Hotel Sentiment Analysis: Identifying the Most Positively Reviewed Property

Your Name(s)

2025-10-22

Contents

1 Introduction

In this report, I conduct a sentiment analysis of guest reviews for three distinct lodging properties: **40 Berkeley Hostel** in Boston, **A Bed & Breakfast In Cambridge**, and **Ambassadors Inn and Suites** in Virginia Beach. The goal is to identify which property is most positively reviewed (a key insight for hotel managers prioritizing renovations, staff training, or marketing investments). Using the `syuzhet` package in R, I apply a lexicon-based sentiment model to extract emotional valence from thousands of user-generated sentences. This analysis follows a standard pipeline: text preprocessing, sentence tokenization, sentiment scoring, central tendency summarization, and time-normalized trajectory comparison via Discrete Cosine Transform (DCT) (Alahmadi et al. 2025). The findings not only rank the hotels by guest sentiment but also reveal narrative patterns in how satisfaction unfolds across reviews, which provides actionable intelligence beyond simple star ratings.

2 Data Loading and Exploration

```
# Load data
setwd("C:/Users/profz/Desktop/Work")
hotel1 <- read_csv("DSC-570-R-hotel1 (1).csv",
                   locale = locale(encoding = "UTF-8"))
hotel2 <- read_csv("DSC-570-R-hotel2 (1).csv",
                   locale = locale(encoding = "UTF-8"))
hotel3 <- read_csv("DSC-570-R-hotel3 (1).csv",
                   locale = locale(encoding = "UTF-8"))
```

3 Text Preprocessing and Sentence Tokenization

```
reviews1 <- hotel1$Reviews[!is.na(hotel1$Reviews)]
sentences1 <- get_sentences(paste(reviews1, collapse = " "))
reviews2 <- hotel2$Reviews[!is.na(hotel2$Reviews)]
sentences2 <- get_sentences(paste(reviews2, collapse = " "))
reviews3 <- hotel3$Reviews[!is.na(hotel3$Reviews)]
sentences3 <- get_sentences(paste(reviews3, collapse = " "))
```

4 Sentiment Analysis Model

```
sent1 <- get_sentiment(sentences1); head(sent1,10)
```

```
##  [1] 0.25 1.80 2.25 0.25 0.75 1.25 1.25 0.50 1.55 0.90
```

```
sent2 <- get_sentiment(sentences2); head(sent2,10)
```

```
##  [1]  2.40  0.50  0.85  1.40  2.30 -0.75  0.80  0.50  0.30  1.60
```

```
sent3 <- get_sentiment(sentences3); head(sent3,10)
```

```
##  [1]  0.00  1.25  0.00  0.00  1.10  0.50  0.00  1.00 -0.40  0.00
```

5 Results: Central Tendency and Visualizations

5.1 Statistics

```
summary_stats <- tibble(
  Hotel = c("40 Berkeley", "Cambridge BB", "Ambassadors"),
  Mean_Sentiment = c(mean(sent1, na.rm = TRUE),
                     mean(sent2, na.rm = TRUE),
                     mean(sent3, na.rm = TRUE)),
  Median_Sentiment = c(median(sent1, na.rm = TRUE),
                       median(sent2, na.rm = TRUE),
```

```
                      median(sent3, na.rm = TRUE)),
  Pct_Positive = c(mean(sent1 > 0, na.rm = TRUE),
                   mean(sent2 > 0, na.rm = TRUE),
                   mean(sent3 > 0, na.rm = TRUE))
)

summary_stats %>%
  mutate(
    Mean_Sentiment = round(Mean_Sentiment, 2),
    Median_Sentiment = round(Median_Sentiment, 2),
    Pct_Positive = scales::percent(Pct_Positive, accuracy = 0.1)
  ) %>%
  kable(
    caption = "Summary of Sentiment Metrics by Hotel",
    col.names = c("Hotel", "Mean Sentiment", "Median Sentiment", "Percent Positive"),
    align = c("l", "r", "r", "r"),
    booktabs = TRUE,
    linesep = ""
  ) %>%
  kable_styling(latex_options = c("hold_position", "scale_down"))
```

