Chapter 7: Decoding Grammar to Uncover the Language of Agency

Language change does not only happen at the level of individual words or phrases. It also takes place in the structure of sentences. By examining patterns in sentence grammar, analysts can uncover deeper shifts in how speakers express ideas and reflect social norms over time (Zeldes 2015). In this chapter we show that the power dynamics imagined by a culture can be intuited from grammatical relationships within sentences. For example, subject-object relations–that is, relations that specify who performs actions (subjects) and who is acted upon (objects)–are not encoded at the level of individual words, but rather emerge from sentence structure, specifically from the syntactic relations between words (Johnson 1987). Digital humanists and social scientists have long analyzed langauge to understand political agency and storytelling (Franzosi 2004, Tangherlini et al 2020, Shahsavari et al. 2020). Their work shows us that subject-object relations are especially revealing because they encode underlying cultural assumptions about agency, identity, and power dynamics as expressed by a community through language.

In this chapter, we focus on how gendered power relations are constructed and reflected through syntactic structures, particularly in the ways language encodes positionality, that is, the context that shapes a person's access to–or lack of–power in a society. We are not the first to recognize that grammar, and syntax in particular, plays a foundational role in shaping the expression of gender, and can serve to empower, constrain, naturalize, or delegitimize people. The basis of Feminist theory is in part built upon the idea that the structures of language both reveal and reinforce gendered expressions of power, making syntax a critical site for analyzing power dynamics as produced through discourse.

Foundational feminist theorists including Hélène Cixous, Luce Irigaray, and Monique Wittig each approach the politics of syntax from distinct angles. Cixous critiques conventional syntactic forms—such as subject-verb-object constructions—as mirroring the rigid hierarchies of patriarchal political order (1976). She calls for forms of writing that defy linear structure and fixed meaning, arguing that fluid, disruptive syntax resists systems in which agency is centralized in a dominant (often male) subject and others are relegated to passive roles. Luce Irigaray, in contrast, interrogates the deeper structural logic of discourse itself (1985). She argues that fundamentally masculine forms of discourse structure what we might call the "syntax" of social and political systems – even structuring thinking itself. Systems of discourse reproduce the male subject as the universal norm and language and politics will remain systems of masculine self-reproduction. Monique Wittig argues that language doesn't just reflect gendered hierarchies—it actively produces them (1992). For Wittig, the very category of syntax is a political fiction sustained through language that encode heterosexual norms. Rather than seek a "feminine" alternative to syntax, Wittig calls for the

destruction of the linguistic structures that make gender seem natural. Language, she insists, must be politically rewritten to dismantle the binary logic on which patriarchy depends.

Feminist theories of syntax offer a powerful lens through which to understand the stakes of grammatical part-of-speech analysis in historical texts. The Nineteenth-century Hansard represents an enormous opportunity in this regard; as a collection of political debates between men, Hansard has been minimally engaged as a resource for understanding how gender norms were perpetuated. Yet as feminist historians have long understood, parliament played a massive role in structuring women's experience, from the Laws of Coverture to the Matrimonial Clauses Act (1857), the Married Women's Property Acts (1870-1882), the Contagious Disease Acts (1864-69) and women's right to vote in the UK (1918-28) (Rackley and Auchmuty 2019, Kent 1999, Griffin 2012). Feminist historians have shown how, in the debates around the Contagious Diseases Acts, women were portrayed as both powerless and degraded, yet also as potentially threatening (Walkowitz, 1970, p. 4).

Members of parliament in the nineteenth century regularly described a world where male subjects acted and female objects were acted upon. Cixous's critique of rigid syntactic forms specifically targets the kinds of grammatical regularities we observe in mid-nineteenth-century Parliamentary discourse. Her call for a syntax that resists fixed meaning aligns with our aim to expose how stable grammatical roles naturalize uneven distributions of agency. Similarly, Irigaray argues that speech acts that assert a masculine subject as the universal norm represent a "political syntax" that reproduces political violence. Through a feminist lens, we can see that Hansard reveals an erasure of women as political actors. Wittig's insistence that language produces the fiction of gender further illuminates how syntactic roles in Hansard reinforce the category of "woman" as passive, pathological, and other.

Using text mining, we have an opportunity to amplify and extend the study of gender dynamics on a wider scale. By examining the rhetoric used across time and debates in parliament, we can trace episodes where attitudes towards gender were shifting – as well as moments when prejudices about women were reinforced. We can identify the work of parliament as a collective in perpetuating or challenging gender norms.

By foregrounding grammatical part-of-speech extraction rather than word counts alone, we bring into focus how syntax structures engendered power, echoing the concerns of these feminist theorists. For example, in Parliamentary debates on the Contagious Diseases Acts, women are rarely subjects of political action. They are objects—spoken of, regulated, and scrutinized—but almost never granted the syntactic position of agent. This syntactic pattern, we argue, does not merely reflect political ideology; it is one of the mechanisms by which that ideology is naturalized in discourse.

This chapter delves into grammatical part-of-speech extraction for analyzing gender and agency by first highlighting the distinction between lexical analysis–the study of words and their meanings– and syntactic analysis, which focuses on the relationships between words and how they form larger structures, such as sentences. Lexical analysis involves categorizing words (e.g., verbs, nouns, adjectives), whereas syntactic analysis investigates how these words interact and connect to convey meaning. By exploring these interactions, researchers performing syntactic analysis can glean insights that lexical analysis alone may not reveal. For instance, syntactic analysis allows us to

compare how different groups of people are described as subjects (those performing actions) versus objects (those affected by actions) in texts over time.

As we begin to explore the usefulness of syntactic analysis for understanding historical text, we circle back to the Hansard corpus. The mid-nineteenth century, marked by the passage of the Contagious Diseases Acts, provides a distinct historical context for examining how the language of agency regarding gender are embedded within syntactic structures.

The Study of Power in Syntax

When we speak of "syntax" as having a role in perpetuating political points of view, we literally mean that the noun-verb relationships in sentences regularly embody power relationships. To demonstrate this concept, consider a passage from George Eliot's **Daniel Deronda** (1876), a novel contemporary to the Hansard corpus that depicts the masculine gaze with exemplary specificity. The narrator describes the female protagonist, Gwendolyn, stating: "She was the central object of that pretty picture, and everyone present must gaze at her." Among the relationships embedded in the sentence is a statement where the object of "gaze" is "her." The female character is the passive receiver of an active gaze. In this case, the subject doing the gaze is "everyone," an ungendered world.

Feminist theorists have shown that from Victorian novels to twentieth-century cinema, the gaze was often gendered. The masculine gaze refers to a way of positioning women as objects to be looked at, evaluated, or acted upon—typically through a masculine perspective that exerts social power. The roots of this theory stem from cinema studies. Laura Mulvey famously argues in "Visual Pleasure and Narrative Cinema" (1975) that cinema positions the viewer as male and women as passive objects of visual pleasure. However, Mulvey's argument has since been widely extended to the study of literature, history, and language in which the male subject functions as a mechanism of visual and narrative control, framing women as passive subjects of observation rather than active agents (Mulvey 1975). Here we will demonstrate that a similar pattern can be identified through syntactic analysis.

With text mining, we have the opportunity to test whether the subjects who gaze are always gendered male and the subjects who are gazed at are always gendered female. We can ask a computer to extract every instance of the word "gaze" and the pronouns associated with it, even when it is presented in different grammatical forms, such as "will gaze at him/her," or "gazing at him/her." Next, we can quantify the number of times the word "gaze" was applied to "her" versus to the number of times "gaze" was applied to "him." Part-of-speech extraction thus allows the analyst to aggregate a great deal of information about any given verb: when it arises, who uses it, and what other words are used in a similar context. The same technique can be applied to all verbs in a given corpus, and so give a portrait of who is doing the action and who receives the action in the minds of people associated with an institution.

In digital literary studies, part-of-speech (PoS) extraction has been widely used to analyze gender within text, for instance, by allowing scholars to count the number of times female and male charac-

ters are referred to, either by name or by pronoun (Chen 2020, Kraicer and Piper 2018, Underwood et. al 2018). The process of breaking prose into discrete linguistic units and then using these units to guide analysis has previously been described as a form of "reductive reading" (Allison 2018). However, this characterization has primarily been applied to the bag-of-words (BoW) approach to text analysis, in which words are counted without consideration of their grammatical context or their syntactic relations (Arnold 2019). In this chapter, we seek to extend this line of inquiry by making a distinction between "part-of-speech extraction," which tags and analyzes parts-of-speech without regard to grammar, and "grammatical part-of-speech extraction," which accounts for the syntactic relationships between words, incorporating grammatical rules to determine the function of each part-of-speech within a sentence.

We suggest that extracting grammatical parts-of-speech offers a more effective framework for analyzing how language encodes gender and power than the bag-of-words (BoW) approach, which ignores linguistic context by analyzing words in isolation. While grammatical relationships have been analyzed in text, the algorithmic techniques that shape and inform interpretive analysis–particularly those related to grammar–have received comparatively little attention (Tangherlini 2016, Franzosi 2024).

For brevity, throughout this chapter we will call this method of extracting and analyzing words based on their grammatical function "grammatical POS analysis." Understanding the parts-of-speech in a sentence begins with recognizing that words take on different roles depending on their relationships to one another. For instance, "women" and "themselves" are both nouns, but they serve different purposes in a sentence. The noun "women" performs an action, while the noun "themselves" receives an action. Recognizing the distinctive roles words play helps clarify how each word functions in the sentence.

In this chapter, we will introduce two approaches to perform grammatical POS analysis. First we will extract parts-of-speech using a rule-based, statistical model. After, we will use a large language model (LLM) to identify and generate parts-of-speech for analysis.

The rule-based, statistical model we will use is `spaCy`. spaCy is a natural language processing library that analyzes text using linguistic rules and pretrained models. It has been trained on annotated texts to identify patterns in language based on fixed data processing approaches. `spaCy`'s models are well-calibrated for text processing tasks, but are limited to extracting data—that is, identifying and labeling linguistic features. LLMs, on the other hand are trained on massive text corpora to predict the next word in a sequence. Instead of extracting text LLMs generate new text.

As we will demonstrate, both `spaCy` and LLMs give us a means to perform grammatical POS analysis. Using `spacyr`, we will extract syntactic relationships to analyze adjective-noun pairs, specifically the adjectives assigned to the word "woman." Following this, we will use a LLM to identify the top action statements–statements about subjects performing actions to objects–by generating subject-verb-object relationships using different prompting techniques. Both approaches will help us see different linguistic features related to gender and agency.

