STAT325: LDA Analysis of Frankenstein

Justin Papagelis (jpapagelis24@amherst.edu)

2024-03-04

Frankenstein; or, The Modern Prometheus was written by Mary Shelley in 1818. The novel tells the story of Victor Frankenstein's journey creating a sapient being and the repercussions that haunt him. There are three different narrators that appear within the novel: Captain Robert Walton, Victor Frankenstein, and The Monster. I am interested to see if the sections told by different narrators vary from each other.

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:

- 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 in 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 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. Then, I created the document-word matrix and an LDA model using a value of `k=12` which means that the LDA model will isolate 12 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 distict 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")



narrators <- tibble (
  section_names
)

narrators <- narrators |>
  mutate(narrator = case_when(
          str_detect(section_names, "letter") ~ "Captain Walton",
          str_detect(section_names, "chapter") & as.numeric(str_extract(section_names, "\\
          str_detect(section_names, "chapter") & between(as.numeric(str_extract(section_na
          str_detect(section_names, "chapter") & as.numeric(str_extract(section_names, "\\
  rename("document" = "section_names")

# Clean and tokenize novel
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))

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

```
  cast_dtm(section_label, word, n)

# Create LDA model using value of k and seed
lda_model <- LDA(dtm, k = 12, control = list(seed = 1))
```

First, I looked at the per-topic-per-word probabilities to extract the top words that are used in each of the topics. Then I created a model to show the most common words in each of the topics.

```
# 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 Tri
                 "4: Walton in Antarctica", "5:Frankenstein's Guilt", "6: Frankenstein on
                 "7: Frankenstein at Home", "8: Creating the Creature", "9: Frankenstein i
                 "10: Frankenstein's Love", "11: The Creature Learning", "12: The Creature


# 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))
```

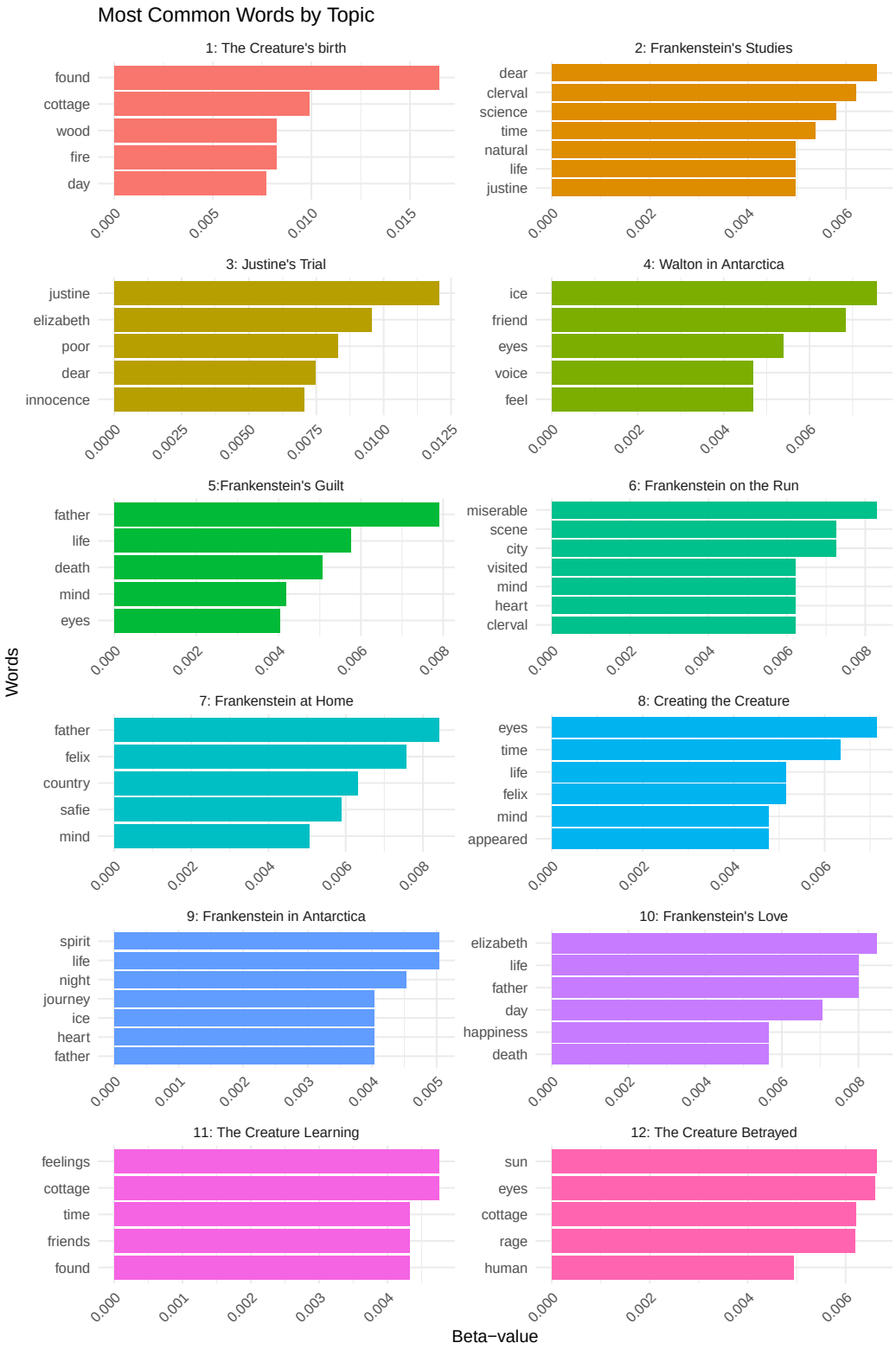

Figure 1: Figure of most common words in each topic

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.