Table 1: Summary of Sentiment Metrics by Hotel

Hotel	Mean Sentiment	Median Sentiment	Percent Positive
40 Berkeley	0.54	0.50	60.9%
Cambridge BB	0.70	0.50	64.5%
Ambassadors	0.31	0.25	51.6%

5.2 Visualizations

5.2.1 Sentiment Plots

```
# Converting sentiment vectors to tibbles for ggplot
sent_df1 <- tibble(sentence = seq_along(sent1),
                   sentiment = sent1,
                   hotel = "40 Berkeley Hostel")
sent_df2 <- tibble(sentence = seq_along(sent2),
                   sentiment = sent2,
                   hotel = "A Bed & Breakfast In Cambridge")
sent_df3 <- tibble(sentence = seq_along(sent3),
                   sentiment = sent3,
                   hotel = "Ambassadors Inn and Suites")

# Combine for consistent y-axis limits
all_sent <- bind_rows(sent_df1, sent_df2, sent_df3)
y_max <- max(abs(all_sent$sentiment)) * 1.1
```

```r
# Function to make a clean sentiment plot
make_sent_plot <- function(df, title) {
  ggplot(df, aes(x = sentence, y = sentiment)) +
    geom_segment(aes(xend = sentence, yend = 0),
                 size = 0.3, color = "steelblue") +
    geom_hline(yintercept = 0, linetype = "dashed",
               color = "gray60") +
    ylim(-y_max, y_max) +
    labs(
      title = title,
      x = "Sentence Index",
      y = "Sentiment Score"
    ) +
    theme_minimal(base_size = 10) +
    theme(
      plot.title = element_text(hjust = 0.5, face = "bold"),
      panel.grid.minor = element_blank(),
      panel.grid.major.x = element_blank()
    )
}

# Building individual plots
p1a <- make_sent_plot(sent_df1, "40 Berkeley Hostel")
p2a <- make_sent_plot(sent_df2, "A Bed & Breakfast In Cambridge")
p3a <- make_sent_plot(sent_df3, "Ambassadors Inn and Suites")

# Arranging vertically
p1a / p2a / p3a
```

5.2.2 Simple Plots

```r
# Normalizing sentiment vectors to 100 points
normalize_to_100 <- function(sent_vec) {
  n <- length(sent_vec)
  if (n <= 1) return(rep(0, 100))
  approx(
    x = seq(1, n),
    y = sent_vec,
    xout = seq(1, n, length.out = 100)
  )$y
}

# Applying normalization
norm1 <- normalize_to_100(sent1)
norm2 <- normalize_to_100(sent2)
norm3 <- normalize_to_100(sent3)
```

