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

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

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 In the lead-up to the Great Reform Act of 1832, when the middle class got the vote in Britain, which historical references were on the table?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

This chapter builds the readers’ sense of how to work with text to create insight about history. It introduces counting for n-grams, or multi-word phrases. It also introduces the use a “controlled vocabulary,” or expert provided list of important words and phrases, to explore which events were spoken about in parliament during the 1830s.

This chapter builds on the previous one to introduce readers to fundamental data processing techniques in R, commonly used for preparing data for analysis. A key task in data analysis is summarizing a dataset’s characteristics to gain insights into its basic features. One essential concept in data processing is “joining” datasets. A join combines two datasets into one, enabling analysts to analyze parliamentary texts alongside metadata such as speaker information, dates, and significant events. Additionally, this chapter teaches readers how to work with event-based data, including years, months, and dates.

As with the previous chapter, this chapter will emphasize the skills of moving from analyzing text as data to performing a close reading of historical speeches in context. This approach provides analysts with the tools of deciding for themselves which patterns identified during text mining are supported by closely reading text. It will emphasize that researchers work with text in an iterative process, moving from applying metrics to word counts and visualizing trends to reading, and then often repeating this deductive process. Reading leads to specifying historical questions, which may mean more measuring, more visualization, and more guided reading in pursuit of a historical analysis.

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

1

**Working With Debate Metadata**

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

First we can load the text data from the 1860s speeches, recalling that one of the columns, sentence\_id, refers to the number of each sentence and its location in the nineteenth-century printed volumes of Hansard’s parliamentary debates, referenced by series (“S”), volume (“V”), and page (“P”). Sentences are listed in the order in which they appeared on the page.

**library**(hansardr)

**data**("hansard\_1860")

**head**(hansard\_1860)

## sentence\_id

## 1 S3V0156P0\_0

## 2 S3V0156P0\_1

## 3 S3V0156P0\_2

## 4 S3V0156P0\_3

## 5 S3V0156P0\_4

## 6 S3V0156P0\_5

##

## 1

## 2

## 3

## 4 My Lords, it is with diffidence that I rise to address your Lordships for the first time, and the ## 5

## 6

Next, we can load metadata that correspond with the Hansard text.

**data**("debate\_metadata\_1860")

**head**(debate\_metadata\_1860)

## sentence\_id speechdate debate

## 1 S3V0156P0\_0 1860-01-24 MEETING OF THE PARLIAMENT.

## 2 S3V0156P0\_1 1860-01-24 THE QUEEN’S SPEECH.

## 3 S3V0156P0\_2 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

## 4 S3V0156P0\_3 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

## 5 S3V0156P0\_4 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

## 6 S3V0156P0\_5 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

hansardr includes detailed contextual information for each of these speeches, using the same system of references that we saw in the previous data set – series, volume, page, and sentence number. Here, however,

2

there is no “text” column, but rather a series of columns that illuminate multiple “fields” of each sentence: the “speechdate” on which the speech was given and the official title of the “debate” in which the sentence was spoken.

hansardr offers two additional data files: speaker\_metadata\_1860 and file\_metadata\_1860. **data**("speaker\_metadata\_1860")

**head**(speaker\_metadata\_1860)

## sentence\_id speaker suggested\_speaker ambiguous ## 1 S3V0156P0\_0 THE PARLIAMENT, 0 ## 2 S3V0156P0\_1 THE QUEEN peter\_quinn\_4670 0 ## 3 S3V0156P0\_2 EARL FITZWILLIAM, charles\_fitzwilliam\_1441 0 ## 4 S3V0156P0\_3 EARL FITZWILLIAM, charles\_fitzwilliam\_1441 0 ## 5 S3V0156P0\_4 EARL FITZWILLIAM, charles\_fitzwilliam\_1441 0 ## 6 S3V0156P0\_5 EARL FITZWILLIAM, charles\_fitzwilliam\_1441 0 ## fuzzy\_matched ignored

## 1 0 0

## 2 1 0

## 3 0 0

## 4 0 0

## 5 0 0

## 6 0 0

The speaker\_metadata\_1860 dataset provides information about the “speaker” of each sentence as recorded in the original Hansard debates. It also includes a “suggested speaker” column, which is an attempt to disambiguate speaker names by distinguishing between individuals with similar or identical names. For instance, it differentiates between the various individuals referred to as “Mr. Gladstone” or “Mr. Smith.”

**data**("file\_metadata\_1860")

**head**(file\_metadata\_1860)

## sentence\_id speech\_id debate\_id src\_file\_id src\_image src\_column ## 1 S3V0156P0\_0 258488 30416 S3V0156P0 S3V0156P0I0033 1 ## 2 S3V0156P0\_1 258489 30417 S3V0156P0 S3V0156P0I0033 1 ## 3 S3V0156P0\_2 258490 30418 S3V0156P0 S3V0156P0I0036 7 ## 4 S3V0156P0\_3 258490 30418 S3V0156P0 S3V0156P0I0036 7 ## 5 S3V0156P0\_4 258490 30418 S3V0156P0 S3V0156P0I0036 7 ## 6 S3V0156P0\_5 258490 30418 S3V0156P0 S3V0156P0I0036 7

The file\_metadata\_1860 dataset provides metadata about the records themselves, including the ID for the original source file, and the column in which the text appears.

We have chosen to store the debate text (hansard\_1860) separately from the contextual information about each speech (debate\_metadata\_1860, speaker\_metadata\_1860), as well as the information about each hard

3

copy file containing the speeches (file\_metadata\_1860). This is because the data in each dataset is quite large. Processing such a large dataset can exceed a computer’s resources, such as its memory. Furthermore, many operations in this book – for instance, counting words per decade – require just one dataset.

Many kinds of analyses, however, will require working with multiple datasets from hansardr joined together. We will use the function left\_join() to join the annotated information about speech context with each of the speeches. A left join returns every record from the left-hand data frame, and all the matched records from the data frame on the right.

