STAT325: Analysis of Frankenstein

Justin Papagelis (jpapagelis24@amherst.edu)

2024-03-15

Table of contents

Introduction

Frankenstein; or, The Modern Prometheus (Shelley 1818), written by Mary Shelley in 1818, is a groundbreaking work in the literary world that interweaves elements of Gothic literature with early ideas of science fiction. The novel tells the cautionary tale of Victor Frankenstein, a scientist who creates a sentient being, only to reject it which leads to tragic consequences for both the creator and the creature. The novel explores themes of ambition, responsibility, and the search for identity. *Frankenstein* has become an important dialogue about the limits of scientific exploration, the responsibilities of creators, and the societal implications of isolation and rejection. It raises essential questions about the nature of humanity, the pursuit of knowledge, and the consequences of ambition unhindered by ethical considerations. Additionally, *Frankenstein* has had an enduring influence on popular culture and media. The narrative unfolds through the perspectives of three distinct narrators: Captain Robert Walton, Victor Frankenstein, and The Creature. The use of multiple narrators allows Shelley to explore different viewpoints and layer the narrative with complex emotions and ethical questions, providing a rich tapestry of human experience and moral contemplation. Each narrative reflects each individual's experiences, biases, and emotional states, contributing to a many different perspectives of the same story. For this analysis, I would like to see if the sections told by different narrators vary from each other. A short example of the first couple of lines of each narrator's first section is shown in Table 1.

```
get_first_n_lines <- function(section_number, letter_or_chapter, n) {
  # Filter data for the specific section and section type

  section_data <- Frankenstein |>
    filter(section == section_number, section_type == letter_or_chapter)

  # Extract the first n lines of text, if there are at least n lines available
  if (nrow(section_data) >= n) {
    lines <- section_data$text[1:n]
  } else {
    lines <- section_data$text[1:nrow(section_data)]
  }
  combined_lines <- paste(lines, collapse = " ")
  combined_lines <- paste0(combined_lines, " ...")
  return(combined_lines)
}
```

```
example_text <- tibble(
  Narrator = c("Captain Robert Walton", "Victor Frankenstein", "The Creature"),
  `Example Text` = c(get_first_n_lines(1, "letter", 15), # Walton's first POV
                     get_first_n_lines(1, "chapter", 15), # Victor's first POV
                     get_first_n_lines(11, "chapter", 15)) # Creature's first POV
)
```

```
# Create the table
example_text |>
  kableExtra::kable(booktabs = TRUE, format = "latex",
                    linesep = "\\addlinespace",
                    caption = "Example Narrative Sections in
                    Frankenstein\\label{example_text}") |>
  kableExtra::column_spec(1, width = "3cm") |>
  kableExtra::column_spec(2, width = "12cm") |>
  kableExtra::kable_styling(latex_options = "scale_down")
```

Methods

The novel is told using a framing device which is basically a story within a story within a story. More specifically, the narrators for each section are as follows:

Table 1: Example Narrative Sections in Frankenstein

Narrator	Example Text
Captain Robert Walton	Letter 1 _To Mrs. Saville, England._ St. Petersburgh, Dec. 11th, 17—. You will rejoice to hear that no disaster has accompanied the commencement of an enterprise which you have regarded with such evil forebodings. I arrived here yesterday, and my first task is to assure my dear sister of my welfare and increasing confidence in the success of my undertaking. I am already far north of London, and as I walk in the streets of ...
Victor Frankenstein	Chapter 1 I am by birth a Genevese, and my family is one of the most distinguished of that republic. My ancestors had been for many years counsellors and syndics, and my father had filled several public situations with honour and reputation. He was respected by all who knew him for his integrity and indefatigable attention to public business. He passed his younger days perpetually occupied by the affairs of his country; a variety of circumstances had prevented his marrying early, nor was it until the decline of life that he became a husband and the father of a family. As the circumstances of his marriage illustrate his character, I cannot refrain from relating them. One of his most intimate friends was a ...
The Creature	Chapter 11 “It is with considerable difficulty that I remember the original era of my being; all the events of that period appear confused and indistinct. A strange multiplicity of sensations seized me, and I saw, felt, heard, and smelt at the same time; and it was, indeed, a long time before I learned to distinguish between the operations of my various senses. By degrees, I remember, a stronger light pressed upon my nerves, so that I was obliged to shut my eyes. Darkness then came over me and troubled me, but hardly had I felt this when, by opening my eyes, as I now suppose, the light poured in upon me again. I walked and, I believe, descended, but I presently found a great alteration in my sensations. Before, dark and opaque bodies had surrounded me, impervious to my touch or sight; but I now found that I could wander on at liberty, with ...