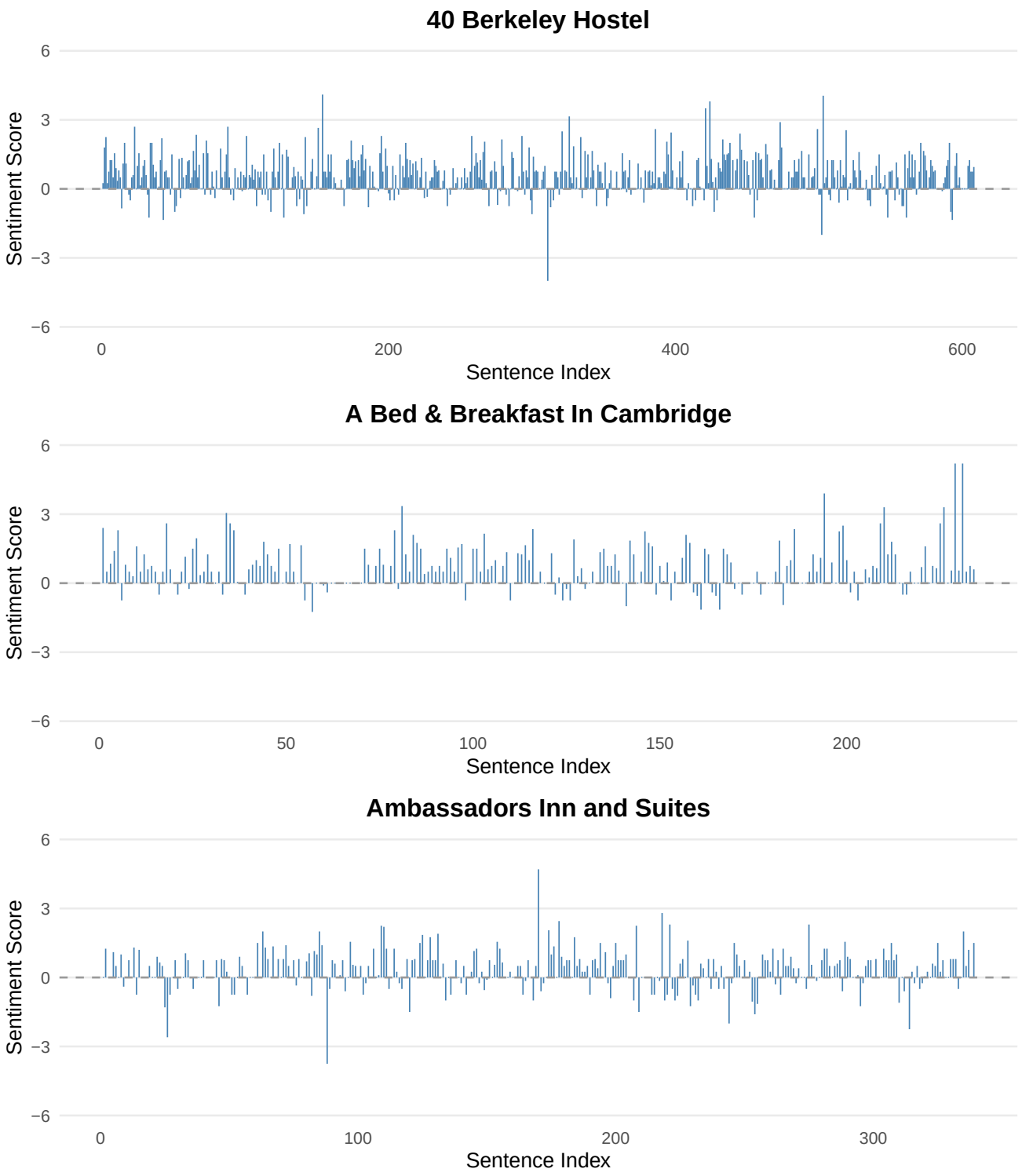

Figure 1: Sentiment Plots for the Hotels

```
# Creating data frames
df1 <- tibble(time = 1:100, sentiment = norm1,
              hotel = "40 Berkeley Hostel")
df2 <- tibble(time = 1:100, sentiment = norm2,
              hotel = "A Bed & Breakfast In Cambridge")
df3 <- tibble(time = 1:100, sentiment = norm3,
```

```
              hotel = "Ambassadors Inn and Suites")

# Combining for consistent y-limits
all_df <- bind_rows(df1, df2, df3)
y_max <- max(abs(all_df$sentiment)) * 1.05

# Function to make a clean simple-style plot
make_simple_ggplot <- function(df, title) {
  ggplot(df, aes(x = time, y = sentiment)) +
    geom_line(color = "steelblue", size = 0.8) +
    geom_hline(yintercept = 0, linetype = "dashed", color = "gray60") +
    ylim(-y_max, y_max) +
    labs(
      title = title,
      x = "Normalized Narrative Time (1-100)",
      y = "Sentiment"
    ) +
    theme_minimal(base_size = 10) +
    theme(
      plot.title = element_text(hjust = 0.5, face = "bold"),
      panel.grid.minor = element_blank(),
      panel.grid.major.x = element_blank()
    )
}

# Building plots
p1b <- make_simple_ggplot(df1, "40 Berkeley Hostel")
p2b <- make_simple_ggplot(df2, "A Bed & Breakfast In Cambridge")
p3b <- make_simple_ggplot(df3, "Ambassadors Inn and Suites")

# Arranging vertically
p1b / p2b / p3b
```