**library**(tidyverse)

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 -- ## v dplyr 1.1.4 v readr 2.1.5

## v forcats 1.0.0 v stringr 1.5.1

## v ggplot2 3.5.1 v tibble 3.2.1

## v lubridate 1.9.3 v tidyr 1.3.1

## v purrr 1.0.2

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() -- ## x dplyr::filter() masks stats::filter()

## x dplyr::lag() masks stats::lag()

## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errorhansard\_1860 <- **left\_join**(hansard\_1860, debate\_metadata\_1860)

## Joining with ‘by = join\_by(sentence\_id)‘

**head**(hansard\_1860)

## sentence\_id

## 1 S3V0156P0\_0

## 2 S3V0156P0\_1

## 3 S3V0156P0\_2

## 4 S3V0156P0\_3

## 5 S3V0156P0\_4

## 6 S3V0156P0\_5

##

## 1

## 2

## 3

## 4 My Lords, it is with diffidence that I rise to address your Lordships for the first time, and the ## 5

## 6

## speechdate debate

## 1 1860-01-24 MEETING OF THE PARLIAMENT.

## 2 1860-01-24 THE QUEEN’S SPEECH.

## 3 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

4

## 4 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

## 5 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

## 6 1860-01-24 ADDRESS IN ANSWER TO HER MAJESTY’S SPEECH.

Notice that the resulting dataset, hansard\_corpus\_1860, has information about speaker and date for each sentence of text.

“Joins” are a crucial concept for combining and analyzing data that is distributed across multiple datasets. They allow you to link related information from different tables based on a shared key or common field, enabling a more comprehensive view of the data.

In the example above, both hansard\_1860 and debate\_metadata\_1860 share a column called sentence\_id. This shared column acts as a common identifier, enabling us to link the speaker and date information from one dataset to the corresponding text of the speech in the other dataset. By using this common identifier, the two datasets can be “joined” together to create a single, unified dataset.

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

**Multi-Word Phrases**

Beyond analyzing individual words, an analyst may want to explore multi-word phrases. Multi-word phrases contain formulations of concepts that had powerful political, social, and cultural meaning to speakers, for example, “the will of the people,” a phrase that has been invoked by both radicals and conservatives to support extremely different imaginations of the people. That is, speakers who invoked “the will of the people” might have been bolstering the importance of their speech by gesturing towards their status as a representative of the workers whose labor is the source of all political legitimacy. On the other hand, some speakers might have used the same phrase – “the will of the people” – to kindle an imagination of parliament as a zone a traditional folk whose national identity legitimizes a politics of excluding from citizenship or rights people with certain racial or immigration status. The question becomes: which version of the people was being represented in the discourse in question?

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

We could, in theory, contrive computational approaches to help us discern why people invoked “the will of the people.” But in text mining for historical analysis, we often want to move from identifying a data-driven phenomenon to gaining more information about historical context by reading the actual speeches. This approach provides insight into the rich nuances suggested by speakers by giving us perspective into how they invoke particular phrase in the original records.

Accordingly, in the sections below, we will begin by counting phrases, but that is not where we will end. We will return to the original speeches to read more deeply. This demonstration will be followed by related exercises.

**Finding Bigrams**

While there are several approaches to finding multi-word phrases, we will demonstrate this by tokenizing the text into bigrams, or sequential, two-word phrases.

5

The following code filteres the speeches of the 1860s for the first months of 1867 before the vote on the Second Reform Act, which gave Britain’s urban working-class men the right to vote.

**library**(tidytext)

**library**(lubridate)

hansard\_1867 <- hansard\_1860 **%>%**

**mutate**(year = **year**(speechdate)) **%>%**

**filter**(year **==** 1867,

speechdate **<=** 1867-08-15)

Now that we have filtered the dataset for our desired timeline, we can tokenize the text into bigrams. Tokenizing after filtering ensures that we are working with a smaller and more focused dataset, which makes the process more efficient. Alternatively, we could have done the reverse—tokenize the text first and then filter for the desired dates—but this approach would have used more of our computer’s resources, as it would require processing unnecessary data before narrowing it down. When working with large data sets– like Hansard–the order in which we process data may be key to ensuring the data is manageable and the processing steps can be completed successfully.

The following code uses the unnest\_tokens() function, which we previously used to split text into individual words. This time, we adjust the function to extract pairs of consecutive words instead. To do this, we set the token argument to ngrams, instructing R to create sequences of consecutive words, and the n argument to 2, specifying that each sequence should consist of two words. These two arguments work together to extract two-word phrases, also known as bigrams.

bigrams\_1867 <- hansard\_1867 **%>%**

**unnest\_tokens**(bigram, text, token = "ngrams", n = 2)

**head**(bigrams\_1867)

## sentence\_id speechdate debate year bigram

## 1 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 being seated

## 2 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 seated on

## 3 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 on the

## 4 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 the throne

## 5 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 throne adorned

## 6 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 adorned with

Note that the bigrams extracted overlap, as if the same sentence had been subdivided several times. This output is expected.

**Counting Bigrams** Now let’s count the top bigrams and remove any that contain stop words so that we can focus on meaningful word pairs.