- Walton: *Letters 1-4*
- Frankenstein: *Chapters 1-10*
- The Creature: *Chapters 11-16*
- Frankenstein: *Chapters 17-24.5*
- Walton: *Rest of Chapter 24*

For this part of my analysis, I use Latent Dirichlet Allocation (LDA) which was first used in machine learning by David Blei, Andrew Ng and Michael I. Jordan (Blei, Ng, and Jordan 2003). It is a method of topic modelling to try that I will use to see if there is evidence of the framing device within the novel. In other words, I would like to see if LDA can detect the parts of the story that are written by the same narrator. To do this in R, I will utilize methods explained in "tidytext" (Silge and Robinson 2016), as well as the `topicmodels` package (Grün and Hornik 2024) and the `stringr` package (Wickham 2023)

Initial Wrangling

To begin, the novel is pre-processed to remove common stop words, numbers, and other miscellaneous words. To create a unique label for each chapter, I combined the section and the section type as well as removed "section 0" which was the table of contents. To further wrangle the data, I split Chapter 24 into two parts, each corresponding to one narrator. I also tokenized the novel which splits the novel into individual words. See Table 2 for an example of the tokenized data.

Next, I created the document-word matrix and an LDA model using a value of `k=`12 which means that the LDA model will isolate topics. I decided to separate the novel into 12 topics because since the novel is long and there are many plot lines, this greater number of topics will help identify distinct topics.

```r
# Pre-processing
other_stop_words <- tibble(
     word = paste(c(0:24, "Chapter", "11th", "_to")))

section_names <- c("letter 1", "letter 2", "letter 3", "letter 4", "chapter 1",
              "chapter 2", "chapter 3", "chapter 4", "chapter 5", "chapter 6",
              "chapter 7", "chapter 8", "chapter 9", "chapter 10", "chapter 11",
              "chapter 12", "chapter 13", "chapter 14", "chapter 15",
              "chapter 16", "chapter 17", "chapter 18", "chapter 19",
              "chapter 20", "chapter 21", "chapter 22", "chapter 23",
              "chapter 24", "letter 24.5")
```

```r
narrators <-
  tibble(section_names) |>
  mutate(
    section_number = readr::parse_number(section_names),
    narrator = case_when(
      str_detect(section_names, "letter") ~ "Captain Walton",
      str_detect(section_names, "chapter")
      & section_number <= 10 ~ "Victor Frankenstein",
      str_detect(section_names, "chapter")
      & between(section_number, 11, 16) ~ "The Creature",
      str_detect(section_names, "chapter")
      & section_number >= 17 ~ "Victor Frankenstein"
    )
  ) |>
  rename(document = "section_names")

# Clean and tokenize novel
# note: line 6854 is where Walton's POV starts back up in chapter 24
Frankenstein_LDA <- Frankenstein |>
  mutate(section = ifelse(line >= 6854, 24.5, section),
         section_type = ifelse(line >= 6854, "letter", section_type)) |>
  filter(section != 0) |>
  mutate(section_label = paste(section_type,section)) |>
  select(-section,-section_type) |>
  unnest_tokens(word, text) |>
  anti_join(stop_words) |>
  anti_join(other_stop_words) |>
  mutate(section_label = factor(section_label, levels = section_names))

Frankenstein_LDA |>
  filter(section_label %in% "chapter 1") |>
  head(20) |>
  kableExtra::kable(booktabs = TRUE, col.names = c("Line", "Section", "Word"),
                    caption = "Example of Tokenized Novel\\label{token}")
```

```r
# Create the document-word matrix
dtm <- Frankenstein_LDA |>
  count(section_label, word) |>
  cast_dtm(section_label, word, n)
```

