Structure For Open Domain Information Extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, 2015, pp. 344–354. [Online]. Available: http://aclweb.org/anthology/P15-1034 9, 12

[7] M. Riaz and R. Girju, "Recognizing Causality in Verb-Noun Pairs via Noun and Verb Semantics," in *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*. Gothenburg, Sweden: Association for Computational Linguistics, 2014, pp. 48–57. [Online]. Available: http://aclweb.org/anthology/W14-0707 9

[8] S. Zhao, T. Liu, S. Zhao, Y. Chen, and J.-Y. Nie, "Event causality extraction based on connectives analysis," *Neurocomputing*, vol. 173, pp. 1943–1950, Jan. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231215013995 9, 10

[9] C. S. G. Khoo, S. Chan, and Y. Niu, "Extracting causal knowledge from a medical database using graphical patterns," in *Proceedings of the 38th Annual Meeting on*

Association for Computational Linguistics - ACL '00. Hong Kong: Association for Computational Linguistics, 2000, pp. 336–343. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1075218.1075261 9

[10] R. Girju, "Automatic detection of causal relations for Question Answering," in *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering* -, vol. 12. Not Known: Association for Computational Linguistics, 2003, pp. 76–83. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1119312.1119322 9

[11] B. Ceran, N. Kedia, S. R. Corman, and H. Davulcu, "Story Detection Using Generalized Concepts and Relations," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. Paris France: ACM, Aug. 2015, pp. 942–949. [Online]. Available: https://dl.acm.org/doi/10.1145/2808797.2809312 9

[12] S. Alashri, J.-Y. Tsai, A. R. Koppela, and H. Davulcu, "Snowball: Extracting Causal Chains from Climate Change Text Corpora," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*. South Padre Island, TX, USA: IEEE, Apr. 2018, pp. 234–241. [Online]. Available: https://ieeexplore.ieee.org/document/8367769/ 9

[13] J. Guldi, *The Dangerous Art of Text Mining*. Forthcoming, 2023. 9

[14] G. Stanovsky and I. Dagan, "Creating a Large Benchmark for Open Information Extraction," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016, pp. 2300–2305. [Online]. Available: http://aclweb.org/anthology/D16-1252 12

[15] P. Gamallo, M. Garcia, and S. Fernandez-Lanza, "Dependency-Based Open Information Extraction," p. 9. 13

[16] L. Del Corro and R. Gemulla, "ClausIE: clause-based open information extraction," in *Proceedings of the 22nd international conference on World Wide Web - WWW '13*. Rio de Janeiro, Brazil: ACM Press, 2013, pp. 355–366. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2488388.2488420 13

[17] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, "Open information extraction from the web," *Communications of the ACM*, vol. 51, no. 12, pp. 68–74, Dec. 2008. [Online]. Available: https://dl.acm.org/doi/10.1145/1409360.1409378 13

[18] Y. Shinyama and S. Sekine, "Preemptive information extraction using unrestricted relation discovery," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* -. New York, New York: Association for Computational Linguistics, 2006, pp. 304–311. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1220835.1220874 13

[19] F. Wu and D. S. Weld, "Open Information Extraction Using Wikipedia," p. 10. 13

[20] M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "Open Language Learning for Information Extraction," p. 12. 14

[21] L. Karttunen, "From Natural Logic to Natural Reasoning," in *Computational Linguistics and Intelligent Text Processing*, A. Gelbukh, Ed. Cham: Springer International Publishing, 2015, vol. 9041, pp. 295–309, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-319-18111-0_23 14

[22] S. Buongiorno and O. A. Cerpa, "posextract," https://github.com/charlespwd/project-title, 2022. 20, 42

[23] AJMC, "A timeline of covid-19 developments in 2020." [Online]. Available: https://www.ajmc.com/view/a-timeline-of-covid19-developments-in-2020 38

[24] S. Cato and A. Inoue, "Libertarian approaches to the COVID-19 pandemic," *Bioethics*, p. bioe.13007, Feb. 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/bioe.13007 38

[25] S. L. Pink, J. Chu, J. N. Druckman, D. G. Rand, and R. Willer, "Elite party cues increase vaccination intentions among Republicans," *Proceedings of the National Academy of Sciences*, vol. 118, no. 32, p. e2106559118, Aug. 2021. [Online]. Available: http://www.pnas.org/lookup/doi/10.1073/pnas.2106559118 38