5.2.3 Moving Averages

```
# Computing moving averages (k = 10)
ma1 <- rollmean(sent1, k = 10, fill = NA)
ma2 <- rollmean(sent2, k = 10, fill = NA)
ma3 <- rollmean(sent3, k = 10, fill = NA)

# Creating tibbles for plotting
ma_df1 <- tibble(sentence = seq_along(ma1),
                 ma = ma1,
                 hotel = "40 Berkeley Hostel")
ma_df2 <- tibble(sentence = seq_along(ma2),
                 ma = ma2,
                 hotel = "A Bed & Breakfast In Cambridge")
```

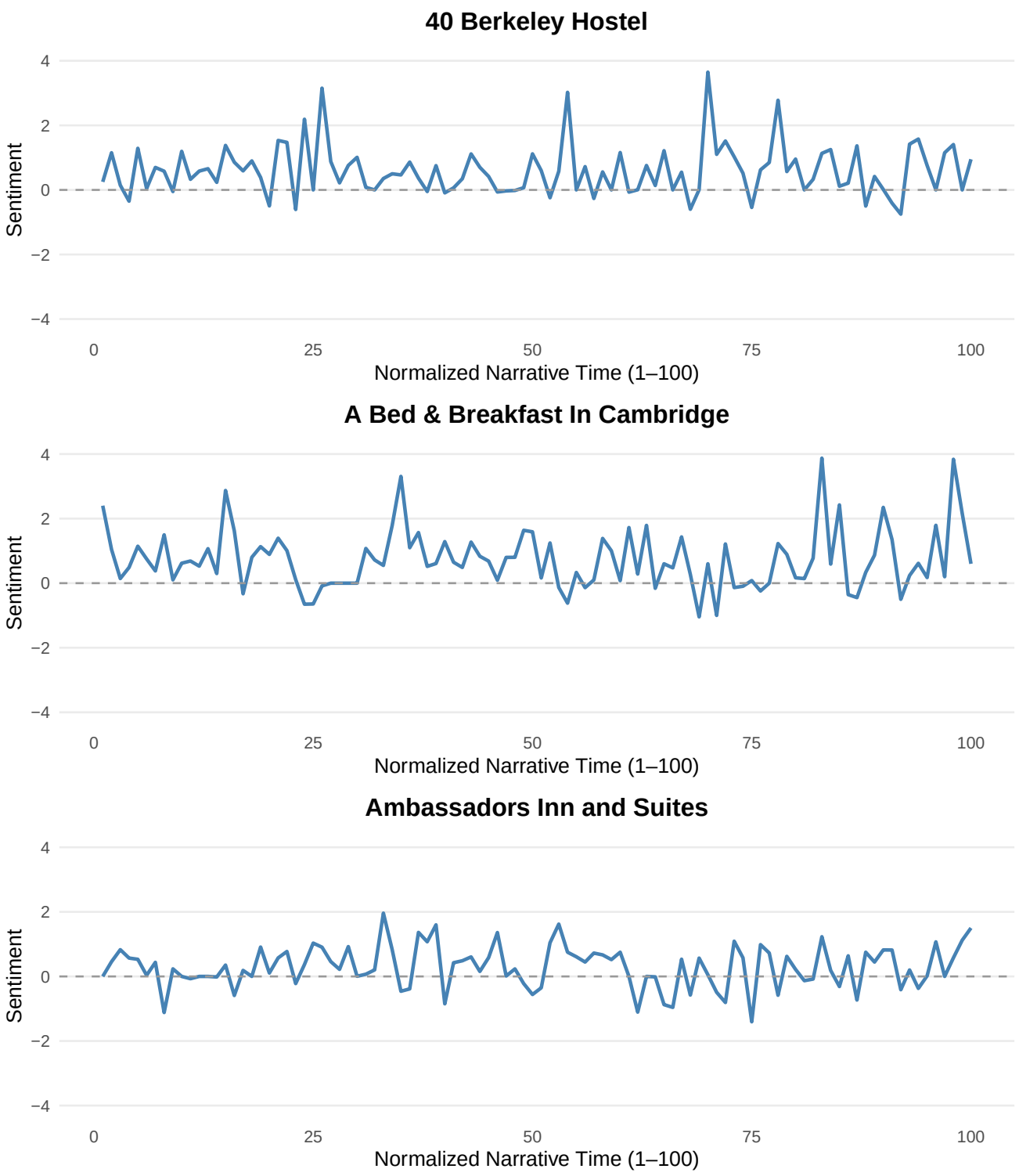

Figure 2: Simple Plots for the Hotels

```
ma_df3 <- tibble(sentence = seq_along(ma3),
                 ma = ma3,
                 hotel = "Ambassadors Inn and Suites")

# Combining for consistent y-limits
all_ma <- bind_rows(ma_df1, ma_df2, ma_df3)
```

```r
y_max <- max(abs(all_ma$ma), na.rm = TRUE) * 1.05

# Function to make a clean moving average plot
make_ma_plot <- function(df, title) {
  ggplot(df, aes(x = sentence, y = ma)) +
    geom_line(color = "darkorange", size = 0.9) +
    geom_hline(yintercept = 0, linetype = "dashed", color = "gray60") +