Table 2: Example of Tokenized Novel

Line	Section	Word
623	chapter 1	chapter
626	chapter 1	birth
626	chapter 1	genevese
626	chapter 1	family
627	chapter 1	distinguished
627	chapter 1	republic
627	chapter 1	ancestors
628	chapter 1	counsellors
628	chapter 1	syndics
628	chapter 1	father
628	chapter 1	filled
628	chapter 1	public
629	chapter 1	situations
629	chapter 1	honour
629	chapter 1	reputation
629	chapter 1	respected
630	chapter 1	integrity
630	chapter 1	indefatigable
630	chapter 1	attention
630	chapter 1	public

Table 3: Beta Values for ”Death”

Topic	Term	Beta
1	death	0.0000
2	death	0.0029
3	death	0.0033
4	death	0.0043
5	death	0.0051
6	death	0.0000
7	death	0.0008
8	death	0.0020
9	death	0.0035
10	death	0.0056
11	death	0.0026
12	death	0.0011

```r
# Create LDA model using value of k and seed
lda_model <- LDA(dtm, k = number_of_topics, control = list(seed = 1))
```

Beta Analysis

First, I looked at the per-topic-per-word probabilities to extract the top words that are used in each of the topics. An example of the beta values for the word “death” are shown in Table 3. As you can see, Topic 5 has a much larger beta value than the other topics which suggests that “death” is a word more associated with Topic 5. Then I created a graphic using `ggplot` (Wickham 2016) to show the most common words in each of the topics. The graphic is shown below in Figure 1.

```r
# Per-topic-per-word probabilities
topics_beta <- tidy(lda_model, matrix = "beta")

topics_beta |>
  filter(term == "death")   |>
  kableExtra::kable(booktabs = TRUE, col.names = c("Topic", "Term", "Beta"),
                    caption = "Beta Values for \"Death\"\\label{beta_death}",
                    digits = 4)
```

```r
# Per-topic-per-word probabilities
topics_beta <- tidy(lda_model, matrix = "beta")

# Get the top terms
top_terms <- topics_beta |>
  group_by(topic) |>
  slice_max(beta, n = 5) |>
  ungroup() |>
  arrange(topic, -beta)

topic_names <- c("1: \"The Creature's birth\"", "2: \"Frankenstein's Studies\"",
                 "3: \"Justine's Trial\"", "4: \"Walton in Antarctica\"",
                 "5: \"Frankenstein's Guilt\"",
                 "6: \"Frankenstein on the Run\"",
                 "7: \"Frankenstein at Home\"", "8: \"Creating the Creature\"",
                 "9: \"Frankenstein in Antarctica\"",
                 "10: \"Frankenstein's Love\"", "11: \"The Creature Learning\"",
                 "12: \"The Creature Betrayed\"")

# Create visual
top_terms |>
  mutate(term = reorder_within(term, beta, topic),
         topic = factor(topic, levels = 1:12, labels = topic_names)) |>