The `spacyr` Programming Library

To analyze the syntax within sentences, analysts typically count nouns (or other parts of speech) and the verbs, adjectives, other nouns, or prepositional phrases associated with them. Appreciating the software behind this counting mechanism can help us to appreciate the high level of accuracy that we can expect from syntactical analysis today.

A linguistic pipeline transforms raw text into structured linguistic information using pre-trained language models (LMs)—not to be confused with "large language models" (LLMs), which are designed for broad generative tasks such as open-ended question answering, text generation, and conversation. The smaller-scale LMs are trained on large collections of text (called corpora) to predict and label specific linguistic features, such as parts of speech, syntactic dependencies, and named entities.

While the concept of a linguistic pipeline has its roots in the 1990s, it gained traction with the development of the Penn Treebank (Marcus et al. 1993), which provided a large, annotated corpus of English text marked with part-of-speech tags and phrase-structure parse trees. This resource enabled researchers to train and evaluate models for individual linguistic tasks in a standardized, sequential fashion, laying the groundwork for modern pipeline architectures. The Penn Treebank was revolutionary because it enabled a shift from rule-based systems—which relied on manually crafted grammatical rules written by experts—to predictive approaches, where statistical models learn patterns directly from annotated examples. This meant that rather than defining how English syntax should work, researchers could build models that learned syntactic patterns from real-world usage. This shift was important because it made NLP systems more adaptable, where models could be trained on different linguistic variations.

By the 2010s, linguistic pipelines had become widely adopted through frameworks such as Stanford CoreNLP and spaCy (Manning et al. 2014; Honnibal and Montani 2017). Linguistic pipelines allow models to learn patterns in language—such as sentence structure, word meanings, and syntactic relationships—by analyzing contextual usage (that is, not just the isolated meaning of a word, but how its meaning and grammatical role are shaped by surrounding words in a sentence—for example, distinguishing whether run is used as a noun in "a long run" or as a verb in "they run daily"). Once trained, the models can process new text by identifying syntactic roles (e.g., subject or object), part-of-speech tags (e.g., noun, verb), and dependencies between words.

The spacyr package builds on top of the Python library spaCy. In this way, spacyr acts as a bridge: it wraps spaCy Python code and runs spaCy behind the scenes while allowing analysts to access its functions through R. This design enables analysts to take advantage of spaCy's robust linguistic tools—such as tokenization, part-of-speech tagging, and dependency parsing—without needing to write Python code or switch programming environments from R to Python. However, this means `spacyr` requires additional setup to integrate Python's `spaCy` library with R. This involves two key steps: installing `spaCy` in a compatible Python environment and configuring R to communicate effectively with Python.

`spacyr` is a natural langauge processing (NLP) library. Analysts can use `spacyr` to perform tasks

such as:

- Tokenization: Breaking text into words, punctuation, and other meaningful units.
- Part-of-speech tagging: Identifying the grammatical role of each word in a sentence (e.g., noun, verb, adjective).
- Named entity recognition (NER): Identifying proper nouns, dates, locations, and other named entities in text.
- Dependency parsing: Analyzing the grammatical relationships between words in a sentence (e.g., which word is the subject, which is the object).
- Lemmatization: Reducing words to their base forms (e.g., "running" to "run").

`spacyr` currently provides NLP support for approximately 75 languages, including Spanish and Latin. However, a major limitation of spaCy's out-of-the-box language coverage is that its pre-trained models primarily target state, majority, or colonial languages. As a result, Indigenous and low-resource community languages remain largely unsupported within the spaCy ecosystem. However, spaCy does offer components for researchers to train their own language models.

By now, it should be clear that tokenization can be performed using a range of packages, such as Silge and Robinson's `unnest_tokens()` or, as shown here, using `spacyr`. However, this is the first time we have engaged with more advanced natural language processing capabilities like named entity recognition, dependency parsing, and lemmatization. Taken together, the tools offered by `spacyr` allow analysts to move beyond bag-of-words style text analysis, where words are counted without regard to grammar, and begin extracting structured information and identifying grammatical roles embedded in Hansard. `spacyr` supports grammar-level analysis through a sequence of computational, data-processing steps known as a "linguistic pipeline."

Installing `spacyr`

To install `spacyr`, we begin by using the standard `install.packages()` function. This gives us access to the `spacy_install()` function. Running `spacy_install()` sets up the required Python environment and installs the spaCy library along with its dependencies, integrating spaCy's capabilities within the R environment.

```
# First time only -- installing spacyr on your computer
install.packages("spacyr")

library(spacyr)

# Install spaCy and the small English model
spacy_install(lang_models = "en_core_web_sm", ask = FALSE)

# Initialize spaCy
spacy_initialize(model = "en_core_web_sm")
```

By default, spacyr installs an English language model that used to tag words with their parts-of-speech, called `en_core_web_sm`. This name provides specific information about the model:

- `en`: Indicates the model is for the English language.
- `core`: Refers to a general-purpose model, suitable for a wide range of tasks.
- `web`: Specifies that the model was trained on web-based data, such as blogs, news articles, and other online text.
- `sm`: Stands for "small," meaning the model is lightweight and optimized for efficiency, prioritizing speed and lower resource usage over maximum accuracy. spaCy can also access medium and large language models, which tend to have higher accuracy but at the cost of more computational resources.

`SpaCy` supports many languages beyond English. The other LMs can be downloaded using the `spacy_download_langmodel()` function. For example, the following code demonstrates how to download and initialize a small German model, which is specifically trained on a news corpus. For a complete list of available models and their details, visit the official spaCy models page: https://spacy.io/usage/models.

```
spacy_download_langmodel("de_core_news_sm") # load the German model
```

In our analysis we will use the `en_core_web_sm` model to analyze text in the English language. Once spaCy has installed, we can initialize the model we wish to use.

```
spacy_initialize(model = "en_core_web_sm") # load the English model
```

In each future session, we will need to not only load the spacyr library, but also initialize the model like so:

```
library(spacyr) # load the library

spacy_initialize(model = "en_core_web_sm") # load the English model
```

`spacyr` in Action: Outlining the Dependency Grammar of a Single Sentence

The Politics of Parliamentary Language

To consider how syntax might work as a tool of historical analysis, let's consider a case well-known to feminist historians, the Contagious Diseases Acts, passed between 1864 and 1869.

The Acts were a series of legislative measures aimed ostensibly at mitigating the spread of venereal disease among military personnel, but in practice they disproportionately targeted working-class

women. Under The Contagious Diseases Acts, women suspected of being prostitutes in garrison towns and naval ports could be subjected to forced medical examinations and, if found to be infected, detained in "lock hospitals"–medical institutions specifically designated for the treatment of venereal diseases–until decidedly "cured". The legislation rested on hegemonic assumptions about female sexual deviance and male sexual entitlement, effectively criminalizing female bodies while exempting the men who procured their services.

In examining the discourse on women, we find that the women themselves are largely absent from the historical record; instead, the debates reveal how members of Parliament imagined and constructed the figure of women and the anxiety of prostitution through language.

Consider an amendment suggested by a member of parliament, Acton Aryton, as recorded in Hansard:

> "He proposed to amend the clause which related to women submitting themselves voluntarily to periodical medical examination by requiring them to make a declaration that they were persons which the law recognizes only to punish."

Aryton feared that allowing women to seek certificates of health from government officials would legitimize prostitution. Aryton's amendment would require that women instead formally declare that they belong to a category of criminals – "people whom the law only recognizes in order to punish." Women who carried venereal diseases would therefore be held responsible for identifying themselves as prostitutes and transgressors, submitting their bodies before the law.

Close attention to grammatical syntax further reveals the way in which Aryton imagines women as immoral agents can be abstracted from the grammatical relations in the sentence. Among the relationships embedded in this sentence, the subject "women" is performing the verb-form "submitting" and the direct object "themselves." This utterance takes the prepositional phrase "to examination" as its complement. With the help of syntactical analysis, we can simplify the statement to its main components:

```
women-submit-themselves
```

Or, if we include the dependent clause, we may render it this way:

```
women-submit-themselves-to-examination
```

Notably, the actual historical text in no way fits the simple binary "male-observer, female-observed" from the theory of the male gaze. In Ayrton's actual phrase, women occupy both the subject and object positions – grammatically. That is, women are portrayed both as potentially the agents doing action as well as the objects receiving action. The syntax specifies a legal framework that would require women to act – seeking health certificates to remedy their supposedly immoral behaviors.

Nevertheless, the remedy that Ayrton is recommending for the spread of venereal disease is one where women recognize themselves as potentially criminal and in need of surveillance. In any case, the verb "submit" captures women's inferior position in this clause – even if the sentence neglects to specify the men to which women are submitting their power. If we find several sentences in which "women submit" plays a major grammatical role in the sentence, then the analyst must conclude that the women are party to a system that requires the deferral of their agency.

What the syntax helps us to discover is the relationship between particular kinds of subjects (women) and the actions required of them (submission). By focusing on words' grammatical functions, we can identify relationships that capture linguistic nuances, even when the words do not appear next to each other in the sentence.

The specifics of what system women were submitting to, or why women must "submit themselves," an analyst may derive from close reading of the sentences involved. Word counts alone do not capture Aryton's description of the women's agency because they are not encoded in any single word, or in any finite span of words (like in the case of n-grams).

`spacyr` analyzes the syntactic structure of sentences using dependency grammar. It represents each sentence as a dependency tree, where words are connected by directed links that show their grammatical relationships, called dependencies. One word functions as the governing word (or head), which determines the grammatical type of the relationship—such whether the word is a subject, object, or modifier—while the other is the child (or dependent), whose role and interpretation are shaped by the head. The child relies on the head to anchor its grammatical function and meaning; for example, an adjective depends on the noun it modifies, and a subject depends on the verb to complete its role in the sentence.

Figure 1 shows a dependency tree generated by `spacyr`, where each word is connected to its head by a directed arc. The arcs indicate the type of grammatical relationship between the words—such as subject, object, or modifier—with the direction of the arc pointing from the head to the dependent (child).


nsubj

In this example, the head of the verb "submitting" is the subject "women." The verb "submitting" also has multiple children: "themselves," "voluntarily," and "to." The head of the adjective "medical" is the object "examination." The head can vary based on the type of phrase or clause. For instance, in noun phrases, the noun is typically the head, while in verb phrases, the verb often functions as the head. With an understanding of how `spacyr` processes linguistic data, we can now use it to parse our Hansard data and extract parts-of-speech with a more critical eye.