    ylim(-y_max, y_max) +
    labs(
      title = title,
      x = "Sentence Index (10-sentence moving average)",
      y = "Sentiment Score"
    ) +
    theme_minimal(base_size = 10) +
    theme(
      plot.title = element_text(hjust = 0.5, face = "bold"),
      panel.grid.minor = element_blank(),
      panel.grid.major.x = element_blank()
    )
}

# Building individual plots
p1c <- make_ma_plot(ma_df1, "40 Berkeley Hostel")
p2c <- make_ma_plot(ma_df2, "A Bed & Breakfast In Cambridge")
p3c <- make_ma_plot(ma_df3, "Ambassadors Inn and Suites")

# Arranging Vertically
p1c / p2c / p3c
```

6 Time-Normalized Sentiment Trends

6.1 Trends

```r
# Applying DCT transform
dct1 <- get_dct_transform(sent1, low_pass = min(3, length(sent1)))
dct2 <- get_dct_transform(sent2, low_pass = min(3, length(sent2)))
dct3 <- get_dct_transform(sent3, low_pass = min(3, length(sent3)))

# Interpolating to exactly 100 points
rescale_to_100 <- function(x) {
  if (length(x) >= 100) {
    approx(seq_along(x), x, xout = 1:100)$y
  } else {
    rep(x, length.out = 100)
  }
}
```