  ggplot(aes(beta, term, fill = factor(topic, levels = topic_names))) +
  theme_minimal() +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free", ncol = 2) +
  scale_y_reordered() +
  labs(x = "Beta-value", y = "Words", fill = "Topic",
       title = "Most Common Words by Topic") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

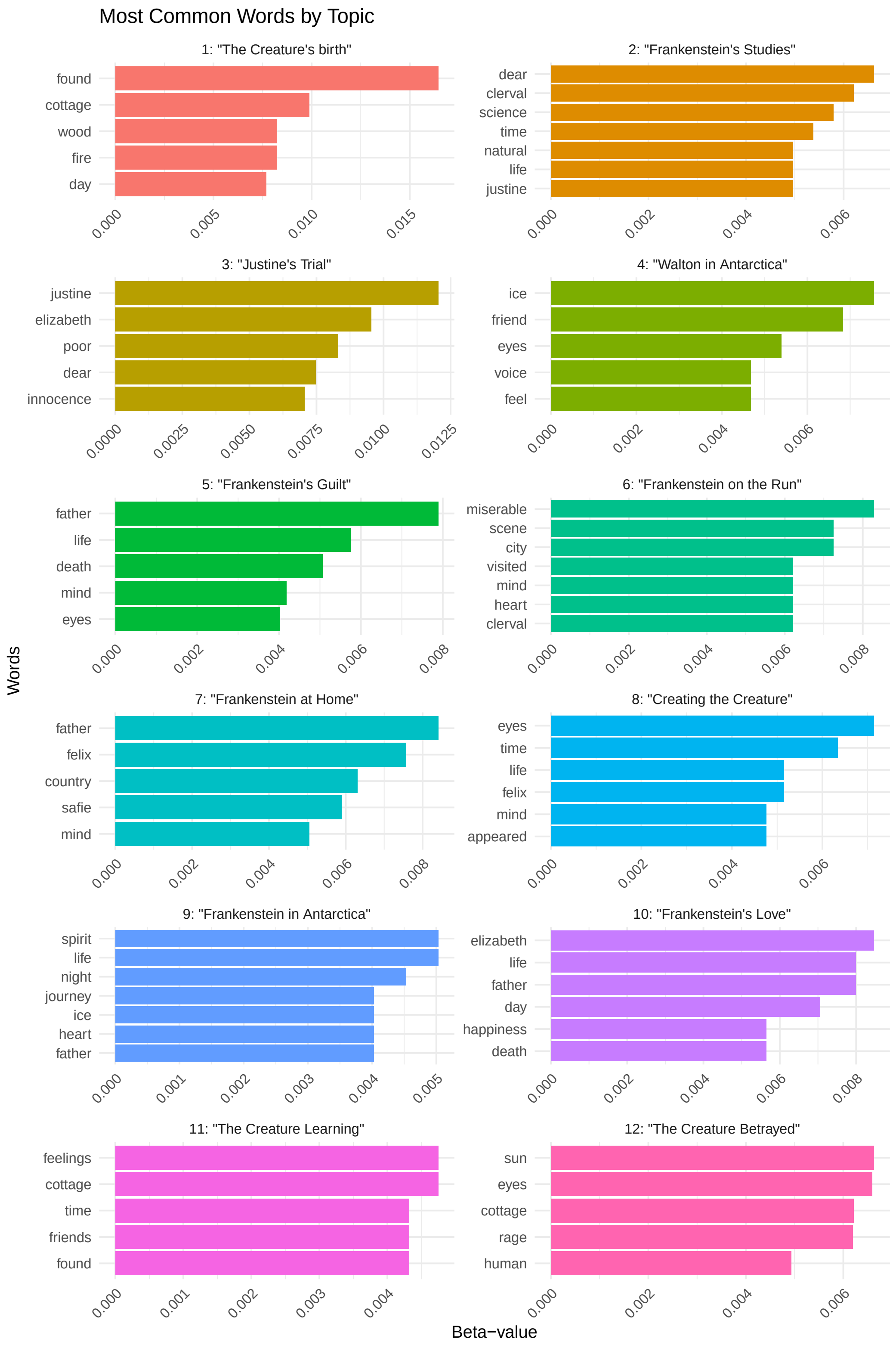

Figure 1: Figure of most common words in each topic

In particular, I chose to focus on three of the topics: 1: The Creature's Birth, 4: Walton in Antarctica, and 10: Frankenstein's Love.

Topic 1 shows that the most common words are found, cottage, wood, fire, and day. This topic gives very simple words that seem to be when The Creature is telling his story and thus has a limited knowledge of the world and talks of basic utilities and needs. He is talking about the day that he was born and first experienced the world.

Topic 4's most common words were ice, friend, eyes, voice, feel. This most likely represents Captain Robert Walton who was on an expedition to Antarctica when he met Victor Frankenstein and the The Creature. These words all seem ominous and make sense with all the death that occurs in the frozen landscape at the culmination of the novel.

Topic 10's most common words were Elizabeth, life, father, day, happiness, and death. This seems to be Victor Frankenstein's narration because one of his main devotions in life was towards his adopted sister, Elizabeth. His father was also an important figure in his life. He also "gave life" to his creation: The Creature.

Gamma Analysis

After, that, I examined the per-document-per-topic probabilities ("gamma") which gives the estimated proportion of words from that document that are generated from that topic. The visual is shown in Figure 2. For example, this model predicts that a large portion of the words from Chapter 11 and Chapter 12 are generated from Topic 1. Looking back, we can see that this topic (The Creature's Birth) corresponded to the beginning of The Creature's life/narration which is true since The Creature's narration spans from Chapter 11 to Chapter 16. Using our other example chapters, we can see that Letter 4 and Letter 24.5 both are estimated to be generated from Topic 4 (Walton in Antarctica) which corresponds to Captain Robert Walton's narration! And finally, Chapter 4 and Chapter 22 are estimated to be generated from Topic 10 (Frankenstein's Love) which we had decided was Victor Frankenstein's narration. See Table 4 for the gamma values used in the examples above.

