Unsupervised Sentiment Analysis of Hotel Reviews: A Clustering Approach with TF-IDF and PCA

1st Md Tasmim Al Tahsin
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
22-46299-1@student.aiub.edu

2nd Abrar Shakil Oishik
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
22-46257-1@student.aiub.edu

3rd Mahir Chowdhury
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
22-46162-1@student.aiub.edu

4th Aishee Debnath
Department of Computer Science
American International University-Bangladesh
City, Country
22-46416-1@student.aiub.edu

5th Abdus Salam
Department of Computer Science
American International University-Bangladesh
Dhaka, Bangladesh
abdus.salam@aiub.edu

***Abstract*—Understanding customer opinions in hotel reviews is important for improving services in the hospitality industry. This study examines three widely used clustering algorithms K-means, DBSCAN, and hierarchical clustering to identify and organize sentiment from online hotel reviews. These methods are used to group reviews in a clear and meaningful way. This organizes the reviews into sentiment categories that show what customers liked, disliked, or felt strongly about, making it easier to understand their overall experience. The study is tested on real hotel review datasets, using text processing methods like Doc2vec and TF-IDF to convert the reviews into numerical form. The experiments show that K-means++, this is a variant of K-means an improved version of K-means, creates more reliable and accurate clusters. Hierarchical clustering effectively represents sentiment differences among hotel categories, while DBSCAN can endure dense sentiment cluster identification in noisy data. Comparative analysis addresses the advantages and disadvantages of each algorithm regarding their ability to manage extensive unstructured review data. Findings obtained by separately clustering positive and negative reviews result in practical knowledge for hotel management focused on enhancing services tailored to customer preferences. This study enhances sentiment analysis techniques by integrating modern clustering approaches, thus advancing automated customer feedback ratings within the hospitality sector. The study shows that selecting algorithms thoughtfully based on dataset features can significantly enhance the quality of sentiment analysis outcomes and their practical use in the real world.**

***Index Terms*—NLP, sentiment analysis, Tf-IDF, clustering, K-means**

I. Introduction

Sentiment analysis of hotel reviews has become a most crucial way of comprehending service quality and customer experience in the hospitality industry. With review websites increasing exponentially, it is both an opportunity and a challenge to derive useful insights from enormous amounts of unstructured text data. K-means, DBSCAN, and hierarchical clustering are the clustering algorithms that have been effective unsupervised learning techniques in organizing and processing large-scale sentiment data by classifying reviews with similar text features. These methods not only support the identification of prevailing customer sentiments but also review segmentation to determine distinct opinion patterns for different hotel attributes. Recent studies have shown the increased ability of state-of-the-art clustering algorithms for sentiment analysis purposes. For example, Wang et al. 2024) provided the superiority of the K means++ algorithm with Doc2vec and TF IDF vectorization in efficiently clustering and analyzing hotel reviews on a Chinese online marketplace, with interesting insights into positive and negative consumer comments [1]. Similarly, Uysal et al. 2025) applied hierarchical clustering and topic modeling techniques to examine the influence of hotel star ratings on customer sentiment in Astana, Kazakhstan, targeting varied sentiment distributions across different hotel categories [2]. Such research illustrates the benefits of clustering techniques in obtaining practical knowledge from sentiment-rich hotel review data with significant implications toward maximizing customer satisfaction and service plans. This paper aims to explore and contrast applications of the use of K-means, DBSCAN, and hierarchical clustering algorithms in hotel review sentiment analysis based on the most recent advancements and applications documented in the literature.

Through systematic review and application of such clustering algorithms, this study attempts to contribute to more effective sentiment-based customer insight extraction techniques in the hospitality sector.