**library**(tidyverse)

**library**(tidytext)

6

clean\_bigrams\_1867 <- bigrams\_1867 **%>%**

**filter**(**!str\_detect**(bigram,

**paste0**("**\\**b(", **paste**(stop\_words**$**word, collapse = "|"), ")**\\**b"))) **%>% filter**(**str\_detect**(bigram, "ˆ[a-z]+ [a-z]+$"))

**head**(clean\_bigrams\_1867)

## sentence\_id speechdate debate year bigram ## 1 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 throne adorned ## 2 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 regal ornaments ## 3 S3V0185P0\_1 1867-02-05 THE QUEEN’S SPEECH. 1867 robes sitting ## 4 S3V0185P0\_2 1867-02-05 THE QUEEN’S SPEECH. 1867 robes commanded ## 5 S3V0185P0\_2 1867-02-05 THE QUEEN’S SPEECH. 1867 gentleman usher ## 6 S3V0185P0\_2 1867-02-05 THE QUEEN’S SPEECH. 1867 black rod

The above code processes the dataset by applying the following transformations and filters:

filter(!str\_detect(...)) - Removes rows where the bigram column contains any stop word. - It con structs a regular expression from a list of stop words (stop\_words$word) by: - Combining them with the OR operator (|). - Surrounding them with word boundaries (\\b) to match complete words only. - This ensures that only bigrams without stop words are retained.

filter(str\_detect(bigram, "ˆ[a-z]+ [a-z]+$")) - Keeps rows where the bigram column matches a specific pattern: - The bigram must consist of exactly two lowercase alphabetic words separated by a single space. - Rows where the bigram contains numbers, punctuation, or non-lowercase letters are excluded.

We can now count the resulting bigrams.

top\_bigrams\_1867 <- clean\_bigrams\_1867 **%>%**

**count**(bigram) **%>%**

**top\_n**(100) **%>%**

**arrange**(**desc**(n))

## Selecting by n

**head**(top\_bigrams\_1867)

## bigram n

## 1 noble lord 1808

## 2 noble earl 1544

## 3 noble friend 1030

## 4 roman catholic 968

## 5 reform bill 742

## 6 select committee 610

7

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

We also see a reference to “public meetings.” Why did speakers in parliament invoke public meetings in the lead-up to the Second Reform Act? We will answer this question by reading the original text.

my\_reading <- hansard\_1867 **%>%**

**filter**(**str\_detect**(text, "public meetings"))

**head**(my\_reading**$**text)

## [1] "I am aware that many of the public meetings have declared in favour of manhood suffrage, but ha## [2] "They are often drawn up with considerable ability; but they bear the mark, I think, of a single## [3] "Speeches at public meetings, and the discussions of the press, great as their influence undoubt## [4] "I know I shall be asked, from what I have seen in the press, and what has been said at public m## [5] "Have you not heard or read what has been said at public meetings of every kind?"

## [6] "There have been more than 1, 000 public meetings; at every one, the doors were open, and any ma

A cursory reading of the sentences suggests that members of parliament were well aware of massive public meetings where more than a thousand members of the public showed up support the vote for working-class people. We read of demonstrations around British Empire, even in Nova Scotia. We read of members of parliament challenging each other to meet the demands of the public: “Have you not heard or read what has been said?” We have only skimmed a few sentences, but the material in these 126 rows gives us what we need to understand how members of parliament talked about public protests in these crucial months of 1867.

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

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

fivegrams\_1867 <- hansard\_1867 **%>%**

**unnest\_tokens**(ngram, text, token = "ngrams", n = 5)

people\_ngrams <- fivegrams\_1867 **%>%**

**filter**(**str\_detect**(ngram, "people")) **%>%**

8

**count**(ngram) **%>%**

**top\_n**(50)

## Selecting by n

**head**(people\_ngrams)

## ngram n

## 1 feelings of the people of 8

## 2 great body of the people 32

## 3 great majority of the people 13

## 4 great mass of the people 15

## 5 majority of the irish people 10

## 6 majority of the people of 27

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

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

minds <- hansard\_1867 **%>%**

**filter**(**str\_detect**(text, "minds of the people"))

**head**(minds**$**text)

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

When speakers referenced the “minds of the people,” they frequently depicted popular politics as a form of collective political achievement to which ordinary people contributed the rational evidence of their own lived experience. We find the phrase connected to ideas about moral “duty” and political “harmony.” We also see it invoked in reference to Ireland, India and Scotland.

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

9

**Using Bigrams to Find Mentions of Events**

For many kinds of textual analyses, incorporating external information can help guide the focus of the analysis and ensure it aligns with specific research questions. For example, a historian analyzing data might be interested in tracking references to notable events in British history, such as the Magna Carta, the Spanish Inquisition, or the Glorious Revolution, to see how often these events are mentioned in parliamentary records. To achieve this, the analyst may create and use a “controlled vocabulary” which is a predefined list of terms or topics of interest related to the analysis. *hansardr* includes an example controlled vocabulary: a curated list of significant events in British history, along with the corresponding years of their occurrence.

In applying this list, this exercise invites readers to investigate the idea that representations of memory change over time, using the approach known in data analysis as a “controlled vocabulary,” where the analyst searches for a set group of phrases. We will use a controlled vocabulary to measure Parliamentarians’ references to past events in the 1860 debates. We are intentionally choosing words such as “riot” or “meeting” that might collect evidence of parliamentary speakers referencing public meetings like those we saw evidence of above. We are also curious about how those references compare with references to famines, wars, strikes, exhibitions, and other contemporary and historical events.