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. 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.

```
# 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")

# Create visual
gamma_with_narrator |>
  mutate(topic = factor(topic, levels = 1:12, labels = topic_names)) |>
  ggplot(aes(x = document, y = gamma, fill = factor(topic))) +
  geom_bar(stat ="identity") +
  facet_wrap(~ narrator, scales = "free", ncol = 1) +
```

```
  theme_minimal() +
  labs(x = "Section", y = "Topic Proportion", fill = "Topic",
       title = "Gamma Topic Distribution Across Frankenstein") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

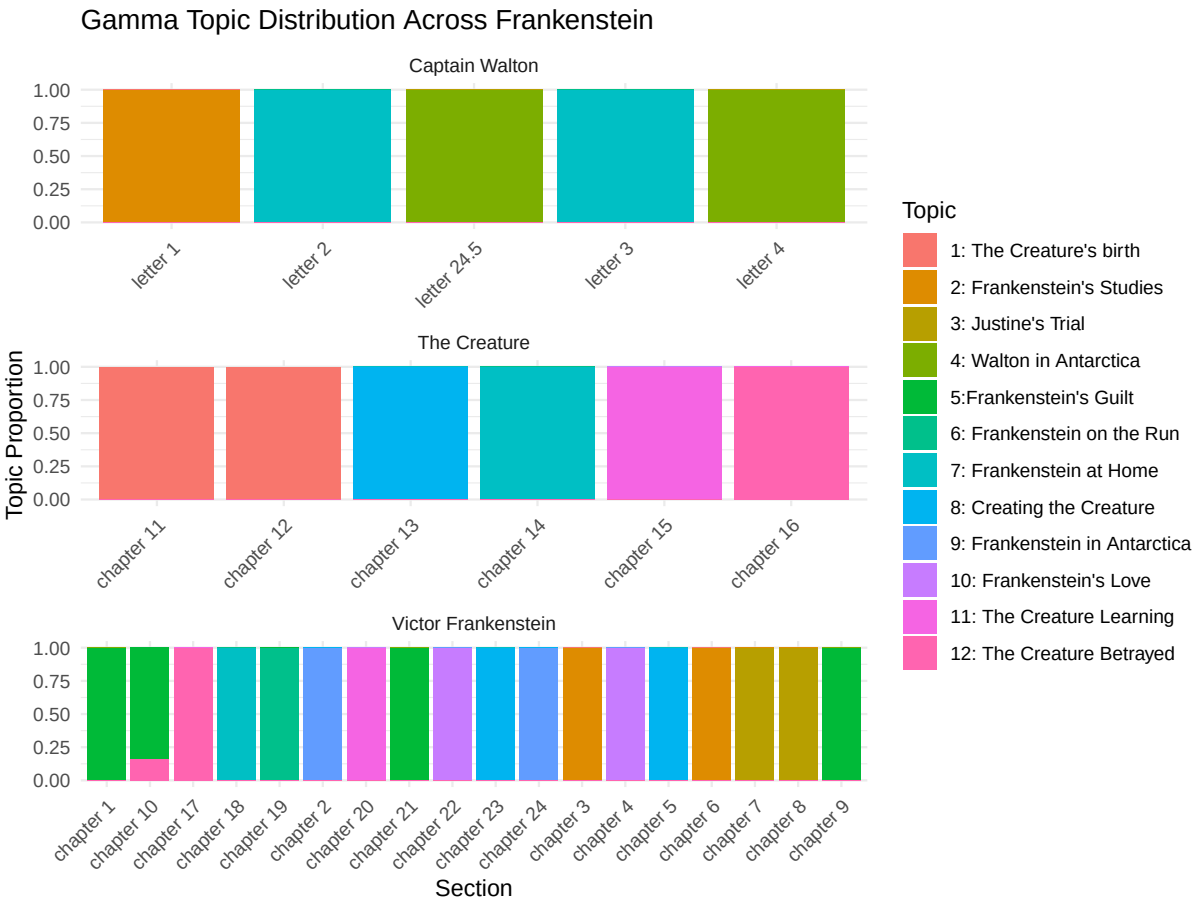

Figure 2: Per-section-per-topic probabilities

Overall, LDA topic modelling was able to detect some of the framing device that was used in Frankenstein. Obviously, the model is not perfect and there were many of the topics that had words from all of the book, but with potentially more fine-tuning of parameters the model could perform better.