[26] H. O.-Y. Li, A. Bailey, D. Huynh, and J. Chan, "YouTube as a source of information on COVID-19: a pandemic of misinformation?" *BMJ Global Health*, vol. 5, no. 5, p. e002604, May 2020. [Online]. Available: https://gh.bmj.com/lookup/doi/10.1136/bmjgh-2020-002604 38

[27] B. Heath, "Americans divided on party lines over risk from coronavirus: Reuters/ipsos poll," Mar 2020. [Online]. Available: https://www.reuters.com/article/us-health-coronavirus-usa-polarization/americans-divided-on-party-lines-over-risk-from-coronavirus-reuters-ipsos-poll-idUSKBN20T2O3 38

[28] M. Motta, D. Stecula, and C. Farhart, "How Right-Leaning Media Coverage of COVID-19 Facilitated the Spread of Misinformation in the Early Stages of the Pandemic in the U.S." *Canadian Journal of Political Science*, vol. 53, no. 2, pp. 335–342, Jun. 2020. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0008423920000396/type/journal_article 38

[29] H. Chu, J. Z. Yang, and S. Liu, "Not My Pandemic: Solution Aversion and the Polarized Public Perception of COVID-19," *Science Communication*, vol. 43, no. 4, pp. 508–528, Aug. 2021. [Online]. Available: http://journals.sagepub.com/doi/10.1177/10755470211022020 38

[30] P. S. Hart, S. Chinn, and S. Soroka, "Politicization and Polarization in COVID-19 News Coverage," *Science Communication*, vol. 42, no. 5, pp. 679–697, Oct. 2020. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1075547020950735 39

[31] S. E. Gollust, R. H. Nagler, and E. F. Fowler, "The Emergence of COVID-19 in the US: A Public Health and Political Communication Crisis," *Journal of Health Politics, Policy and Law*, vol. 45, no. 6, pp. 967–981, Dec. 2020. [Online]. Available: https://read.dukeupress.edu/jhppl/article/45/6/967/165291/The-Emergence-of-COVID19-in-the-US-A-Public-Health 39

[32] C. N. . C. M, "Trump rallies his base to treat coronavirus as a 'hoax'." [Online]. Available: https://www.politico.com/news/2020/02/28/trump-south-carolina-rally-coronavirus-118269 39

[33] S. T. Ahmed, "Managing News Overload (MNO): The COVID-19 Infodemic," *Information*, vol. 11, no. 8, p. 375, Jul. 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/8/375 39

[34] WHO, "Infodemic," https://www.who.int/health-topics/infodemic#tab=tab_1. 40

[35] A. Gollwitzer, C. Martel, W. J. Brady, P. Pärnamets, I. G. Freedman, E. D. Knowles, and J. J. Van Bavel, "Partisan differences in physical distancing are linked to health outcomes during the COVID-19 pandemic," *Nature Human Behaviour*, vol. 4, no. 11, pp. 1186–1197, Nov. 2020. [Online]. Available: http://www.nature.com/articles/s41562-020-00977-7 40

[36] E. Zhao, Q. Wu, E. M. Crimmins, and J. A. Ailshire, "Media trust and infection mitigating behaviours during the COVID-19 pandemic in the USA," *BMJ Global Health*, vol. 5, no. 10, p. e003323, Oct. 2020. [Online]. Available: https://gh.bmj.com/lookup/doi/10.1136/bmjgh-2020-003323 40

[37] A. T. Brian Kennedy and C. Funk, "Americans' trust in scientists, other groups declines," 2021. [Online]. Available: https://www.pewresearch.org/science/2022/02/15/americans-trust-in-scientists-other-groups-declines/ 40, 41, 51

[38] M. Honnibal, I. Montani, S. Van Landeghem, and B. Adriane, "spacy: Industrial-strength natural language processing in python," 2020. [Online]. Available: https://github.com/explosion/spaCy 40

[39] E. Ash, S. Galletta, D. Hangartner, Y. Margalit, and M. Pinna, "The Effect of Fox News on Health Behavior during COVID-19," p. 55. 41

[40] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text." in *ICWSM*, E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, and A. H. Oh, Eds. The AAAI Press, 2014. [Online]. Available: http://dblp.uni-trier.de/db/conf/icwsm/icwsm2014.html#HuttoG14 49

[41] A. S. E. . W. M. Jurkowitz M., Mitchell, "U.s. media polarization and the 2020 election: A nation divided," 2020. [Online]. Available: https://www.pewresearch.org/journalism/2020/01/24/u-s-media-polarization-and-the-2020-election-a-nation-divided/ 51