**data**("hansard\_1860")

**data**("debate\_metadata\_1860")

hansard\_1860 <- **left\_join**(hansard\_1860, debate\_metadata\_1860)

## Joining with ‘by = join\_by(sentence\_id)‘

hansard\_1867 <- hansard\_1860 **%>%**

**mutate**(year = **year**(speechdate)) **%>%**

**filter**(year **==** 1867,

speechdate **<=** 1867-08-15)

bigrams\_1867 <- hansard\_1867 **%>%**

**unnest\_tokens**(bigram, text, token = "ngrams", n = 2)

clean\_bigrams\_1867 <- bigrams\_1867 **%>%**

**filter**(**!str\_detect**(bigram, **paste0**("**\\**b(", **paste**(stop\_words**$**word, collapse = "|"), ")**\\**b"))) **%>% filter**(**str\_detect**(bigram, "ˆ[a-z]+ [a-z]+$"))

**head**(clean\_bigrams\_1867)

## sentence\_id speechdate debate year bigram

## 1 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 throne adorned

## 2 S3V0185P0\_0 1867-02-05 THE QUEEN’S SPEECH. 1867 regal ornaments

## 3 S3V0185P0\_1 1867-02-05 THE QUEEN’S SPEECH. 1867 robes sitting

## 4 S3V0185P0\_2 1867-02-05 THE QUEEN’S SPEECH. 1867 robes commanded

## 5 S3V0185P0\_2 1867-02-05 THE QUEEN’S SPEECH. 1867 gentleman usher

## 6 S3V0185P0\_2 1867-02-05 THE QUEEN’S SPEECH. 1867 black rod

10

The following code filters the clean\_bigrams\_1867 dataset to retain only rows where the bigram column contains any word from the pattern1 list (e.g., “riot,” “meeting,” “famine”). It uses a regular expression to match whole words, ensuring accurate filtering. After filtering, the resulting dataset, events\_1867, includes only the bigram and speechdate columns. This allows the analyst to focus on specific bigrams related to the predefined topics of interest.

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

events\_1867 <- clean\_bigrams\_1867 **%>%**

**filter**(**str\_detect**(bigram, **paste0**("**\\**b(", **paste**(pattern1, collapse = "|"), ")**\\**b"))) **%>% select**(bigram, speechdate)

**head**(events\_1867)

## bigram speechdate

## 1 sanguinary war 1867-02-05

## 2 dreadful famine 1867-02-05

## 3 late war 1867-02-05

## 4 civil war 1867-02-05

## 5 civil war 1867-02-05

## 6 china war 1867-02-08

If we look carefully at the events list so generated, we will recognize only a few of these references as actual events. Many of the bigrams including our controlled vocabulary are simply descriptions, for example, “disorderly riot.” A few are references to actual events and meetings specified by the name of an event, for instance, the “Bristol Riot” or a “Yorkshire meeting.” Far more frequent are general references to a “public meeting” or “recent meeting.” All of this is interesting, of course, because as we have seen members of parliament leaned on each other to acknowledge the public sentiment for expanding the vote.

How did references to meetings vary over time? By this time, you will have noticed that searching for a controlled vocabulary, visualization, guided reading, and deliberation about the meaning of the data is an iterative process that we engage over many rounds – not an automatic process where the analyst looks for a word like “riot” and finds simple and meaningful results. Let’s search for our key phrases and visualize them. Note that the code below uses a new command for faceted counting – group\_by() – which we will investigate later, in Chapter 4.

pattern2 = **c**('yorkshire meeting', 'clontarf meeting', 'recent meeting', 'public meeting', 'paris exhibition', 'orissa famine', 'irish famine', 'crimean war',

'civil war', 'affghan war', 'kaffir war', 'bristol riot')

top\_events\_1867 <- events\_1867 **%>%**

**filter**(bigram **%in%** pattern2) **%>%**

**group\_by**(bigram) **%>%**

**summarize**(total = **n**()) **%>%**

**top\_n**(10)

## Selecting by total

11

**head**(top\_events\_1867)

## # A tibble: 6 x 2

## bigram total

## <chr> <int>

## 1 affghan war 3

## 2 bristol riot 1

## 3 civil war 31

## 4 clontarf meeting 1

## 5 crimean war 55

## 6 irish famine 5

top\_events\_1867\_w\_speech\_metadata <- top\_events\_1867 **%>% left\_join**(events\_1867, by = "bigram") **%>%**

**mutate**(month = **month**(speechdate)) **%>%**

**count**(bigram, speechdate) **%>%**

**ungroup**()

**head**(top\_events\_1867\_w\_speech\_metadata)

## # A tibble: 6 x 3

## bigram speechdate n

## <chr> <IDate> <int>

## 1 affghan war 1867-02-26 2

## 2 affghan war 1867-07-26 1

## 3 bristol riot 1867-06-28 1

## 4 civil war 1867-02-05 5

## 5 civil war 1867-02-07 1

## 6 civil war 1867-02-14 1

**library**(viridis)

## Loading required package: viridisLite

**ggplot**(top\_events\_1867\_w\_speech\_metadata,

**aes**(x = speechdate,

y = bigram,

size = n,

color = n)) **+**

**scale\_color\_viridis**(breaks = round,

trans = "log",

option = "A",

discrete = F,

direction = 1) **+**

**scale\_x\_date**(date\_breaks = "1 month", date\_labels = "%B") **+** 12

**scale\_size\_continuous**(range = **c**(1, 10)) **+**

**geom\_point**(alpha = .5,

shape = 15) **+**

**coord\_cartesian**(clip = 'off') **+**

**theme**(legend.position = "bottom",

axis.text.x = **element\_text**(angle = 45),

plot.margin = **unit**(**c**(1, 20, 1, 1), "lines")) **+**

**guides**(shape = "none") **+**

**labs**(x = "when parliament spoke",

y = "events mentioned in the past by number of mentions", subtitle = "In the Months Leading Up to August 1867") **+ ggtitle**("Events Mentioned by Name")

Events Mentioned by Name

events mentioned in the past by number of mentions

yorkshire meeting recent meeting public meeting paris exhibition orissa famine

kaffir war

irish famine

crimean war

clontarf meeting civil war

bristol riot

affghan war

In the Months Leading Up to August 1867

February

March

April

June

August

May

July

when parliament spoke

n 5 10 15 20 n

1 23

What do we make of this timeline? It suggests a swell of public meetings referenced in parliament in the months leading up to August 1867, which received more intense attention than either the Crimean War or the

13

Paris Exhibition. Individual meetings such as those in Yorkshire or Clontarf received passing acknowledgment rather than deeper debate. Understanding more than this requires more cycles of research and reading.