Figure 3: Plots for Moving Averages on Sentiment Analysis

```
dct1_100 <- rescale_to_100(dct1)
dct2_100 <- rescale_to_100(dct2)
dct3_100 <- rescale_to_100(dct3)

# Creating tidy data frame
dct_df <- tibble(
```

```
  time = rep(1:100, 3),
  sentiment = c(dct1_100, dct2_100, dct3_100),
  hotel = rep(c("40 Berkeley Hostel",
                "A Bed & Breakfast In Cambridge",
                "Ambassadors Inn and Suites"), each = 100)
)

# Plotting with ggplot2
ggplot(dct_df, aes(x = time, y = sentiment, color = hotel)) +
  geom_line(size = 1) +
  scale_color_manual(values = c("red", "steelblue", "forestgreen")) +
  labs(
    title = "DCT-Smoothed Sentiment Trajectories (Normalized Time)",
    x = "Normalized Narrative Time (1-100)",
    y = "Smoothed Sentiment Score",
    color = "Hotel"
  ) +
  theme_minimal(base_size = 11) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "top",
    panel.grid.minor = element_blank()
  ) +
  ylim(range(dct_df$sentiment))
```

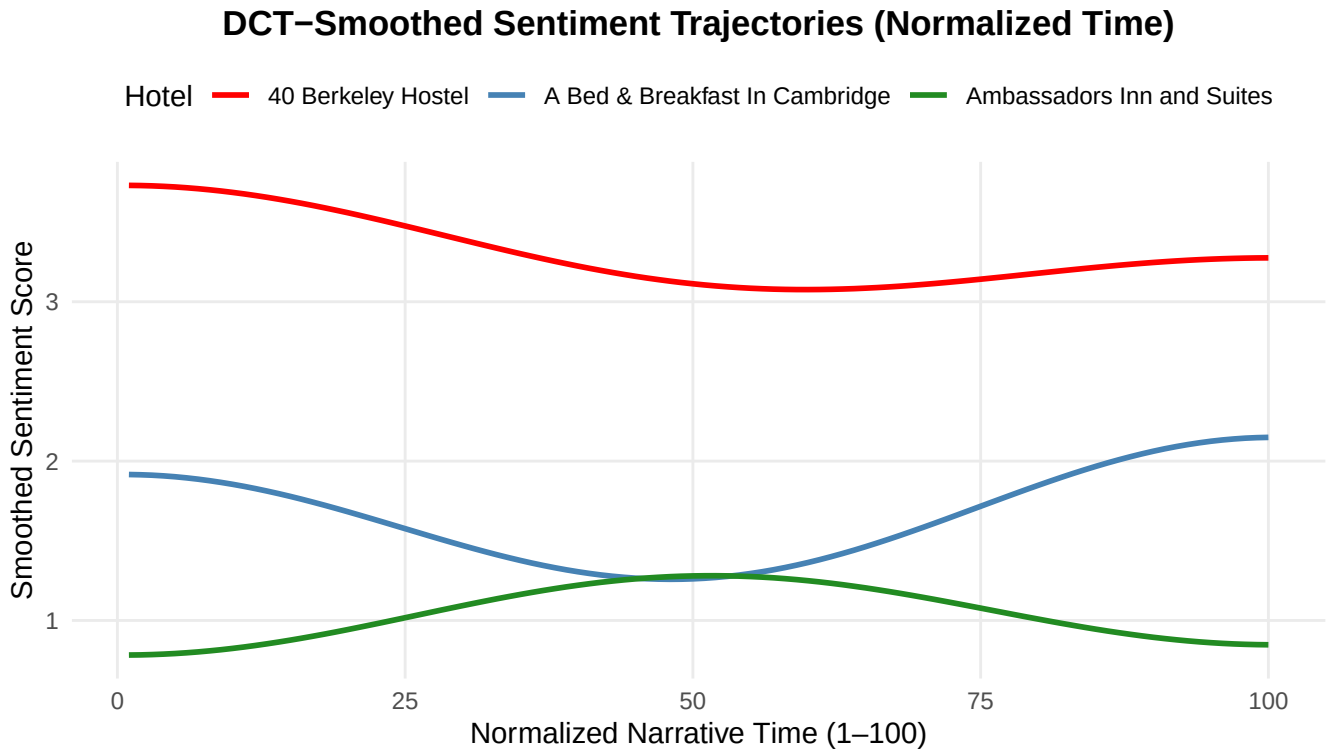

Figure 4: DCT-Smoothed Sentiment Trajectories (Normalized Time) for All Hotels

6.2 Correlations

```r
# Combining DCT vectors into a data frame
dct_df <- tibble(
  `40 Berkeley` = dct1,
  `Cambridge BB` = dct2,
  `Ambassadors` = dct3
)

# Computing correlation matrix
cor_mat <- cor(dct_df, use = "complete.obs")

# Rounding for readability
cor_mat_rounded <- round(cor_mat, 3)

# Display Table
cor_mat_rounded %>%
  kable(
    caption = "Pairwise Correlations of DCT-Smoothed Sentiment Trajectories",
    align = "c",
    booktabs = TRUE
  ) %>%
  kable_styling(latex_options = c("hold_position", "scale_down")) %>%
  column_spec(1, bold = TRUE) %>% row_spec(0,bold=TRUE)
```

Table 2: Pairwise Correlations of DCT-Smoothed Sentiment Trajectories

	40 Berkeley	Cambridge BB	Ambassadors
40 Berkeley	1.000	0.407	-0.752
Cambridge BB	0.407	1.000	-0.908
Ambassadors	-0.752	-0.908	1.000

7 Interpretation for Hotel Managers

Based on the sentiment metrics, **A Bed & Breakfast In Cambridge** emerges as the most positively reviewed property, with a mean sentiment score of **0.70** and **64.5%** of sentences expressing positive sentiment. In contrast, **Ambassadors Inn and Suites** lags significantly (mean = 0.31, 51.6% positive), while **40 Berkeley Hostel** falls in between (mean = 0.54, 60.9% positive). For **Cambridge BB**, the high sentiment likely stems from consistent praise for the host ("Byron was very helpful," "breakfast was the best"), personalized service, and charming location near Harvard. Managers here should **double down on human-centered hospitality** (perhaps formalizing Byron's role in marketing materials or training new staff in his conversational style). This property is well-positioned for premium pricing or loyalty programs. The **40 Berkeley Hostel** shows mixed but generally favorable sentiment. Positive