[42] K. Ognyanova, D. Lazer, M. A. Baum, J. Druckman, J. Green, R. H. Perlis, M. Santillana, J. Lin, M. Simonson, and A. Uslu, "THE COVID STATES PROJECT: A 50-STATE COVID-19 SURVEY REPORT #60: VACCINE MISINFORMATION, FROM UNCERTAINTY TO RESISTANCE," p. 16. 51

[43] M. L. Jockers and R. Thalken, *Text analysis with R: For students of literature.* Springer Nature, 2014. 57

[44] T. Arnold and L. Tilton, *Exploring Humanities Data in R: Exploring Networks, Geospatial Data, Images and Texts.* Springer, 2015. 58

[45] J. Silge and D. Robinson, *TidyText Mining with R.* O'Reilly, 2017. 59

[46] A. Crymble, *Technology and the Historian.* University of Illinois Press, 2021. 59

[47] G. Ignatow and R. Mihalcea, *Text Mining: A Guidebook for the Social Sciences.* Sage, 2017. 60

[48] X. lin Shu, *Knowledge Discovery in the Social Sciences A Data Mining Approach.* University of California Press, 2020. 61

[49] D. Sarkar, *Text Analytics with Python: A Practitioner's Guide to Natural Language Processing.* Apres / Springer, 2019. 62

[50] S. Burns, *Natural Language Processing: A Quick Introduction to NLP with Python and NLTK.* Amazon KDP, 2019. 62

[51] E. K. Steven Bird and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O'Reilly, 2009. 63

[52] Y. Vasiliev, *Natural Language Processing with Python and spaCy: A Practical Introduction.* No Starch Press, 2021. 64

[53] I. Milligan, *History in the Age of Abundance?: How the Web Is Transforming Historical Research.* No Starch Press, 2019.

[54] S. Buongiorno, R. Kalescky, E. Godat, O. A. Cerpa, and J. Guldi, "The hansard 19th-century british parliamentary debates with improved speaker names: Parsed debates, n-gram counts, special vocabulary, collocates, and topics," 2022.

[55] S. Buongiorno, O. A. Cerpa, and J. Guldi, "The hansard 19th-century british parliamentary debates with improved speaker names: Speaker metadata," 2022.

[56] S. Zhao, Q. Wang, S. Massung, B. Qin, T. Liu, B. Wang, and C. Zhai, "Constructing and Embedding Abstract Event Causality Networks from Text Snippets," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining - WSDM '17*. Cambridge, United Kingdom: ACM Press, 2017, pp. 335–344. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3018661.3018707

[57] B. MacCartney and C. D. Manning, "An extended model of natural logic," in *Proceedings of the Eighth International Conference on Computational Semantics - IWCS-8 '09*. Tilburg, The Netherlands: Association for Computational Linguistics, 2009, p. 140. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1693756.1693772

[58] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit." in *ACL (System Demonstrations)*. The Association for Computer Linguistics, 2014, pp. 55–60. [Online]. Available: http://dblp.uni-trier.de/db/conf/acl/acl2014-d.html#ManningSBFBM14

[59] J. B. Ruiz and R. A. Bell, "Predictors of intention to vaccinate against COVID-19: Results of a nationwide survey," *Vaccine*, vol. 39, no. 7, pp. 1080–1086, Feb. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0264410X21000141

[60] K. Hall Jamieson and D. Albarracín, "The Relation between Media Consumption and Misinformation at the Outset of the SARS-CoV-2 Pandemic in the US," *Harvard Kennedy School Misinformation Review*, Apr. 2020. [Online]. Available: https://misinforeview.hks.harvard.edu/article/the-relation-between-media-consumption-and-misinformation-at-the-outset-of-the-sars-cov-2-panden

[61] A. Lupia, "Communicating science in politicized environments," *Proceedings of the National Academy of Sciences*, vol. 110, no. supplement_3, pp. 14 048–14 054, Aug. 2013. [Online]. Available: https://pnas.org/doi/full/10.1073/pnas.1212726110

[62] A. B. Bayram and T. Shields, "Who Trusts the WHO? Heuristics and Americans' Trust in the World Health Organization During the COVID-19 Pandemic," *Social Science Quarterly*, vol. 102, no. 5, pp. 2312–2330, Sep. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/ssqu.12977

[63] H. Houser, *Infowelm*. Cambridge University Press, 2020.