Other research questions appear that the individual analyst, engaged in a process of research, would want to follow. Did references to public meetings change from January to August, while their mentions intensified? Were all of these public meetings about the vote? What was the meaning of references to a “civil war” – the English civil war or recent events in America – and was this civil war invoked in reference to 1867? What about the references to the contemporary famine in Orissa (in India) or the (presumably historical) famine in Ireland?

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

**Finding Historical Events Using a Controlled Vocabulary**

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

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

**data**("events")

eventslist <- events **%>%**

**distinct**(event\_name, scholar\_assigned\_date) **%>%**

**filter**(**!**scholar\_assigned\_date **>** 1840) **%>%**

**mutate**(event\_name = **tolower**(event\_name)) **%>%**

**select**(event\_name, scholar\_assigned\_date)

**head**(eventslist)

## # A tibble: 6 x 2

## event\_name scholar\_assigned\_date

## <chr> <dbl>

## 1 french revolution 1789

## 2 magna carta 1215

## 3 norman conquest 1066

## 4 corn laws 1815

## 5 battle of boyne 1690

## 6 glorious revolution 1688

14

Notice that events contains two columns: events by their name, and the historical dates on which those events occurred.

We will now count events mentioned in parliamentary speeches, starting with using an inner\_join(). Like its cousin, left\_join(), inner\_join() works with two data sets to make a new, combined data set. We use inner\_join() when we want to analyze shared areas of overlap between two data sets. In this case, we want to find the ngrams from annotated\_1830 that are also listed as entities in the events controlled vocabulary. We will discuss joins in greater detail in the following chapter, Chatper 3.

**data**("hansard\_1830")

**data**("debate\_metadata\_1830")

hansard\_1830 <- hansard\_1830 **%>%**

**left\_join**(debate\_metadata\_1830)

## Joining with ‘by = join\_by(sentence\_id)‘

bigrams\_1830 <- hansard\_1830 **%>%**

**unnest\_tokens**(ngram, text, token = "ngrams", n = 2)

events\_1830 <- bigrams\_1830 **%>%**

**inner\_join**(eventslist, by = **c**("ngram" = "event\_name"))

**head**(events\_1830)

## sentence\_id speechdate debate ## 1 S2V0022P0\_158 1830-02-04 ADDRESS ON THE LORDS COMMISSIONERS SPEECH. ] ## 2 S2V0022P0\_1722 1830-02-18 POUTUGAL. ] ## 3 S2V0022P0\_1729 1830-02-18 POUTUGAL. ] ## 4 S2V0022P0\_1735 1830-02-18 POUTUGAL. ] ## 5 S2V0022P0\_1833 1830-02-18 POUTUGAL. ] ## 6 S2V0022P0\_1923 1830-02-18 POUTUGAL. ] ## ngram scholar\_assigned\_date

## 1 navigation laws 1660

## 2 portuguese constitution 1822

## 3 portuguese constitution 1822

## 4 portuguese constitution 1822

## 5 portuguese constitution 1822

## 6 portuguese constitution 1822

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

In the next blocks of code, we use our new list of events to generate a visualization. Even if you do not recognize the purpose of every function or argument in the following lines of code, try to noice the ones that look familiar. Also note the comments, signaled by a hashtag (‘#’), which tells the language R to stop looking for code. We are using comments here to note to the reader what a section or line of code is doing.

15

counted\_events <- events\_1830 **%>%**

**mutate**(year = **year**(speechdate)) **%>%** *# create a new column called "year" and assign it just the year* **group\_by**(ngram, year, speechdate, scholar\_assigned\_date) **%>%**

**summarize**(n = **n**()) **%>%**

**ungroup**()

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

**head**(counted\_events)

## # A tibble: 6 x 5

## ngram year speechdate scholar\_assigned\_date n

## <chr> <dbl> <IDate> <dbl> <int>

## 1 american constitution 1831 1831-10-05 1776 1

## 2 american constitution 1831 1831-10-06 1776 2

## 3 american constitution 1832 1832-04-13 1776 2

## 4 belgian revolution 1831 1831-08-09 1830 1

## 5 belgian revolution 1831 1831-08-18 1830 1

## 6 belgian revolution 1832 1832-03-16 1830 10

cleaned\_events <- counted\_events **%>%**

**group\_by**(ngram, scholar\_assigned\_date) **%>%**

**summarize**(total = **sum**(n)) **%>%**

**ungroup**() **%>%**

**arrange**(scholar\_assigned\_date)

## ‘summarise()‘ has grouped output by ’ngram’. You can override using the

## ‘.groups‘ argument.

**head**(cleaned\_events)

## # A tibble: 6 x 3

## ngram scholar\_assigned\_date total

## <chr> <dbl> <int>

## 1 norman conquest 1066 5

## 2 twelfth century 1100 3

## 3 thirteenth century 1200 7

## 4 fourteenth century 1300 2

## 5 game laws 1389 112

## 6 fifteenth century 1400 7

16

indexed\_events <- cleaned\_events **%>%**

**mutate**(index = scholar\_assigned\_date **%/%** ((**max**(scholar\_assigned\_date) **- min**(scholar\_assigned\_date))**/**3*# group into chronological periods of equal size and select one per tranche*