Counting Adjectives and Nouns Collocated With the Word "Woman"

In investigating parliamentarian's speech about women, we used a layered approach to analysis allows us to trace patterns of language use. Our aim is to explore how members of Parliament in the mid-19th century described and imagined women, and to consider how speaker's linguistic patterns connect back to broader cultural frameworks. Specifically, we will ask: How are women represented in the language of 19th-century Parliamentarians, as reflected in the 1860 Hansard debates?

We employ a series of data processing steps to reveal how different dimensions of meaning and association emerge from the data. We begin by counting the adjectives that occur in the same sentences as the nouns woman or women, using co-occurrence as a proxy for thematic or descriptive relevance. Next, we identify other nouns that frequently appear alongside the nouns woman or women within those sentences, mapping the broader semantic field in which the term operates. Building on this overview, we then narrow our focus to identifying and counting adjective–noun pairs, where woman is directly modified by an adjective.

To begin, we preprocess the text of the 1860 Hansard debates. This involves filtering the corpus to include only sentences that contain the word "woman" or "women," regardless of capitalization. Because our focus is on sentences involving references to these words, we limit the scope of the data processing by first filtering the Hansard data. This filtering step narrows the dataset considerably, allowing for a more manageable analysis using `spacyr`.

```
library(tidyverse)
library(hansardr)

data("hansard_1860")

hansard_woman_1860 <- hansard_1860 %>%
  filter(str_detect(text, regex("woman|women", ignore_case = TRUE)))
```

We then provide our filtered hansard data to the `spacy_parse()` function to tokenize just the text column. From the text column, we extract linguistic features like part-of-speech tags by specifying `dep = TRUE`. To save computational resources and focus on only the features relevant to our analysis, we then disable lemmatization (`lemma = FALSE`) and named entity recognition (`entity = FALSE`).

```
parsed_hansard_woman <- spacy_parse(hansard_woman_1860$text,
                                    dep = TRUE,
                                    lemma = FALSE,
                                    entity = FALSE)

head(parsed_hansard_woman)
```

```
##   doc_id sentence_id token_id   token  pos head_token_id   dep_rel
## 1  text1           1        1     The  DET             2       det
## 2  text1           1        2   women NOUN             6 nsubjpass
## 3  text1           1        3   alone  ADV             2    advmod
## 4  text1           1        4    were  AUX             6   auxpass
## 5  text1           1        5    well  ADV             6    advmod
## 6  text1           1        6 clothed VERB             6      ROOT
```

spacy_parse() returns a new data frame in which each row represents an individual token from the text. This data now includes the token's part-of-speech (POS) tag and syntactic dependency, which can be used for further analysis. For instance, I can filter and count to find the top adjectives stated in the Hansard sentences containing the word "woman" or "women."

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
# Filter for just words tagged by spacyr as adjectives
adjectives <- filter(parsed_hansard_woman, pos == "ADJ")

# Count occurrences of each adjective and sort in descending order
top_adjectives <- adjectives %>%
  count(token, sort = TRUE)

# View the top adjectives
head(top_adjectives, 30) %>%
  kable()
```

token	n
other	156
young	155
married	133
great	127
poor	114
old	107
many	104
such	100
same	65
own	60
more	57
large	52
public	50
last	48
noble	48
first	45
present	43
certain	42
small	40
good	38
deceased	34
guilty	31
much	30
unfortunate	29
bad	28
whole	28
able	26
second	26
little	25
necessary	25

The results of this process show that the sentences that mention "woman" or "women" from the 1860s most often also contain the word "married." The finding is hardly earth-shattering; some of these debates led up to the first Married Women's Property Act (1870), which overturned centuries of precedent that had ruled that married women could not own or inherit property. The new act allowed married women to keep wages and inherit property up to £200 – giving women a wide degree of freedom if they found themselves divorced, abandoned, or married to an abusive or otherwise incompatible husband from whom they needed to separate. The point is that the adjective "married" doesn't give us an exceptionally new information about history. We should look down the list further.

However, many of our top words are common stop words (e.g. “other,” “such,” “many,” and “same”), which can obscure more meaningful patterns when attempting to analyze the words that co-occured in the same sentence as “woman” or “women.” Removing stop words using the `tidytext` package can help uncover more insightful trends in the corpus.

```
library(tidytext)

data(stop_words)

top_adjectives_clean <- top_adjectives %>%
  anti_join(stop_words, by = c("token" = "word"))

head(top_adjectives_clean, 30) %>% kable()
```

token	n
married	133
poor	114
public	50
noble	48
deceased	34
guilty	31
unfortunate	29
bad	28
true	25
Turkish	24
common	24
proper	24
male	23
single	23
impossible	22
strong	22
female	21
fair	20
Christian	19
English	19
moral	19
separate	17
similar	17
Catholic	16
Roman	16
domestic	16

token	n
liable	16
medical	16
political	16
sufficient	16

Removing stop words reveals something interesting: several of the remaining adjectives refer to a woman's marital status, such as "single" and "unmarried." Others suggest class distinctions, including words like "poor," "noble," "common," and "domestic." Some discuss issues with inheritance, like "deceased." Others suggest debates about crime and punishment – "guilty," "bad," and "impossible." The words about which we have the most questions are "Turkish" – what were Turkish women doing in Britain in the 1860s, and why were women in particular gaining the attention of parliament?

Analyzing Nouns

We can also use part-of-speech extraction to analyze nouns. Nouns often indicate the focus of discourse, showing what ideas, objects, or individuals are central to the narrative or argument. Repeated nouns might signify motifs. In the following code, we filter for the top nouns that occur in sentences with the word "woman" or "women" (other than "woman" or "women" themselves).

```
# Filter for just nouns from this same data
nouns <- filter(parsed_hansard_woman, pos == "NOUN")

# Exclude 'woman' and 'women' from the nouns
filtered_nouns <- nouns %>%
  filter(!(token %in% c("woman", "women")))

# Count occurrences of each noun and sort in descending order
top_nouns <- filtered_nouns %>%
  count(token, sort = TRUE)

# View the top nouns
head(top_nouns, 20)
```

```
##         token   n
## 1    children 411
## 2         men 388
## 3         man 242
## 4         law 142
```

```
## 5        case 136
## 6     country 120
## 7     persons 116
## 8      number 111
## 9     husband 101
## 10      cases  96
## 11   property  93
## 12      child  89
## 13       time  86
## 14      years  85
## 15       wife  84
## 16   marriage  81
## 17 employment  77
## 18   question  66
## 19      place  62
## 20        day  61
```

The top nouns to appear in sentences containing the word "woman" or "women" are "men," followed by "children" and "man." The prevalence of these words in the same context highlights an association between women, the Victorian family structure, and motherhood. Women were often framed as caregivers and moral guardians. In the context of Hansard, this may reflect debates on women's character. For example, discussions surrounding the Contagious Diseases Acts often framed women's morality and behavior as directly impacting children and family stability.

Comparing the top nouns in sentences containing "woman" or "women" against those containing "man" or "men" reveals significant differences in how these genders were discussed in 19th-century parliamentary debates. To demonstrate this, we perform the same data processing steps but on just sentence with the words "man" or "men."

```
hansard_man_1860 <- hansard_1860 %>%
  filter(str_detect(text, regex("\\b(man|men)\\b", ignore_case = T)))

parsed_hansard_man <- spacy_parse(hansard_man_1860$text,
                                  dep = TRUE,
                                  lemma = FALSE,
                                  entity = FALSE)

nouns <- filter(parsed_hansard_man, pos == "NOUN")

# Exclude 'man' and 'men' from the nouns
filtered_nouns <- nouns %>%
  filter(!(token %in% c("man", "men")))
```

```r
# Count occurrences of each noun and sort in descending order
top_nouns <- filtered_nouns %>%
  count(token, sort = TRUE)
```

```r
library(gridExtra)

# Arrange the output tables in a grid layout
# [1:15, ] selects the first fifteen rows (and all columns) from the dataset
# tableGrob() converts each chunk into a table for plotting
# grid.arrange() places the tables side by side so we can view more at once
grid.arrange(tableGrob(top_nouns[1:15, ]),
             tableGrob(top_nouns[16:30, ]),
             tableGrob(top_nouns[31:45, ]),
             ncol = 3)
```

	token	n
1	country	3659
2	number	2704
3	time	2374
4	years	2069
5	service	1718
6	question	1698
7	year	1647
8	case	1617
9	opinion	1611
10	officers	1527
11	army	1401
12	system	1315
13	law	1304
14	class	1276
15	hon	1202

	token	n
16	part	1112
17	position	1069
18	day	1040
19	character	1039
20	body	1034
21	power	1033
22	life	1032
23	force	1031
24	people	1020
25	war	1020
26	subject	1000
27	persons	963
28	duty	948
29	place	930
30	fact	912

	token	n
31	money	909
32	course	896
33	way	871
34	property	790
35	means	775
36	order	763
37	ships	734
38	matter	727
39	respect	725
40	franchise	722
41	opinions	697
42	state	694
43	principle	665
44	right	659
45	work	654

Some of these words, including "country," "number," "time," "question," "case," "opinion," and "respect" rank among the most prominent words in parliament overall, suggesting that "man" and

"men" feature as the default personality for all discussions of politics. Collocates like "officers," "army," "position," "force," "war," and "ships" remind us that these debates transpired in an era when both the military and government bureaucracies were entirely occupied by men. Mentions of "money," "fact," and "property" remind us that in this era, capitalism and science were equally realms of male prerogative, while "franchise" reminds us that contemporary debates over the vote reflected the contestation of the vote for men.

Most of these words are easy to overinterpret. A naive analyst might classify "respect" as a signifier that men, not women, were deserving of respect, whereas the fact that the word is also one of the most prominent in Hansard overall reminds us that the language of the time often used clauses like "a debate respecting the new tax;" such respect has nothing to do with Aretha Franklin's plea for respect for women. Women as well as men were sometimes debated in terms of "rights" and "work;" it is possible to project exclusions on this list that are in no way indicated by the data, and careful analysts will guard against this mistake.

As with most processes of text mining, it is rarely the most frequent words and categories that are the most meaningful. Looking for collocates of "man" and "men," we're unlikely to make new discoveries about history; more compelling would be tracing the distinctiveness of the collocates of "man" and "men" over time, say comparing the collocates of the 1860s with those of the 1870s.

Another approach is to examine grammatical relationships or dependencies, where relationships are encoded at the level of syntax.

Extracting Adjective-Noun Dependencies

While analyzing individual parts of speech—such as nouns and adjectives—can offer insights into language patterns, a more nuanced understanding arises when we consider how these words interact within a sentence. For example, examining how nouns are modified by adjectives reveals not just what is being discussed, but how it is being described. For instance, examining the adjectives that modify nouns like "woman" or "man" can reveal perceptions embedded in text. Adjectives often convey evaluative or descriptive judgments, shaping how the nouns they modify are framed. They reflect the speaker's perspective and can influence how the audience interprets the subject. When words are analyzed in isolation, these contextual nuances are lost, leading to an interpretation that may be disconnected from the deeper meanings embedded in the text. To capture these contextual nuances more effectively, we now shift our focus to analyzing adjective-noun pairs.

The code below is used to extract adjective-noun pairs from the Hansard debates. Specifically, it focuses on finding adjectives that describe the nouns woman or women.

When we processed the Hansard text using the `spacyr` package, it automatically added a column called `doc_id`. This column assigns a unique ID to each sentence, which retains metadata on the structure of the original data for when we tokenize the text. Therefore, we can be sure that parts-of-speech (like adjectives and nouns) are only matched within the same sentence and not across unrelated sentences.

We then define a function, `get_adjective_noun_pairs()`, that extracts adjective-noun pairs from text. While the present analysis focuses on utterances containing the terms "woman" and "women," the function is designed for broader applicability. By encapsulating these steps in a reusable function we can easily use these data processing steps in a subsequent analysis of sentences involving the terms "man" and "men."