```r
# Per-document-per-topic probabilities
topics_gamma <- tidy(lda_model, matrix = "gamma") |>
  mutate(document = factor(document, levels = section_names))

gamma_with_narrator <- topics_gamma |>
  pivot_wider(names_from = topic, values_from = gamma) |>
  inner_join(narrators, by = "document") |>
  pivot_longer(cols = 2:13, names_to = "topic", values_to = "gamma") |>
  mutate(document = factor(document, levels = section_names))

gamma_with_narrator <- gamma_with_narrator |>
```

Table 4: Gamma Values for Example Chapters

Section	1	2	3	4	5	6	7	8	9	10	11	12
letter 4	0	0	0	1	0	0	0	0	0	0	0	0
chapter 4	0	0	0	0	0	0	0	0	0	1	0	0
chapter 11	1	0	0	0	0	0	0	0	0	0	0	0
chapter 12	1	0	0	0	0	0	0	0	0	0	0	0
chapter 22	0	0	0	0	0	0	0	0	0	1	0	0
letter 24.5	0	0	0	1	0	0	0	0	0	0	0	0

```
  mutate(topic_POV = case_when(topic %in% c(4) ~ "Walton Topic",
      topic %in% c(2,3,5,6,7,9,10)  ~ "Frankenstein Topic",
      topic %in% c(1,8,11,12)  ~ "Creature Topic"))

topics_gamma |>
  filter(document %in% c("chapter 11", "chapter 12", "letter 4", "letter 24.5",
                         "chapter 4", "chapter 22")) |>
  pivot_wider(names_from = topic, values_from = gamma) |>
  kableExtra::kable(booktabs = TRUE, col.names = c("Section", 1:12),
              caption = "Gamma Values for Example Chapters\\label{gamma_chap}",
                    digits = 2)
```

```
# Create visual
walton_pov <- gamma_with_narrator |>
  mutate(topic = factor(topic, levels = 1:12, labels = topic_names)) |>
  filter(topic_POV == "Walton Topic") |>
  ggplot(aes(x = document, y = gamma, group = topic, color = topic)) +
  geom_rect(xmin = 0, xmax = 4, ymin = 0, ymax = 2, alpha = 0.01,
            fill = 'lightblue', inherit.aes = FALSE) +
  geom_rect(xmin = 28.5, xmax = 29, ymin = 0, ymax = 2, alpha = 0.01,
            fill = 'lightblue', inherit.aes = FALSE) +
  geom_point() + geom_smooth(se = FALSE, span = 0.5, method = 'loess',
                             formula = 'y - x') + theme_minimal() +
  labs(x = "Section", y = "Topic Proportion", fill = "Topic",
       title = "Gamma Topic Distribution Across Frankenstein With Walton's
       Topics Highlighted") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom") + coord_cartesian(ylim = c(0, 1))
```