**group\_by**(index) **%>%**

**filter**(total **== max**(total)) **%>%**

**ungroup**() **%>%**

**left\_join**(counted\_events, by = **c**("ngram", "scholar\_assigned\_date"))

**head**(indexed\_events)

## # A tibble: 6 x 7

## ngram scholar\_assigned\_date total index year speechdate n

## <chr> <dbl> <int> <dbl> <dbl> <IDate> <int>

## 1 norman conquest 1066 5 41 1830 1830-11-15 1

## 2 norman conquest 1066 5 41 1834 1834-03-14 1

## 3 norman conquest 1066 5 41 1834 1834-04-25 1

## 4 norman conquest 1066 5 41 1839 1839-07-12 2

## 5 twelfth century 1100 3 43 1831 1831-07-06 1

## 6 twelfth century 1100 3 43 1832 1832-02-27 1

left\_range <- **max**(events\_1830**$**speechdate) **+** 8

**library**(viridis)

**ggplot**(data = indexed\_events,

**aes**(x = speechdate,

y = scholar\_assigned\_date,

size = n,

label = **paste0**(' ', ngram, ' (', scholar\_assigned\_date, ')'),

color = n)) **+**

**scale\_color\_viridis**(breaks = round,

trans = "log",

option = "A",

discrete = F,

direction = 1) **+**

**scale\_size\_continuous**(range = **c**(1, 10)) **+**

**geom\_point**(alpha = .5,

shape = 15) **+**

**coord\_cartesian**(clip = 'off') **+**

**scale\_x\_date**(date\_breaks = "1 year", date\_labels = "%Y") **+**

**geom\_text**(data = indexed\_events **%>% group\_by**(ngram) **%>% sample\_n**(1),

show.legend = FALSE,

**aes**(color = 1,

x = left\_range,

y = scholar\_assigned\_date,

hjust = 0,

17

size = 8)) **+**

**theme**(legend.position = "bottom",

plot.margin = **unit**(**c**(1,250,50,1), unit = "pt"),

axis.text.x = **element\_text**(angle = 45, vjust = 0.5, hjust=1), axis.title.x = **element\_text**(vjust=**-**0.5)) **+**

**guides**(shape = "none") **+**

**labs**(x = "when parliament spoke", y = "events mentioned in the past") **+ ggtitle**("Events Mentioned by Name in Parliament")

18

events mentioned in the past

1800 1600 1400 1200

Events Mentioned by Name in Parliament

corn laws (1815)

french revolution (1789)

american constitution (1776)

riot act (1714)

glorious revolution (1688)

corporation act (1661)

gunpowder plot (1605)

sixteenth century (1500)

game laws (1389)

fourteenth century (1300)

thirteenth century (1200)

twelfth century (1100)

norman conquest (1066)

1830

1831

1832

1833

1834

1835

1836

1837

1838

1839

1840

when parliament spoke

n 50 100

n

1 145

19

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

How should we interpret this graph? One approach is to identify the most surprising aspects of the visu alization and explore their implications. For instance, as a British historian, I am unsurprised by frequent references in Parliament to events from the previous half-century, such as the French and American Rev olutions. Similarly, it’s expected that the Glorious Revolution was heavily referenced in the early 1830s, with mentions tapering off after 1836, and that the Riot Act was invoked annually. However, the most striking feature of the graph may be the vertical line around 1832, where references to much older historical periods—the sixteenth, thirteenth, and twelfth centuries—appear. This invocation of the distant past invites further investigation into its context and significance.

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

We continue the explanation through reading.

hansard\_1830 <- hansard\_1830 **%>%**

**mutate**(year = **year**(speechdate))

c15\_mentions <- hansard\_1830 **%>%**

**filter**(year **>** 1831 **&** year **>** 1833,

**str\_detect**(text, "fifteenth century")) **%>%**

**select**(text)

c15\_mentions**$**text

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

c14\_mentions <- hansard\_1830 **%>%**

**filter**(year **>** 1831 **&** year **>** 1833,

**str\_detect**(text, "fourteenth century")) **%>%**

**select**(text)

c14\_mentions**$**text

## [1] "Scarcely a valuable discovery had been introduced; the spinning in some cases by hand, in other20

c13\_mentions <- hansard\_1830 **%>%**

**filter**(year **>** 1831 **&** year **>** 1833,

**str\_detect**(text, "thirteenth century")) **%>%**

**select**(text)

c13\_mentions**$**text

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

c12\_mentions <- hansard\_1830 **%>%**

**filter**(year **>** 1831 **&** year **>** 1833,

**str\_detect**(text, "twelfth century")) **%>%**

**select**(text)

c12\_mentions**$**text

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

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

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

**Notes on Data Processing Order**

Throughout this chapter, we worked with large datasets and demonstrated how to process them effectively. Here’s a typical workflow for handling such data:

• Load the Dataset: Import the dataset into your R environment using appropriate functions like read\_csv(), read\_excel(), or readRDS() depending on the file format. Ensure the data is prop erly loaded and ready for further processing.

• Subset the Dataset: Narrow down the dataset to include only the data relevant to your analysis. For example, if you’re analyzing speeches by William Gladstone, you can filter rows where the speaker is Gladstone using functions like filter() from the dplyr package. Subsetting early in the process reduces the amount of data to be cleaned, saving time and computational resources.

21

• Clean the Dataset: After subsetting, clean the dataset to prepare it for analysis. Cleaning typically involves handling missing values, removing stop words, and normalizing text (e.g., converting text to lowercase). Use functions like mutate() from dplyr, or string processing functions like str\_to\_lower() from the stringr package.

• Analyze the Dataset: Now that you have just the needed data, perform your analysis by applying metrics.

Why Process in This Order?

While it’s possible to clean the dataset before subsetting, doing so can lead to inefficiencies:

• Wasted Effort: Cleaning the entire dataset takes extra time, even for rows or columns that will later be excluded during subsetting.

• Memory Limitations: Large datasets may exceed your system’s memory capacity during cleaning. Subsetting first reduces the size of the dataset, making cleaning more manageable.

By following this order—Load -> Subset -> Clean-> Analyze—analysts can optimize their workflow and process larger amounts of data with less computational resources, a consideration that may be especially useful when analyzing change over decades or centuries of data.

**Exercises**

1) In the code above, we searched for 5-word phrases that reference “the people.” Use the code to find two-word phrases in 1830, 1840, 1850, and 1860 that reference “meeting.” Write a paragraph, quoting at least 10 phrases from each decade. For each of the phrases mentioned more than five times, describe what prejudices, positive or negative, the phrase encapsulates. Use your evidence to answer the question: how did parliamentary references to meetings change between 1830 and 1860?