```
# Filter for adjectives and their associated nouns
# Extract adjective-noun pairs where the adjective modifies
# the noun within the same document

get_adjective_noun_pairs <- function(parsed_data, nouns) {
  adjective_noun_pairs <- parsed_data %>%

    # Process each document separately using a grouping variable
    group_by(doc_id) %>%

    # Filter for adjectives with the dependency relation
    # "amod" (adjective modifying noun)
    filter(pos == "ADJ" & dep_rel == "amod") %>%

    # Select columns for adjective token, its token ID, and the head
    # noun's token ID
    select(doc_id, token_id, token, head_token_id) %>%

    # Join to match adjectives to their corresponding nouns within
    # the same document
    inner_join(
      parsed_data %>%
        select(doc_id, token_id, token),
      by = c("doc_id", "head_token_id" = "token_id"),
      relationship = "many-to-many",
      suffix = c("_adjective", "_noun")) %>%

    # Filter for cases where the noun is specified in the 'nouns' argument
    filter(token_noun %in% nouns) %>%

    # Combine adjectives and nouns into a single string for readability
    mutate(adjective_noun_pair = paste(token_adjective, token_noun)) %>%

    # Ungroup after processing
    ungroup() %>%
    select(adjective_noun_pair)
```

```
  return(adjective_noun_pairs)}

nouns_to_filter <- c("woman", "women")
adjective_noun_pairs_woman <- get_adjective_noun_pairs(parsed_hansard_woman,
                                                       nouns_to_filter)
```

The code performs a series of steps: first it filters for adjectives with the dependency label `amod`, which stands for "adjectival modifier." Adjectival modifiers are adjectives that directly modify nouns. The code then joins each adjective with the noun it modifies. At this point, we have processed all adjective-noun pairs within these sentences. We are, however, only interested in adjective-noun pairs containing the word "woman" or "women." Therefore, we added a filtering step to only keep relevant pairs. Finally, the code combines the adjective and noun into a single string to make the results easier to read. The final result is a data frame that lists all such adjective-noun combinations found in the corpus.

```
adjective_noun_pair_counts_woman <- adjective_noun_pairs_woman %>%
  count(adjective_noun_pair, name = "count") %>%
  arrange(desc(count))
```

```
grid.arrange(tableGrob(adjective_noun_pair_counts_woman[1:15, ]),
             tableGrob(adjective_noun_pair_counts_woman[16:30, ]),
             tableGrob(adjective_noun_pair_counts_woman[31:45, ]),
             ncol = 3)
```

	adjective_noun_pair	cou	adjective_noun_pair	cou	adjective_noun_pair	cou
1	married women	87	unhappy women	71	Catholic women	2
2	young woman	58	Christian women	62	Indian women	2
3	poor woman	53	helpless women	63	Many women	2
4	old woman	39	other women	64	Presbyterian woman	2
5	married woman	35	Chinese women	45	Turkish women	2
6	young women	36	drunken woman	46	aged woman	2
7	old women	26	respectable women	47	aged women	2
8	poor women	18	Irish women	38	black woman	2
9	many women	17	bad woman	39	bodied women	2
10	unfortunate women	10	common women	130	defenceless women	2
11	unmarried women	10	few women	131	deserted women	2
12	English women	9	most women	132	disorderly women	2
13	single women	8	naked women	133	drunken women	2
14	unfortunate woman	8	single woman	134	handsome woman	2
15	unemployed women	7	Catholic woman	125	harmless women	2

It is easy to explore the top 10, 20, or even top 100 adjective-noun pairs by slicing the data and presenting the results in a table. Adjusting the number of pairs displayed allows us to shift between a broader overview and a more focused examination of the corpus.

Without considering the historical context of the extracted words, a list of top adjective-noun pairs suggests that members of parliament were particularly concerned with women in a state of distress, including the "poor," "unmarried," "unfortunate," and "unemployed," which suggests parliament's role in welfare. Other common patterns, suggest that parliament was also presiding in moral judgment, not only distinguishing the "respectable woman" who merited welfare, but also shaming the "unhappy," "helpless," "drunken," "bad," and "disorderly" women who merited criminal attention rather than welfare. That parliament had so many categories for female misbehavior suggests how much space they gave this question. The words are likely a reflection of the beliefs of members of Parliament expressed in te course of legislating the Contagious Disease Acts, where questions of sexual propriety and behavior came to the fore.

Other questions are easily raised for which the analyst needs to resort to close reading: in what context did Chinese, Indian, Irish, and Turkish women come to parliament's attention during this period? When subjects come to parliament, they have necessarily risen to the level of public concern. Whther or not the two mentions of Chinese women is indexical of a particular political

event in parliament, the story that brought these Chinese women to parliament in the first place was necessarily an episode of social history that rose above the common pulse of everyday life, and the analyst would do well to investigate this story.

Returning to a single debate helps clarify one way a Parliamentarian might use the word "poor" in relation to "woman."

```
woman_context <- hansard_woman_1860 %>%
  filter(str_detect(text, regex("Chinese wom", ignore_case = TRUE))) %>%
  select(text)
```

```
## So great has been the satisfaction of the Chinese with those regulations that
## whereas before it was impossible to obtain Chinese women and children to
## accompany Chinese men to the Colonies-whereby great evils resulted-whole
## families have since been embarked, and 300 or 400 respectable Chinese women
## have been sent to our Colonies.
##
## Mr. Austin had also succeeded last year in inducing 300 Chinese women to
## accompany the immigrants to the West Indies
##
## He also knew that there was a strong feeling in New South Wales against
## increasing the number of Chinese males in Australia, there being scarcely any
## Chinese women in the country, and the Chinese in the Colony amounted to 21,
## 000, or about one-fourth of the adult male population.
```

The speech refers to the questioning of managing British colonies in the West Indies. After the abolition of slavery in the British Empire in 1834, plantation owners in the Caribbean (notably in British Guiana, Trinidad, and Jamaica) sought replacement labor for sugar plantations. They turned to indentured labor from India, China, and other parts of Asia and Africa. Laborers were often recruited (or deceived) into contracts. The Chinese laborers were typically from southern China, especially Fujian and Guangdong.

The speech indicates a shift of policy, where empire became specifically interested in recruiting "respectable Chineese women." The mentions of "great evils" suggests administrators' hopes that these indentured women were expected to pacify a spirit of resistance amongst Chinese indentured laborers. We also read about racial tensions in New South Wales that suggest that similar policies would not be welcome in other parts of empire.

We see parliament here managing ethnic populations – and specifically women from China – as a means of providing labor to ensure the unceasing flow of commodities. Women's bodies are expected to be part of that machinery, pacifying resistance through providing outlets for the sexual frustrations of male immigrants. Whether that outlet was consensual or not, and how women felt about being used to oil the wheels of empire, we would need to seek in other sources.

The point is that, if we were to trace the mentions of the Chinese, Irish, and Turkish women in the 1860s, or the women of different ethnicities mentioned in parliament overall, we would have an index of some of the most prominent and publicly-acknowledged ways that Britain's government was using female bodies around the world – the policies they set, the utilitarian imaginings of how women's bodies could be used to make empire more productive. That source might not be all we need for a sensitive portrait of women's experience of a global empire, but text mining may nevertheless be a part of a larger process. We can use parliament as an index for important events that punctuate the flow of time over decades or centuries, and then turn to other archives to elaborate the cases as needed.

```
nouns_to_filter <- c("man", "men")
adjective_noun_pairs_man <- get_adjective_noun_pairs(parsed_hansard_man,
                                                     nouns_to_filter)

adjective_noun_pair_counts_man <- adjective_noun_pairs_man %>%
  count(adjective_noun_pair, name = "count") %>%
  arrange(desc(count))
```

```
#library(gridExtra)
#library(grid)

small <- ttheme_minimal(
  core = list(fg_params = list(cex = 0.9)),
  colhead = list(fg_params = list(cex = 0.9))
)