```r
frankenstein_pov <- gamma_with_narrator |>
  mutate(topic = factor(topic, levels = 1:12, labels = topic_names)) |>
  filter(topic_POV == "Frankenstein Topic") |>
  ggplot(aes(x = document, y = gamma, group = topic, color = topic)) +
  geom_rect(xmin = 4, xmax = 14, ymin = 0, ymax = 2, alpha = 0.01,
            fill = 'pink', inherit.aes = FALSE) +
  geom_rect(xmin = 21, xmax = 28.5, ymin = 0, ymax = 2, alpha = 0.01,
            fill = 'pink', inherit.aes = FALSE) +
  geom_point() + geom_smooth(se = FALSE, span = 0.5, method = 'loess',
                             formula = 'y - x') + theme_minimal() +
  labs(x = "Section", y = "Topic Proportion", fill = "Topic",
       title = "Gamma Topic Distribution Across Frankenstein With
       Frankenstein's Topics Highlighted") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom") + coord_cartesian(ylim = c(0, 1))

creature_pov <- gamma_with_narrator |>
  mutate(topic = factor(topic, levels = 1:12, labels = topic_names)) |>
  filter(topic_POV == "Creature Topic") |>
  ggplot(aes(x = document, y = gamma, group = topic, color = topic)) +
  geom_rect(xmin = 14, xmax = 21, ymin = 0, ymax = 2, alpha = 0.01,
            fill = 'lightgreen', inherit.aes = FALSE) +
  geom_point() + geom_smooth(se = FALSE, span = 0.5, method = 'loess',
                             formula = 'y - x') + theme_minimal() +
  labs(x = "Section", y = "Topic Proportion", fill = "Topic",
       title = "Gamma Topic Distribution Across Frankenstein With The
       Creature's Topics Highlighted") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom") + coord_cartesian(ylim = c(0, 1))