Tips: Instead of joining every decade together, process the decades separately to avoiding using all your computer’s resources.

2) In the code above, we created two kinds of timelines, one for 1867 using a controlled vocabulary of words such as “meeting” and the other for the 1830s using the list of historical events, eventlist. Make two more timelines. Apply the controlled vocabulary to the 1830s, and search for the items in eventlist in the 1860s. Use the code above to graph the results. Keep in mind that you will need to adjust your controlled vocabulary to what you find in the data.

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

22

**Practicing with AI**

Throughout this book, we emphasize the importance of iterative research, such as refining one’s understand ing by revisiting the original records, or by questioning one’s assumptions through interrogative questions that build upon one-another.

AI chatbots can support iterative research, too, by supporting an interactive bnrainstorming process. An alysts can ask questions, review responses, and adjust their queries, engaging in a cycle of exploration and refinement. When used this way, an AI chatbot should be guided by the analyst to generate focused, relevant responses. The analyst must also critique the chatbot’s output to ensure it is accurate and useful. This approach allows analysts to use AI chatbots as a tool for inquiry while maintaining control over the research process.

In the following example, we use our queries from earlier in the chapter to guide the AI chatbot in helping us to brainstorm additional events that we might want to explore. Specifically, we ask the chatbot to generate additional events that might be of interest to someone studying language relating to war in the 1860 Hansard debates. We will guide the chatbot’s response by giving it an example of the types of events that interest us taken from the eventslist dataset.

The template we provide the chatbot (e.g. “return your answer in this fomrat. . . ”) instructs it to return its response in the specified format.

User Prompt

I am providing you with a list of events followed by their year. The purpose of this list is to analyze key events that were mentioned in the 1860 Hansard debates. Name some different events that might be relevant based on the list I provide you. Don’t mention the same as provided by the list. Also provide a reason for why the events are relevant.

Return your answer in this format:

**Event:**

**Reason:**

These are the events from the list: (copy/paste contents of eventslist)

23

ChatGPT Response

**Event: Abolition of the Slave Trade (1807)**

**Reason:** The abolition of the slave trade would be highly relevant in discussions about moral and economic reforms during the 1860 Hansard debates, particularly in the context of human rights and British colonial policies.

**Event: Reform Act (1832)**

**Reason:** This act, which expanded suffrage and restructured parliamentary representation, would likely influence discussions on electoral reform and representation in 1860.

**Event: Crimean War (1853–1856)**

**Reason:** The Crimean War had significant impacts on British foreign policy, military reforms, and public health improvements, all of which could be relevant topics of debate in 1860. **Event: Irish Potato Famine (1845–1852)**

**Reason:** The famine’s aftermath would remain a pressing issue, influencing debates about Irish governance, economic policies, and relief efforts.

**Event: Great Exhibition (1851)**

**Reason:** As a symbol of industrial progress and imperial power, the Great Exhibition could be discussed in terms of economic innovation and Britain’s global role in the mid-19th century. **Event: Indian Rebellion (1857)**

**Reason:** The rebellion was a turning point in British imperial governance, leading to significant debates about military strategy, colonial administration, and the justification of imperial rule. **Event: Public Health Act (1848)**

**Reason:** This act, which addressed urban sanitation and public health, would be relevant to discus sions about improving living conditions and public health policies in 1860.

**Event: Establishment of the Penny Post (1840)**

**Reason:** The reform in postal services, increasing communication efficiency and accessibility, could relate to debates on economic and infrastructural development.

Our initial prompt to the AI chatbot successfully guided it to identify and return events similar to the query. However, when we provided the chatbot with the entire contents of eventslist, the results lacked precision and specificity. This outcome suggests that more targeted guidance is necessary to focus on the specific characteristics of interest. This suggests that without more specific guidance, the chatbot may not generate results based on the specific characteristics of the events we are interested in. To tailor the results to a specific research question, we will refine the prompt further by narrowing in on the events we are most interested in from eventslist and instructing the chatbot to return like-events, in this case, other names of meetings that occurred in the 1860s.

24

User Prompt

I am providing you with a list of events followed by their year. The purpose of this list is to analyze key events that were mentioned in the 1860 Hansard debates. Name some different events that might be relevant based on the list I provide you. Don’t mention the same as provided by the list. Also provide a reason for why the events are relevant based on the list I provide.

Return your answer like this:

**Event:**

**Date:**

**Reason:**

These are the events from the list: ’greek war’, ’napoleonic wars.’

ChatGPT Response

Here are some different events that might be relevant based on your list:

**Event: Chartist Movements**

**AI Assigned Date:** 1838–1850

**Reason:** The Chartist movements, with their organized public gatherings for political reform, directly relate to the mentioned events like ’public meeting’ and ’recent meeting,’ illustrating how assemblies have historically been used to drive societal change.

**Event: Anti-Corn Law League Meetings**

**AI Assigned Date:** 1839–1846

**Reason:** These meetings share a thematic connection to ’public meeting’ by demonstrating how public gatherings have been pivotal in shaping discourse around economic reform, similar to those discussed in the 1860 debates.

**Event: Hyde Park Demonstrations**

**AI Assigned Date:** 1855–1860

**Reason:** Hyde Park demonstrations resonate with ’public meeting’ and ’recent meeting,’ showing how such gatherings functioned as platforms for political and social advocacy, reflecting the period’s reliance on public discourse for progress.

**Event: Meetings on the Reform Act of 1832**

**AI Assigned Date:** 1831–1832

**Reason:** These meetings are akin to ’yorkshire meeting’ and ’clontarf meeting,’ showcasing historical instances where regional or national public assemblies catalyzed significant legislative reforms, serving as a possible reference point in 1860.