grid.arrange(
  tableGrob(adjective_noun_pair_counts_man[1:15, ], theme = small),
  tableGrob(adjective_noun_pair_counts_man[16:30, ], theme = small),
  tableGrob(adjective_noun_pair_counts_man[31:45, ], theme = small),
  ncol = 3
)
```

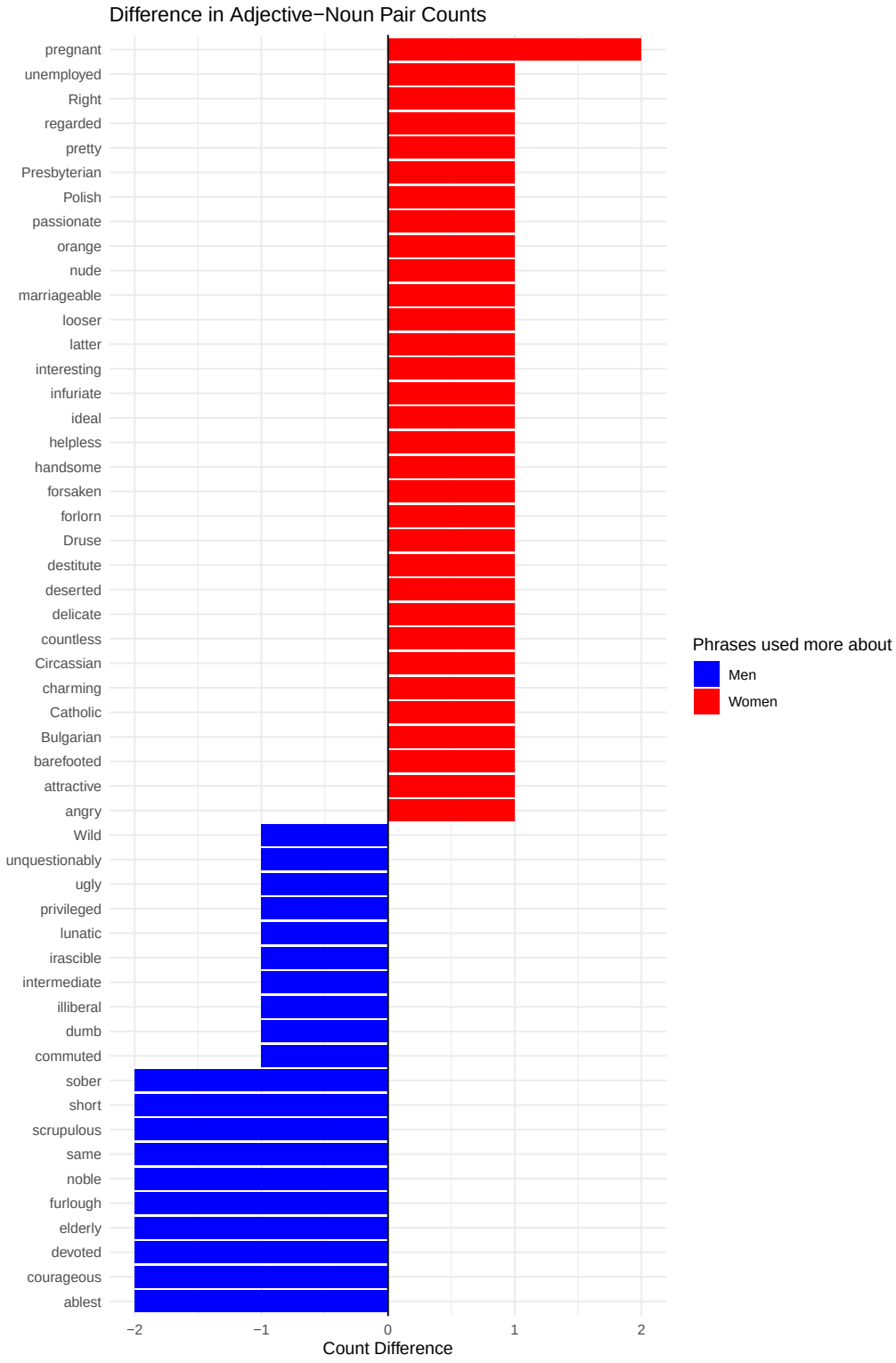
	adjective_noun_pair	count	adjective_noun_pair	count	adjective_noun_pair	cou
1	young men	8731	professional men	1431	honest men	87
2	poor man	5742	rich man	1432	commercial men	84
3	many men	4273	great man	1373	public man	83
4	such men	3194	practical men	1314	eminent man	78
5	young man	3065	distinguished men	1305	honourable men	74
6	eminent men	2756	poor men	1286	distinguished man	72
7	medical men	2517	good men	1257	unfortunate man	70
8	other men	2128	medical man	1168	naval men	64
9	public men	2119	honest man	1139	intelligent men	63
10	military men	2031	single man	1091	competent men	60
11	best men	1801	military man	1031	old men	60
12	able men	1761	few men	1012	practical man	60
13	last man	1741	other man	1013	armed men	57
14	scientific men	1511	able man	961	more men	54
15	great men	1441	old man	891	very men	54

To compare the words used of each gender, the appropriate next step is to make a list of just the adjectives and to subtract the counts for one gender from the counts for the other. This step, known as "vector subtraction," will reveal the words that are most distinctive of each gender.

```
# Vector subtraction: words applied to men vs words applied to women

# extract just the adjectives used to describe men
man_adjectives <- adjective_noun_pair_counts_man %>%
  separate(adjective_noun_pair, into = c("adjective", "noun"), sep = " ") %>% # split into two
  count(adjective, name = "count")

# extract just the adjectives used to describe women
woman_adjectives <- adjective_noun_pair_counts_woman %>%
    separate(adjective_noun_pair, into = c("adjective", "noun"), sep = " ") %>% # split into t
  count(adjective, name = "count")

# join the two into one database and subtract the count of men from the count of women.
comparison <- full_join(man_adjectives, woman_adjectives, by = "adjective",
```

```
                        suffix = c("_1", "_2")) %>% #join
  mutate(count_1 = replace_na(count_1, 0),
         count_2 = replace_na(count_2, 0),
         count_diff = count_2 - count_1) %>% # subtract men's count from women's
  filter(count_diff!=0) # delete all entries where there is no difference

# In this dataset, there are a large number of adjectives with a count of two
# We will therefore "sample" from the results to draw our visualization
# This means each regeneration of the visualization will produce different results.

# take the top adjectives unique to women
positive_sample <- comparison %>%
  filter(count_diff > 0) %>% # must show a difference
  arrange(desc(count_diff)) %>%
  top_n(20) %>%
  mutate(Category = "Women")

# take the top adjectives unique to men
# because there are so many with a max of 2, sample some positives
negative_sample <- comparison %>%
  filter(count_diff < 0) %>% # must show a difference
  sample_n(20) %>% # random sample
  mutate(Category = "Men")

filtered_df <- negative_sample %>%
  arrange(desc(count_diff)) %>%
  bind_rows(positive_sample) %>%
    mutate(adjective = fct_reorder(adjective, count_diff))

# Plot
filtered_df %>%
  ggplot(aes(x = adjective, y = count_diff, fill = Category)) +
  geom_col() +
  coord_flip() +
  scale_fill_manual(values = c("Women" = "red", "Men" = "blue"),
                    name = "Phrases used more about") +
  theme_minimal() +
  labs(
    title = "Difference in Adjective-Noun Pair Counts",
    x = NULL,
    y = "Count Difference",
    fill = "Category"
```

```
  ) +
  geom_hline(yintercept = 0, color = "black")
```


Difference in Adjective−Noun Pair Counts
pregnant
unemployed
Right
regarded
pretty
Presbyterian
Polish
passionate
orange
nude
marriageable
looser
latter
interesting
infuriate
ideal
helpless
handsome
forsaken
forlorn
Druse
destitute
deserted
delicate
countless
Circassian
charming
Catholic
Bulgarian
barefooted
attractive
angry
Wild
unquestionably
ugly
privileged
lunatic
irascible
intermediate
illiberal
dumb
commuted
sober
short
scrupulous
same
noble
furlough
elderly
devoted
courageous
ablest
−2
−1
0
1
2
Count Difference
Phrases used more about
Men
Women

```r
# take the top adjectives unique to men; because there are so many with a max of 2, sample som
positive_sample <- comparison %>%
  filter(count_diff > 0) %>%
  arrange(desc(count_diff)) %>%
  top_n(20) %>%
  mutate(Category = "Women", Color = "blue")

# take the top adjectives unique to women
negative_sample <- comparison %>%
  filter(count_diff < 0) %>%
  top_n(-20) %>%
  mutate(Category = "Men", Color = "red")

# combine men and women
filtered_df <- negative_sample %>%
  bind_rows(positive_sample) %>%
  mutate(adjective = fct_reorder(adjective, count_diff)) %>%
  arrange(desc(count_diff)) %>%
  select(adjective, count_diff, Category, Color) %>%
  arrange(desc(abs(count_diff))) %>%
  mutate(adjective = fct_reorder(adjective, count_diff))

make_kable_table <- function(data_slice) {
  data_slice %>%
    mutate(
      `Used More About` = if_else(
        Category == "Women",
        cell_spec(Category, format = "latex", color = "blue"),
        cell_spec(Category, format = "latex", color = "red")
      )
    ) %>%
    select(
      Adjective        = adjective,
      `Count Difference` = count_diff,
      `Used More About`
    ) %>%
    kable(
      format   = "latex",
      booktabs = TRUE,
      escape   = FALSE
    ) %>%
    kable_styling(
```

```
      full_width = FALSE,
      position   = "center",
      font_size  = 7
    )
}


table1 <- make_kable_table(filtered_df %>%
                             slice(1:100))

table2 <- make_kable_table(filtered_df %>%
                             slice(101:200))

table3 <- make_kable_table(filtered_df %>%
                             slice(201:300))
```

```
table1
```

```
table2
```

```
table3
```

```
# top adjectives used more about women (count_diff > 0)
positive_sample <- comparison %>%
  filter(count_diff > 0) %>%
  slice_max(order_by = count_diff, n = 20) %>%
  mutate(Category = "Women")

# top adjectives used more about men (count_diff < 0)
negative_sample <- comparison %>%
  filter(count_diff < 0) %>%
  slice_min(order_by = count_diff, n = 20) %>%
  mutate(Category = "Men")