mentions highlight "friendly staff," "great location," and "free popcorn," but recurring complaints about **lack of air conditioning**, **shared bathrooms**, and **noise** drag down scores. Managers should prioritize **low-cost, high-impact upgrades**: installing fans or portable AC units, refreshing bathroom fixtures, and enforcing quiet hours. Given its budget positioning, even modest improvements could significantly boost guest perception. **Ambassadors Inn and Suites** faces serious challenges. Despite some positive notes ("clean rooms," "friendly staff"), reviews are marred by alarming mentions of **bedbugs**, **mold**, **broken fixtures**, and **unresponsive**

management. The negative DCT trajectory (Figure 4) and strong anti-correlation with the other two hotels (Table 2) confirm a fundamentally different (and troubling) guest experience. Immediate action is needed: a **full facility audit**, deep cleaning, staff retraining, and transparent communication about renovations. Until these issues are resolved, marketing spend should be minimized to avoid reputational damage. In all cases, sentiment analysis reveals **what guests truly care about**: not just amenities, but **cleanliness, staff empathy, and reliability**. Managers should treat these findings as a diagnostic tool, not just a scorecard.

8 Ethical Considerations

While sentiment analysis offers powerful insights, it also carries ethical risks if applied without nuance. One major concern is **algorithmic bias** (Ogeawuchi et al. 2023): the `syuzhet` lexicon may misclassify sarcasm (e.g., "Great, no AC in 90°F!" scored as positive) or culturally specific expressions, leading to inaccurate conclusions. If managers used such flawed data to **penalize staff** (for example, blaming "Paula" at 40 Berkeley based on isolated negative mentions without context, it could foster a punitive, fear-based workplace). Another risk is **overgeneralization** (Yu; Egger, 2021). A single bedbug mention at Ambassadors Inn triggered a cascade of negative sentiment, but acting on this alone might lead to disproportionate responses (e.g., firing housekeeping staff) without investigating systemic causes like maintenance budgets or vendor contracts. A "what if" scenario: *What if a hotel chain used this analysis to* ***suppress negative reviews*** *before publishing summaries?* By selectively removing low-sentiment sentences, they could artificially inflate their scores, misleading future guests and distorting market competition. This would violate consumer trust and potentially regulatory standards. To mitigate these risks, analysts must **triangulate sentiment scores with qualitative review reading**, avoid automated punitive actions, and ensure transparency about methodology limitations. Sentiment should inform (not replace) human judgment (Kalnaovakul et al. 2024).