**Event: Fenian Meetings**

**AI Assigned Date:** 1858–1860

**Reason:** While slightly later, Fenian meetings parallel ’clontarf meeting’ by highlighting Irish politi cal concerns and illustrating how public gatherings were used to address issues of representation and governance.

Now that we have a more specific list of events that aligns with our interests, we can use them in our analysis. To use our new events, we will create a dataframe for them. We can prompt the chatbot to return the needed code. Given that we are very familiar with creating dataframes at this point, we can also proof the chatbot’s results and ensure they match our desired output.

25

User Prompt

Return this text as a dataframe in R with two columns: a column for the event, named event\_name and a column for the date, named AI Assigned Date. Write the events in lower case.

ChatGPT Response

Here is the R code to create the requested dataframe:

# Create the dataframe

events\_df <- data.frame(

event\_name = c(

"crimean war",

"russo-turkish war",

"congress of vienna",

"siege of acre",

"battle of navarino",

"schleswig-holstein wars"

),

AI\_Assigned\_Date = c(

"1853–1856",

"1828–1829",

"1814–1815",

"1799",

"1827",

"1848–1852"

)

)

# Print the dataframe

print(events\_data)

This will create a dataframe in R with two columns: event\_name (containing the event names in lowercase) and AI Assigned Date.

*# Create the dataframe*

events\_df <- **data.frame**(

event\_name = **c**(

"crimean war",

"russo-turkish war",

"congress of vienna",

"siege of acre",

"battle of navarino",

"schleswig-holstein wars"

),

AI\_Assigned\_Date = **c**(

26

"1853–1856",

"1828–1829",

"1814–1815",

"1799",

"1827",

"1848–1852"

)

)

*# Print the dataframe*

**print**(events\_df)

## event\_name AI\_Assigned\_Date

## 1 crimean war 1853–1856

## 2 russo-turkish war 1828–1829

## 3 congress of vienna 1814–1815

## 4 siege of acre 1799

## 5 battle of navarino 1827

## 6 schleswig-holstein wars 1848–1852

**data**("hansard\_1860")

**data**("debate\_metadata\_1860")

hansard\_1860 <- hansard\_1860 **%>%**

**left\_join**(debate\_metadata\_1860)

## Joining with ‘by = join\_by(sentence\_id)‘

The following code filters the hansard\_1860 dataset to include rows where the text column matches any event name in events\_df$event\_name, using case-insensitive matching. It then adds a new column, matched\_event, which contains the specific matched event name from the text. The result is a filtered dataframe with an additional column identifying the matched event for each row.

*# Match with each event and*

*# add a column to capture the matched event*

new\_matched\_events <- hansard\_1860 **%>%**

**filter**(**str\_detect**(

text,

**regex**(**paste0**("**\\**b(", **paste**(events\_df**$**event\_name, collapse = "|"), ")**\\**b"), ignore\_case = TRUE))) **%>mutate**(

matched\_event = **str\_extract**(

text,

**regex**(**paste0**("**\\**b(", **paste**(events\_df**$**event\_name, collapse = "|"), ")**\\**b"), ignore\_case = TRUE)))

*# Print the resulting dataframe*

*# Display only the first 10 rows*

**print**(**head**(new\_matched\_events, 10))

27

## sentence\_id

## 1 S3V0156P0\_3555

## 2 S3V0156P0\_8154

## 3 S3V0156P0\_8400

## 4 S3V0156P0\_10788

## 5 S3V0156P0\_14522

## 6 S3V0156P0\_16926

## 7 S3V0156P0\_18210

## 8 S3V0156P0\_18686

## 9 S3V0157P0\_2701

## 10 S3V0157P0\_3616

##

## 1 As to the noble Earl’s argument against the abandonment of a prohibitive duty on coal in the## 2 The Crimean war, while it created an enormous extra demand f## 3 Member for Sunderland himself, who, in a work which he had recently published, made mention of th## 4 With regard to the claims of officers of the Land Transport Corps, a Committee of tha## 5

## 6 I## 7 During the last two years there ## 8 I have no doubt that the improved health## 9

## 10

## speechdate debate matched\_event

## 1 1860-02-20 QUESTION. Crimean war

## 2 1860-01-31 COMMITTEE MOVED FOR. Crimean war

## 3 1860-01-31 COMMITTEE MOVED FOR. Crimean war

## 4 1860-02-03 OBSERVATIONS. Crimean War

## 5 1860-02-10 COMMITTEE. Crimean war

## 6 1860-02-16 RESOLUTION. Crimean war

## 7 1860-02-17 THE MILITIA. —QUESTION. Crimean war

## 8 1860-02-17 SUPPLY. —ARMY ESTIMATES. Crimean war

## 9 1860-03-23 MUTINY BILL. —SECOND READING. Crimean war

## 10 1860-03-30 QUESTION. Crimean war

The results demonstrate that not all of the events were mentioned verbatim in the Hansard corpus; how ever, we successfully matched some events, including the Crimean War, Congress of Vienna, and Battle of Navarino. This highlights an iterative process, where analysts may need to refine their search by identifying additional or alternative words, either manually or through further interaction with the AI chatbot.

Another important note is that the AI chatbot did not automatically handle certain formatting issues in the text, such as inconsistent capitalization. For more consistent results that could then be visualized, the analyst should implement additional preprocessing steps, including cleaning the text data for case consistency and resolving other formatting discrepancies.

*# Count the occurrences of each matched event*

event\_counts <- new\_matched\_events **%>%**

**count**(matched\_event, name = "count") **%>%**

**arrange**(**desc**(count))

28

*# Print the result*

**print**(event\_counts)

## matched\_event count

## 1 Crimean war 360

## 2 Crimean War 143

## 3 Congress of Vienna 42

## 4 battle of Navarino 5

29