# combine, with Women first and all Women rows together
filtered_df <- bind_rows(positive_sample, negative_sample) %>%
  mutate(
    Category = factor(Category, levels = c("Women", "Men")),
    adjective = fct_reorder(adjective, count_diff)
  ) %>%
```

Adjective	Count Difference	Used More About
pregnant	2	Women
-	-2	Men
American	-2	Men
British	-2	Men
Gentle	-2	Men
Gentle-	-2	Men
Liberal	-2	Men
Military	-2	Men
Poor	-2	Men
able	-2	Men
ablest	-2	Men
absent	-2	Men
accomplished	-2	Men
active	-2	Men
acute	-2	Men
aggrieved	-2	Men
alive	-2	Men
amiable	-2	Men
anxious	-2	Men
apt	-2	Men
armed	-2	Men
astute	-2	Men
benevolent	-2	Men
best	-2	Men
blind	-2	Men
bold	-2	Men
brave	-2	Men
busy	-2	Men
calm	-2	Men
candid	-2	Men
capable	-2	Men
careful	-2	Men
certain	-2	Men
charitable	-2	Men
chief	-2	Men
civilized	-2	Men
clergy-	-2	Men
cleverest	-2	Men
commercial	-2	Men
competent	-2	Men
conscientious	-2	Men
considerable	-2	Men
consistent	-2	Men
conversant	-2	Men
corrupt	-2	Men
courageous	-2	Men
crotchetty	-2	Men
cruel	-2	Men
dangerous	-2	Men
deserving	-2	Men
desirable	-2	Men
desirous	-2	Men
determined	-2	Men
devoted	-2	Men
devout	-2	Men
different	-2	Men
disabled	-2	Men
disappointed	-2	Men
disciplined	-2	Men
discreet	-2	Men
dishonest	-2	Men
disinterested	-2	Men
disloyal	-2	Men

Adjective	Count Difference	Used More About
gallant	-2	Men
generous	-2	Men
gentle	-2	Men
gentle-	-2	Men
genuine	-2	Men
gifted	-2	Men
greater	-2	Men
greatest	-2	Men
happy	-2	Men
hardy	-2	Men
healthiest	-2	Men
high	-2	Men
highminded	-2	Men
holy	-2	Men
honest	-2	Men
honourable	-2	Men
humane	-2	Men
humble	-2	Men
hungry	-2	Men
idle	-2	Men
ignorant	-2	Men
illiterate	-2	Men
illustrious	-2	Men
impartial	-2	Men
important	-2	Men
incompetent	-2	Men
independent	-2	Men
individual	-2	Men
industrious	-2	Men
inefficient	-2	Men
inexperienced	-2	Men
inferior	-2	Men
infirm	-2	Men
influential	-2	Men
ing	-2	Men
ingenious	-2	Men
injured	-2	Men
intellectual	-2	Men
interested	-2	Men
judicious	-2	Men
just	-2	Men
large	-2	Men
legal	-2	Men
less	-2	Men
liable	-2	Men
liberal	-2	Men
like	-2	Men
likely	-2	Men
literary	-2	Men
little	-2	Men
local	-2	Men
logical	-2	Men
loyal	-2	Men
marked	-2	Men
medical	-2	Men
meritorious	-2	Men
middle	-2	Men
military	-2	Men
miserable	-2	Men
moderate	-2	Men
monied	-2	Men
mortal	-2	Men
native	-2	Men

Adjective	Count Difference	Used More About
rational	-2	Men
ready	-2	Men
real	-2	Men
reasonable	-2	Men
reckless	-2	Men
representative	-2	Men
respected	-2	Men
responsible	-2	Men
rich	-2	Men
right	-2	Men
robust	-2	Men
safe	-2	Men
sagacious	-2	Men
same	-2	Men
sane	-2	Men
scientific	-2	Men
scrupulous	-2	Men
second	-2	Men
selfish	-2	Men
sensible	-2	Men
sensitive	-2	Men
serious	-2	Men
sharp	-2	Men
short	-2	Men
shrewd	-2	Men
sincere	-2	Men
skilled	-2	Men
small	-2	Men
sober	-2	Men
sound	-2	Men
spirited	-2	Men
stalwart	-2	Men
steady	-2	Men
stronger	-2	Men
strongest	-2	Men
stupid	-2	Men
substantial	-2	Men
successful	-2	Men
sufficient	-2	Men
suitable	-2	Men
superior	-2	Men
talented	-2	Men
tall	-2	Men
temperate	-2	Men
thorough	-2	Men
thoughtful	-2	Men
true	-2	Men
trustworthy	-2	Men
unable	-2	Men
unaccustomed	-2	Men
uneducated	-2	Men
unfit	-2	Men
unknown	-2	Men
unofficial	-2	Men
unpopular	-2	Men
unprincipled	-2	Men
unprofessional	-2	Men
unreasonable	-2	Men
unrepresented	-2	Men
unscrupulous	-2	Men
untried	-2	Men
upright	-2	Men
useful	-2	Men

```r
  arrange(Category, desc(abs(count_diff))) %>%
  select(adjective, count_diff, Category)

make_kable_table <- function(data_slice) {
  data_slice %>%
    mutate(
      UsedMoreAbout = if_else(
        Category == "Women",
        cell_spec(Category, format = "latex", color = "blue"),
        cell_spec(Category, format = "latex", color = "red"))) %>%
    select(
      Adjective = adjective,
      `Count Difference` = count_diff,
      `Used More About` = UsedMoreAbout    # Category not shown, just used for color
    ) %>%
    kable(
      format = "latex",
      booktabs = TRUE,
      escape = FALSE
    ) %>%
    kable_styling(full_width = FALSE, position = "center")
}

# ---- Create slices (adjust ranges as needed) ----
# If you have fewer rows, make these smaller (e.g., 1:20, 21:40, etc.)

table1 <- make_kable_table(filtered_df %>%
                             slice(1:200))

table2 <- make_kable_table(filtered_df %>%
                             slice(201:400))

table3 <- make_kable_table(filtered_df %>%
                             slice(401:600))
```

```r
table1
```

```r
table2
```

Adjective	Count Difference	Used More About
pregnant	2	Women
Catholic	1	Women
helpless	1	Women
interesting	1	Women
Bulgarian	1	Women
Circassian	1	Women
Druse	1	Women
Polish	1	Women
Presbyterian	1	Women
Right	1	Women
angry	1	Women
attractive	1	Women
barefooted	1	Women
charming	1	Women
countless	1	Women
delicate	1	Women
deserted	1	Women
destitute	1	Women
forlorn	1	Women
forsaken	1	Women
handsome	1	Women
ideal	1	Women
infuriate	1	Women
latter	1	Women
looser	1	Women
marriageable	1	Women
nude	1	Women
orange	1	Women
passionate	1	Women
pretty	1	Women
regarded	1	Women
unemployed	1	Women
-	-2	Men
American	-2	Men
British	-2	Men
Gentle	-2	Men
Gentle-	-2	Men
Liberal	-2	Men
Military	-2	Men
Poor	-2	Men
able	-2	Men
ablest	-2	Men
absent	-2	Men

Adjective	Count Difference	Used More About
notorious	-2	Men
odd	-2	Men
official	-2	Men
older	-2	Men
oldest	-2	Men
only	-2	Men
open	-2	Men
oppressed	-2	Men
ordinary	-2	Men
outspoken	-2	Men
own	-2	Men
patient	-2	Men
patriotic	-2	Men
peaceable	-2	Men
pious	-2	Men
plain	-2	Men
police-	-2	Men
political	-2	Men
poorer	-2	Men
popular	-2	Men
possible	-2	Men
practical	-2	Men
private	-2	Men
professional	-2	Men
prominent	-2	Men
prone	-2	Men
proper	-2	Men
proud	-2	Men
qualified	-2	Men
quiet	-2	Men
rash	-2	Men
rational	-2	Men
ready	-2	Men
real	-2	Men
reasonable	-2	Men
reckless	-2	Men
representative	-2	Men
respected	-2	Men
responsible	-2	Men
rich	-2	Men
right	-2	Men
robust	-2	Men
safe	-2	Men

Adjective	Count Difference	Used More About

```
table3
```

When tested, we find that there are very few words that are used much more often to refer to one gender than the other, and those that there are occur quite rarely – a maximum of two times over all of the 1860s. The small numbers are a warning against overinterpreting our findings. Nevertheless, the differences are instructive. Along with being "pregnant," women are more likely to be depicted as vulnerable – "forlorn," "delicate," "forlorn," "deserted," "destitute" – and as an aesthetic object of desire – "charming," "attractive," "ideal," "passionate." When women are depicted as dangerous, they show a surplus of passion; they are "angry," "infuriat[ing]." Men are more likely to be assigned virtues of character ("intellectual," "enterprising," "courageous," "determined," "devoted," "humble," "honourable," "holy," "reasonable," "scrupulous") or capacity ("sturdy," "hale," "robust," "strongest) – or vices that show them as agents ("dangerous," "evil," "selfish," "reckless") or as having had their agency temporarily taken away ("aggrieved", "inefficient," "ignorant").

Our findings confirm that members of parliament articulated a gendered politics of agency as they discussed the matters of their day – assigning agency as the entitled domain of one gender, and vulnerability as characteristic of the other. While these findings may not be unexpected, they nonetheless affirm that our methodological approach can mirror established understandings of the period. At the same time, the results offer a different vantage point into the debates. Comparing the two visualizations side-by-side perhaps emphasizes the difference in the ways in which parliamentarian's imagined men and women.

Extracting Simple Subject-Verb-Object Triples

Extracting subject-verb-object (SVO) triples can provide another pattern through which we can understand the relationships and actions within a sentence, enabling analysis of agency and interactions between entities. Adjective-noun pairs, while useful for identifying descriptive attributes, lack this relational context and may not capture the dynamics of who is doing what, and to whom.

To learn about SVOs, we will explore how language was used by Parliamentarians during debates over the Contagious Diseases Acts in nineteenth-century Britain, as the adjectives used to describe women carried ideological significance. The pairings of adjectives and nouns both reflected social attitudes and also actively constructed categories of blame and protection under the law, determining who was subject to regulation and who was deemed worthy of sympathy. However, adjective-noun pairs are only one angle of analysis and must be situated within broader discourse patterns to fully understand how gendered power operated through legislative discourse.

For this analysis, we will extract simple subject-verb-object (SVO) triples. By calling these "simple" triples, we refer to subjects and objects that have a direct syntactic relationship with a root verb. A root verb is the main verb of a sentence that serves as the core action or state. Sentences can also

include other types of verbs. These verb types include auxiliary verbs (e.g., "is," "have"), which provide grammatical support to the main verb; modal verbs (e.g., "can," "should"), which express necessity or possibility; and participles, which are verb forms used as adjectives or to create verb phrases (e.g., "running" in "He is running"). Additionally, sentences may contain compound verbs, which consist of two or more verbs working together to express a single action or idea, such as "has been running." Phrasal verbs combine a verb with one or more particles (prepositions or adverbs) to create a meaning distinct from the individual words, as in "give up," which means "to quit." Infinitive verbs, the base form of a verb often preceded by "to," such as "to run," can function as nouns, adjectives, or adverbs in a sentence. However, for the purpose of extracting simple SVO triples, our focus will remain on the root verb and its direct relationships.

We will focus our analysis on sentences extracted from the 1860 Hansard debates that include the words "woman" or "women" stored in the variable named `hansard_woman_1860`. Our goal is to investigate how women are framed in Parliamentary discourse—specifically, whether they are portrayed as political agents, social subjects, or as objects within legislative or societal contexts. Additionally, we will examine the frequency and nature of specific associations made with women in these discussions to uncover possible trends.

For this analysis, we will also perform lemmatization on the sentences. We will again use spaCy for this task. To lemmatize with spaCy, we set the `lemma` parameter of `spacy_parse()` to `TRUE`. We will keep `entity` as `FALSE` to save computational resources and processing time, since we are not analyzing named entities.

We will also transition to analyzing the 1870 debates, a period when women's voting rights were frequently contested in Parliament, revealing deeper struggles over power and political agency.