gridExtra::grid.arrange(walton_pov, frankenstein_pov, creature_pov, ncol = 1)
```

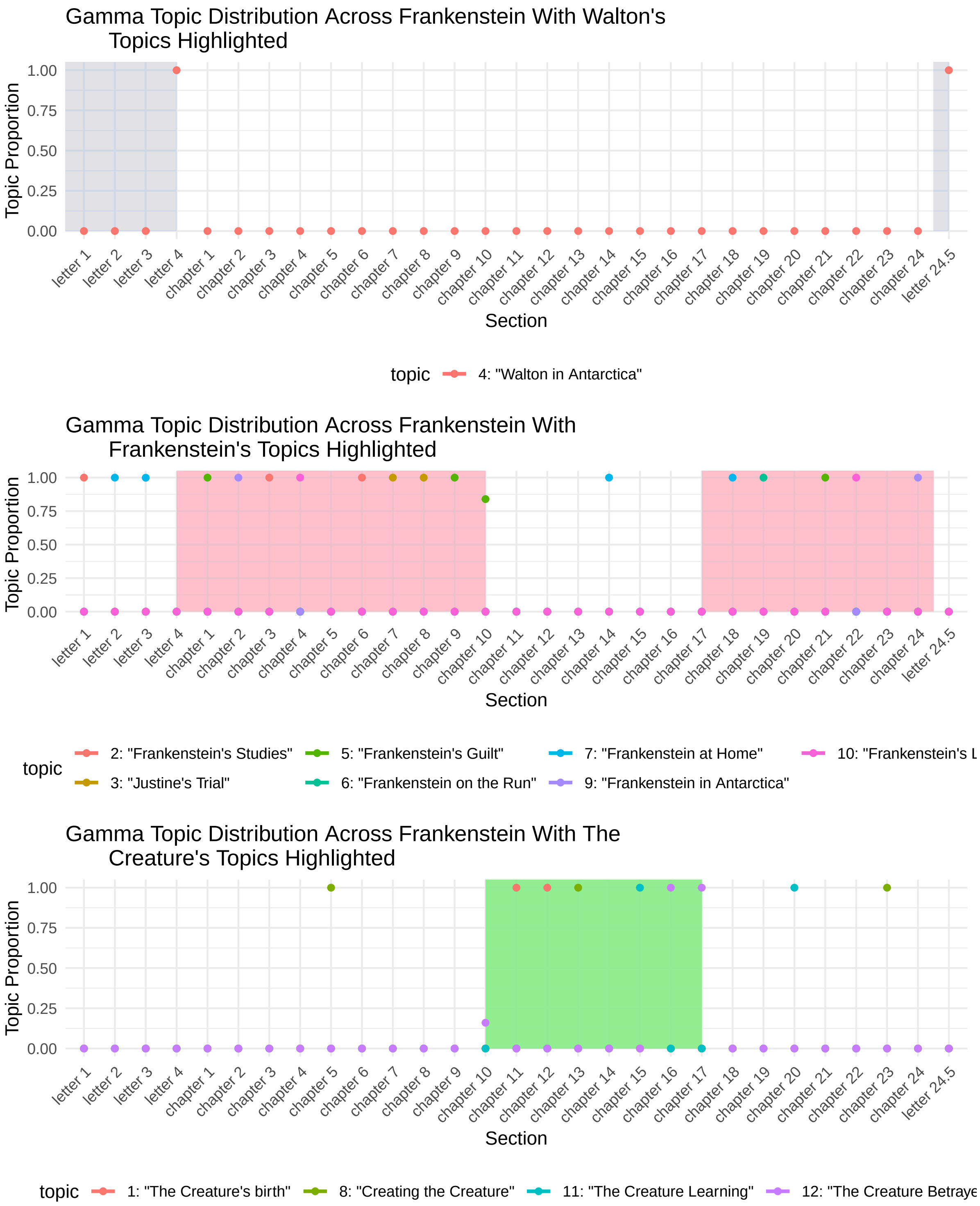

Figure 2: Per-section-per-topic probabilities

Shiny App

Additionally, I created a Shiny app that can be used to further analyze the sentence structure within *Frankenstein*. To accomplish this, we used the `cleanNLP` package (Arnold 2017) as well as the `plotly` package (Inc. 2015).

In particular, you can see the distribution of different parts of speech by chapter as well as select a certain chapter and see the frequencies of the parts of speech in specific sentences. Click here to open the app: Frankenstein Shiny App.

If we look at the "Parts of Speech" Tab, we can see an interesting distribution of proper nouns over the chapters and you can accurately see where the narrator sections should be broken up. There seem to be spikes in the center of most of the narrator's point of view sections. The rest of the parts of speech seem to be similar across different narrator sections.

Additionally, looking at sentence structure, there does not seem to be any distinguishing features that set the different narrator sections apart. The longest sentence is told by Victor Frankenstein and contain 159 words.

Conclusion

Overall, the LDA topic modeling provided insights into the narrative structure within *Frankenstein* and was able to identify several elements of the framing device employed throughout the text. We were semi-successful in seeing how the topics correlated with certain narrator sections. Although the model was not perfect as there were numerous topics with words dispersed across the entire book, refining the parameters could enhance its precision and effectiveness in distinguishing distinct narrative elements.

References

Arnold, Taylor. 2017. "A Tidy Data Model for Natural Language Processing Using cleanNLP." *The R Journal* 9 (2): 1–20. https://journal.r-project.org/archive/2017/RJ-2017-035/index.html.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. "Latent Dirichlet Allocation." *J. Mach. Learn. Res.* 3 (null): 993–1022.

Grün, Bettina, and Kurt Hornik. 2024. *Topicmodels: Topic Models*. https://CRAN.R-project.org/package=topicmodels.

Inc., Plotly Technologies. 2015. "Collaborative Data Science." Montreal, QC: Plotly Technologies Inc. 2015. https://plot.ly.

Shelley, Mary Wollstonecraft. 1818. *Frankenstein; or, the Modern Prometheus*.

Silge, Julia, and David Robinson. 2016. "Tidytext: Text Mining and Analysis Using Tidy Data Principles in r." *JOSS* 1 (3). https://doi.org/10.21105/joss.00037.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

———. 2023. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://stringr.tidyverse.org.