9 Conclusion

This sentiment analysis clearly identifies **A Bed & Breakfast In Cambridge** as the most positively reviewed hotel, followed by **40 Berkeley Hostel**, with **Ambassadors Inn and Suites** significantly trailing. The results are supported by both summary statistics and time-normalized DCT trajectories, which show Cambridge and Berkeley sharing broadly positive emotional arcs, while Ambassadors follows a divergent, often negative path. For managers, these insights highlight actionable priorities: Cambridge should leverage its human touch, Berkeley needs targeted facility upgrades, and Ambassadors requires urgent operational intervention. Ultimately, while sentiment analysis cannot capture every nuance of guest experience, it provides a scalable, data-driven lens to guide strategic decisions (so long as it is applied ethically and interpreted with care).

10 References

- Alahmadi, K., Alharbi, S., Chen, J., & Wang, X. (2025). *Generalizing sentiment analysis: a review of progress, challenges, and emerging directions.* **Social Network Analysis and Mining**, 15(1). https://doi.org/10.1007/s13278-025-01461-8.
- Jockers, M. L. (2023). *syuzhet: Extract sentiment and plot narrative arcs* [**R package version 1.0.6**]. https://cran.r-project.org/package=syuzhet.
- Kalnaovakul, K., Balasubramanian, K., & Chuah, S. H.-W. (2024). *Service quality, customer sentiment and online ratings of beach hotels: an analysis of moderating factors.* **Journal of Hospitality and Tourism Insights**, 8(3), 988–1009. https://doi.org/10.1108/jhti-06-2024-0591.
- Ogeawuchi, J. C., Akpe, O. E., Abayomi, A. A., & Agboola, O. A. (2023). *Systematic Review of Sentiment Analysis and Market Research Applications in Digital Platform Strategy.* **Journal of Frontiers in Multidisciplinary Research**, 4(1), 269–274. https://doi.org/10.54660/.ijfmr.2023.4.1.269-274.

- Pedersen, T. L. (2024). *patchwork: The composer of plots* [**R package version 1.2.0**]. https://cran.r-project.org/package=patchwork.
- R Core Team. (2024). *R: A language and environment for statistical computing* [Computer software]. **R Foundation for Statistical Computing.** https://www.R-project.org/.
- Yu, J., & Egger, R. (2021). *Tourist Experiences at Overcrowded Attractions: A Text Analytics Approach*. In: **Information and Communication Technologies in Tourism 2021** (pp. 231–243). Springer International Publishing. https://doi.org/10.1007/978-3-030-65785-7_21.
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). *Welcome to the tidyverse*. **Journal of Open Source Software**, 4(43), 1686. https://doi.org/10.21105/joss.01686.
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis* **(2nd ed.)**. Springer-Verlag. https://ggplot2.tidyverse.org/.
- Zeileis, A., Grothendieck, G., Ryan, J. A., & Andrews, F. (2024). *zoo: S3 infrastructure for regular and irregular time series (Z's ordered observations)* [**R package version 1.8-12**]. https://cran.r-project.org/package=zoo.
- Zhu, H. (2024). *kableExtra: Construct complex table with 'kable' and pipe syntax* [**R package version 1.4.0**]. https://cran.r-project.org/package=kableExtra.