```
library(tidyverse)
library(hansardr)

data("hansard_1870")

hansard_woman_1870 <- hansard_1870 %>%
  filter(str_detect(text, regex("woman|women", ignore_case = T)))

parsed_hansard_woman_1870 <- spacy_parse(hansard_woman_1870$text,
                                         dep = TRUE,
                                         lemma = TRUE,
                                         entity = FALSE)

head(parsed_hansard_woman_1870)
```

```
##   doc_id sentence_id token_id     token     lemma   pos head_token_id dep_rel
## 1  text1           1        1       Nor       nor CCONJ             2      cc
## 2  text1           1        2       was        be   AUX             2    ROOT
```

```
## 3  text1           1        3 education education  NOUN             2   nsubj
## 4  text1           1        4       the       the   DET             6     det
## 5  text1           1        5      only      only   ADJ             6    amod
## 6  text1           1        6    object    object  NOUN             2   nsubj
```

We can now extract simple SVO triples from the lemmatized sentences. To do so, we will define a new function called `get_simple_lemmatized_svo_triples()`. In this function, we are careful to match dependencies such as subjects (e.g., "nsubj" and "nsubjpass"), verbs (e.g., "ROOT"), and objects (e.g., "dobj," "pobj," "iobj") in a way that preserves the semantic relationships between these elements within a sentence.

A single sentence can contain multiple subjects, verbs, and objects, but not all of these components relate directly to each other. For instance, a complex sentence from the 19th-century Hansard debates might read: "The honorable gentleman proposed the motion, and the members of the House debated the issue vigorously." Here, the verb "proposed" relates to "gentleman" as the subject and "motion" as the object, while "debated" relates to "members" as the subject and "issue" as the object. By joining based on the dependency relations and ensuring the tokens share the same governing head (using the "root_token_id" and "head_token_id"), we attempt to extract only meaningful SVO triples while avoiding inaccuracies, such as mistakenly attributing actions to subjects that were not actually performed in the sentence.

Instead of matching raw tokens we match on lemmas. Lemmas are the "base" forms of words — their simplest and most fundamental forms, without any inflectional changes. Inflection refers to grammatical modifications of a word, such as changes in tense, case, or number. For example, "run," "runs," and "ran" are inflectional variations of the same base word, or lemma, "run." By shifting to lemma-based matching, we reduce the variability in the extracted triples. This approach unifies different forms of the same word, enabling us to identify broader patterns. As a result, our text analysis becomes more comprehensive for our purposes by focusing on the underlying meaning of the triples rather than their morphological variations.

```
get_simple_lemmatized_triples <- function(parsed_data) {
  svo_triples <- parsed_data %>%
    group_by(doc_id, sentence_id) %>%

    # Identify ROOT tokens (usually verbs) by doc and sentence
    filter(dep_rel == "ROOT") %>%
    select(doc_id,
           sentence_id,
           root_token_id = token_id,
           verb = lemma) %>%
    ungroup() %>%

    # Join with subject tokens
```

```
    left_join(parsed_data %>%
                filter(dep_rel %in% c("nsubj", "nsubjpass")) %>%
                select(doc_id,
                       sentence_id,
                       head_token_id,
                       subject = lemma),
              by = c("doc_id", "sentence_id", "root_token_id" = "head_token_id"),
              relationship = "many-to-many") %>%

    # Join with object tokens
    left_join(parsed_data %>%
                filter(dep_rel %in% c("dobj", "obj", "pobj", "iobj")) %>%
                select(doc_id,
                       sentence_id,
                       head_token_id,
                       object = lemma),
              by = c("doc_id", "sentence_id", "root_token_id" = "head_token_id"),
              relationship = "many-to-many") %>%

    # Combine SVO into a single string for readability, skipping NAs
    mutate(svo_triple = paste(if_else(is.na(subject), "", subject),
                              if_else(is.na(verb), "", verb),
                              if_else(is.na(object), "", object),
                              sep = " "),
           svo_triple = str_squish(svo_triple)) %>% # Remove excess white space

    filter(!is.na(subject),
           !is.na(verb),
           !is.na(object)) %>%

    ungroup() %>%
    select(doc_id, sentence_id, subject, verb, object, svo_triple)

  return(svo_triples) }
```

```
simple_svo_triples_1870 <- get_simple_lemmatized_triples(parsed_hansard_woman_1870)

head(simple_svo_triples_1870)
```

```
## # A tibble: 6 x 6
##   doc_id sentence_id subject    verb     object  svo_triple
```

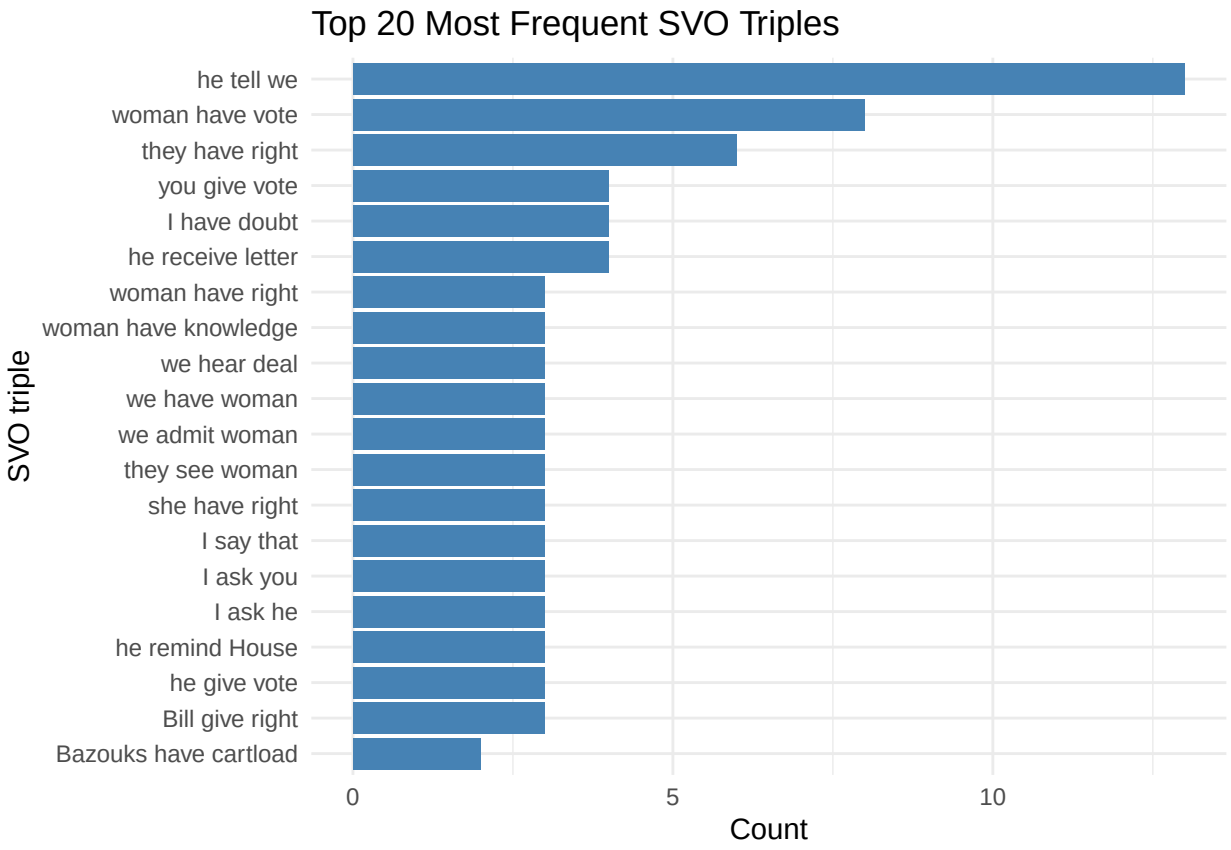
```
##   <chr>        <int> <chr>      <chr>    <chr>   <chr>
## 1 text12           1 they       state    opinion they state opinion
## 2 text17           1 Returns    show     excess  Returns show excess
## 3 text20           1 authority  palm     bad     authority palm bad
## 4 text21           1 Parliament think    it      Parliament think it
## 5 text26           1 they       purchase land    they purchase land
## 6 text27           1 he         know     man     he know man
```

Exploring Our Results

We can now visualize our results. Within the top most frequent subject-verb-object triples are utterances such as "woman have vote," "they have right," "woman have right," "she have right," and "you give vote." The utterances represented by the triples gesture towards discussions about women's suffrage. While 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, Victorian feminists pushed for the right to vote during the 1860s and 1870s.

Not all of the extracted triples are easy to interpret. The top triple, "he tell we," results from idiosyncracies in how spaCy predicts grammatical relationships, especially when analyzing historic texts or sentences with complex structures. Older forms of English might use pronouns like "we" in ways that modern models misinterpret, or phrases like "He tells us what we must do" might be incorrectly simplified, leading to outputs such as "he tell we." Such misrepresentations of language highlight the limitations of applying modern NLP tools to historical or linguistically diverse texts. Further preparing the data for presentation could include manually removing these oddities from the output. For the sake of transparency, we will leave these constructions for now.

```
# Find the Most Common Simple SVO Triples
triples_counts_1870 <- simple_svo_triples_1870 %>%
  count(svo_triple, sort = TRUE) %>%
  slice_head(n = 20)

ggplot(triples_counts_1870,
       aes(x = reorder(svo_triple, n),
           y = n)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(x = "SVO triple",
       y = "Count",
       title = "Top 20 Most Frequent SVO Triples") +
  theme_minimal()
```



If the phrases look ungrammatical, remember that we have lemmatized the words previous to analyzing their grammatical dependencies. Thus "I asked him" is rendered "I ask he," because we have gathered instances like "I ask him," "I am asking him," etc., counting them all the same.

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. In this exercise, we see that many of the most prevalent SVO phrases have to do with reporting – someone says "he told us such and such facts," or "we heard a great deal about this," or "I am saying that. . . " Women as well as men are being reported on. Someone is saying, "they saw women [doing something]."

We read many sentences in this chart where gender and rights are foregrounded, including "woman have vote," "woman have right," and "woman have knowledge." They seem to indicate that some members of parliament are talking about the empowerment of women – a different perspective than that suggested by our previous exercises.

We could perform different kinds of analyses with our extracted triples. We could compare times in which women and men are imagined to have the vote. But perhaps the first thing we should do is to call up some context for the handful of phrases alluding to women having rights or the vote.

```
svo_triples_context_1870 <- hansard_woman_1870 %>%
  filter(str_detect(text,
                    regex("right|vote",
                          ignore_case = TRUE))) %>%
  select(text)
```

text
He held, therefore, that in all cases where persons had been naturalized against their will, and where married women and children sought to be reinstated in their rights in this country, the Government should have power to grant their request.
, in moving for leave to bring in a Bill to remove the Electoral Disabilities of Women, said, that the effect of the measure would be to enable women householders who were possessed of the qualification established by Parliament to vote for Members of Parliament.
It was proposed to inquire into the “ “character” ” of conventual and monastic institutions, their existence, increase, and property; its objects were evident from the antecedents of all those who were its chief promoters, and who had, in 1851, 1852, 1853, and 1854, inflamed the religious passions and fanaticism of the people, flooded the country with the vilest calumnies and accusations against those holy women who devoted themselves to the noblest objects, and carried out, in.
A woman might have married a British subject, and might never have intended that by any force of law she should become a foreign subject; yet, by a former clause in the Bill combined with the present one, if the husband made himself an alien, the wife, although residing with her children in this country, and judicially separated from her husband, who lived abroad, would be made a foreign subject against her own will, and against that of her family, through his changing his nationality, and if she were afterwards to reside in any foreign country, she would be deprived of all the rights, privileges, and protection to which a British subject would be entitled.
He was surprised that, at a time when the rights of women were so loudly advocated, the House should seem determined thus to curtail them, and that no one but the worthy Alderman (Mr. Alderman Lawrence) had been found to say a word in their defence.

What we see is mainly parliament entertaining a series of propositions about the circumstnaces under which women gain or lose rights through marriage or naturalization. In one case, a debate is taking place on empowering women through removing the “electoral disabilities of women,” but it does not seem to have gone very far – in fact, the last excerpt contains a speaker wondering why so few members of parliament are speaking up for women’s rights.

This, however is an incomplete picture, because we are only looking at the top triples overall.

Looking for Women as Subject or Object

We can better explore how parliament spoke about women more by searching for all the triples where "woman" or "women" appear. Do they appear as a subject or an object?

To specifically match rows where the subject is "woman," we can take one of two approaches. The first option is to filter directly on the subject column, identifying rows where the word "woman" appears. Alternatively, we can use the `str_detect()` function on the `svo_triple` column to find triples that start with "woman." By applying the regular expression ^woman, we ensure that only strings beginning with the word "woman" are matched, as the ^ symbol anchors the pattern to the start of the string.

```
# Finding Triples Where Women are the Subject
triples_with_woman_as_subject_1870 <- simple_svo_triples_1870 %>%
  filter(str_detect(svo_triple, "^woman"))

top_10_triples_with_woman_as_subject_1870 <- triples_with_woman_as_subject_1870 %>%
  count(svo_triple) %>%
  arrange(desc(n)) %>%
  slice(1:10)

head(top_10_triples_with_woman_as_subject_1870, 10)
```

```
## # A tibble: 10 x 2
##    svo_triple              n
##    <chr>               <int>
##  1 woman have vote         8
##  2 woman have knowledge    3
##  3 woman have right        3
##  4 woman carry trade       2
##  5 woman do work           2
##  6 woman earn bread        2
##  7 woman exercise right    2
##  8 woman feel sense        2
##  9 woman have interest     2
## 10 woman marry brother     2
```

As shown in the above table, the top triple with the word woman as subject in the 1860's British Parliamentary debates is "woman have vote" followed by "woman have right." Other top triples discuss women's agency, such as "woman exercise right." Women are also presented as having trades, doing work, earning their bread, and having interests – in other words, in the era of the Married Women's Property Act, women are very much represented as participating in the economy.

```r
# Finding Triples Where Women are the Object

triples_with_woman_as_subject_1870 <- simple_svo_triples_1870 %>%
  filter(str_detect(svo_triple, " woman"))

top_10_triples_with_woman_as_object_1870 <- triples_with_woman_as_subject_1870 %>%
  count(svo_triple) %>%
  arrange(desc(n)) %>%
  slice(1:10)

head(top_10_triples_with_woman_as_object_1870, 10)
```

```
## # A tibble: 10 x 2
##    svo_triple           n
##    <chr>            <int>
##  1 they see woman       3
##  2 we admit woman       3
##  3 we have woman        3
##  4 I ask woman          2
##  5 I find woman         2
##  6 I say woman          2
##  7 it enable woman      2
##  8 it prevent woman     2
##  9 they have woman      2
## 10 we find woman        2
```

Here, women are seen, interviewed, searched for, talked about, and prevented from doing things.

Looking for the Verb 'Vote'

One last exercise with Hansard. We can also look for SVO triples where the verb is "vote," and see who is doing the voting. In the code below, we use the `kwic()` (Key Word in Context) function from the `quanteda` library (introduced in earlier chapters) to revisit the original documents and provide context for our analysis. By specifying "vote" as the pattern to match, we focus on instances of this word in the text. Additionally, we can define the size of the window, which determines the number of words displayed before and after each occurrence of the pattern, giving us insight into its surrounding context. To ensure we capture all instances of the word "vote" regardless of capitalization (e.g., "Vote" or "VOTE"), we can enable case-insensitive matching. We will use `sample()` to show just 20 of the results, randomly selected. Therefore, your 20 results might look different from ours. Using `sample()` multiple times could reveal more interesting results.

```
library(quanteda)

hansard_woman_1870_corpus <- corpus(hansard_woman_1870,
                                    text_field = "text")

hansard_woman_1870_corpus_sample <- sample(hansard_woman_1870_corpus, 20)

vote_kwic <- kwic(tokens(hansard_woman_1870_corpus_sample),
                  pattern = "vote",
                  window = 8,
                  case_insensitive = TRUE)
```

pre	keyword	post

We see from these passages that in the 1860s, members of parliament were rexplicitly reflecting on when women had previously enjoyed a right to vote and whether they might again in the future.

Conclusion: `spaCy` as Method

Many of the findings in this chapter are hardly earth-shattering, but they do give the pattern for a kind of exercise that can be executed over centuries and over databases with far more text – asking, what are men and women imagined to do? Is it the same or different? What is done to them? Does it vary by gender? The answers open up a whole world of analysis.

Triples extraction can extend into realms beyond the analysis of gendered pronouns. Grammatical POS analysis can show how agency was imagined not only around gender, but also around ethnicity, nationality, labor, and commodities, for instance by extracting the verb-object structures invoked when British parliamentarians discussed labor in the colonies (we might expect to see "beating laborers" as a common construction in some places, and "sold slaves" as a construction elsewhere). Linguistic constructions encode specific historical imaginaries of control, resistance, and economic exploitation. In *The Dangerous Art of Text Mining,* Guldi uses triples to explore statements about the past and future, comparing how imagined futures changed over the nineteenth century.

By tracing the imagination of who is permitted to act upon whom—or what—and how, grammatical POS analysis allows analysts to map patterns of agency and subjectivity across large corpora and over time. This, in turn, makes it possible to reconstruct the logics that underpinned imperial governance, including the ways in which laboring bodies were rendered legible as property, threat, or burden. Such an approach thus offers a means of analyzing narrative, revealing both the overt and subtle mechanisms by which domination was justified, contested, and naturalized.

Analysts must remain critically attuned to how tools mediate knowledge. Technical choices—such as using a pre-trained model from `spaCy`, applying default tokenization, or choosing whether or not

to lemmatize words are never just technical. The choices we make shape the contours of our inquiry. To analyze a corpus is not just to study a set of texts, but to negotiate a methodological contract: in saying that we will analyze "men" or "women" we choose what we can see, and in doing so, define what can be known. At the same time, the object of study can push back. Analyzing women might impress upon us questions about men, or might lead to questions about whether adjective-noun pairs are even the right approach for analyzing action and agency, or whether analyzing constructs with verbs might be better. Method and object of study are in constant dialogue: as we process and measure the corpus, the corpus in turn reveals not only the affordances but also the limitations of our tools and frameworks.

Such a problem also extends to the type of model we use to analyze the data. Using `spaCy` we are able to transition fluidly between analyzing the words "woman" and "man" using the same functions, and by using `spaCy`'s pre-trained language models (LMs), we prioritize linguistic analysis over the computationally intensive task of training our own model on the Hansard data. This methodological choice allows the analyst to transition directly into research, but also constrains the kinds of questions we can ask, and how reliably we can answer them.

We are able to transition so fluidly between an analysis of the words "woman" and "man" by using the same function and using `spaCy`'s pre-trained language model we can prioritize linguistic analysis over the computationally intensive task of model training. However, the results of these models must be examined critically, as they may reflect biases or inaccuracies, particularly when applied to historical corpora. Pre-trained models are typically trained on contemporary language data, which can differ substantially from sources like the 19th-century Hansard debates. This mismatch often results in parsing errors. In such cases, researchers must choose between manually correcting these errors or retraining the model on domain-specific data. Given the time and resource demands of retraining, manual correction is frequently the more practical approach.

When relying on a language model to perform part-of-speech extraction, we have to remember that the model is not actually parsing the sentence in the linguistic sense. It is predicting the most statistically likely tags, based on a distribution of training examples. That distinction matters a great deal in historical work.

For contemporary, conversational English, the model's internal priors line up fairly well with the grammatical structures it sees. But once we shift into syntactically dense prose—Victorian parliamentary rhetoric, nineteenth-century religious writing, or the elaborate hypotactic style that stacks clause upon clause—the model's assumptions begin to misfire. A statistical model trained mostly on contemporary language is far more likely to prematurely assign a tag based on surface cues than to actually follow the logic of the syntactic construction.

This limitation is not about vocabulary but structure. A model can recognize the word "whereupon" or "therewith" or "henceforward," but it does not internalize the rhetorical function those connectives play in a multi-clausal construction. The tagging becomes especially fragile in passages where subject and predicate are separated by highly ornamented dependent clauses—a defining stylistic trait of Victorian prose.

From a diigtal history perspective, this matters because part-of-speech data is never just a neutral

preprocessing step. It shapes what becomes legible downstream: topic modelling, collocations, sentiment arcs, the mapping of agency and attribution—each of these can shift meaningfully depending on whether the model correctly identified a subject, object, or modifier. If the parser constantly reinterprets earlier portions of a sentence through a modern syntactic lens, the interpretive terrain is already tilted before analysis even begins.

This is why corpus-linguistic methods built for historical language typically rely on rule-based or hybrid parsing architectures, often supplemented by domain-specific grammars or custom training data, rather than statistical next-token prediction. A language model can approximate grammatical structure, but approximation may not always be enough when grammatical nuance is the object of interpretation.

Exercises

Grammatical part-of-speech analysis has meaningful implications for historical research and for understanding how action and agency are imagined within social and legal contexts.

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 by speakers. To strengthen your understanding of how grammatical POS analysis 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 1860 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 a verb of choice 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 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 spaces.

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 adjective noun pairs are distinctive of one year but not another, and vice-versa (e.g. which pairs 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).