II. Literature Review

Recent studies on hotel review sentiment analysis are increasingly using clustering methods to understand customer experiences. These algorithms help group online reviews into meaningful categories, making it easier to see what users think and feel about their stay. One implemented study (Wang et al., 2024) used unsupervised clustering algorithms—traditional K-means, Canopy K-means, and the improved K-means++—with Doc2vec text vectorization and TF-IDF weighting to classify hotel reviews from a major Chinese platform [1]. The study identified that K-means++ provided superior clustering cohesion and better data segmentation, enabling extraction of both broad and detailed feature sets from positive and negative reviews. The analysis found that guests valued good service and a comfortable hotel environment the most, while problems with sleep quality and indoor amenities were the main reasons for negative reviews. This gives hotels a clear idea of which areas need improvement. In a 2025 study, Uysal and colleagues examined how hotel star ratings affect guest opinions in Kazakhstan's capital, where star ratings are not standardized. They looked at nearly 6,000 TripAdvisor reviews and used methods like hierarchical clustering, correspondence analysis, and topic modeling to understand patterns in guest feedback [2]. The study found that 2- and 3-star hotels had similar guest opinions, while 4- and 5-star hotels had more complicated and connected issues.. Even 5-star hotels received neutral and negative reviews, showing that top-rated hotels do not always fully meet guest expectations.. Using VADER to measure sentiment and exploring topics in detail helped provide a clear picture of how different hotel features relate to customer feelings based on star ratings. These implemented studies show that combining unsupervised clustering methods with sentiment analysis and sophisticated text representation models enhances detailed extraction and differentiation of customer sentiment in hotel review data. K-means++ improves clustering reliability in large review datasets, while hierarchical clustering aids in revealing sentiment disparities across hotel categories. Both methods contribute valuable analyses for service improvement and customer satisfaction optimization in the hospitality industry. If needed, more recent studies using DBSCAN to cluster hotel reviews based on sentiment can be explored to add to these findings. Looking at such research can provide additional insights and help strengthen the comparisons between different clustering methods, giving a clearer picture of how sentiment analysis can be applied to hotel reviews..

TABLE I
Comparison of Clustering and Sentiment Analysis Methods

Aspect	**Wang et al. (2024) [1]**	**Uysal et al. (2025) [2]**
Dataset	4000 Chinese hotel reviews (Meituan platform)	5894 TripAdvisor hotel reviews (Astana, Kazakhstan)
Clustering Methods	K-means, Canopy K-means, K-means++	Hierarchical clustering and correspondence analysis
Text Representation	Doc2vec, TF-IDF weighting	Non-negative Matrix Factorization (NMF) for topic modeling
Sentiment Analysis Approach	Separate clustering of positive and negative reviews	VADER lexicon-based sentiment scoring
Key Findings	K-means++ shows superior clustering; environmental protection topics highlighted	Sentiment variation linked to hotel star ratings

III. Methodology

The methodology of this study consisted of a structured workflow to perform sentiment analysis and clustering on hotel booking reviews using R software. The approach included the following steps:

A. Data Preparation

The initial dataset comprised review text, titles, ratings, and hotel names. Records with null or empty fields were removed to ensure data quality. From the cleaned dataset, a representative sample of 2,000 reviews was randomly selected for analysis.

B. Text Cleaning

The review texts underwent comprehensive preprocessing which involved converting all characters to lowercase, removing numerical digits and punctuation marks, and stripping excess whitespace. Special attention was given to handling contractions, emojis, and emoticons to normalize textual expressions. The cleaned texts were then compiled into a corpus for further processing.

C. Tokenization and Normalization

The clean corpus was tokenized by breaking the text into individual words. To unify word forms, both stemming and lemmatization techniques were applied, reducing words to their root or base forms.

D. TF-IDF Matrix and Dimensionality Reduction

A Term Frequency-Inverse Document Frequency (TF-IDF) matrix was constructed from the lemmatized corpus to quantify the importance of words across reviews. Sparse terms were pruned to decrease noise, and the matrix was standardized through scaling. Principal Component Analysis (PCA) was employed to reduce the dimensionality of the TF-IDF features, facilitating clustering and visualization by focusing on the most informative components.

E. Sentiment Analysis

Sentiment scores were computed using the `syuzhet` package. Each review received a sentiment score, based on which it was categorized into positive, neutral, or negative sentiment classes.

F. Clustering

Three clustering algorithms were applied to the reduced feature set:

- K-means clustering with the number of clusters $k = 3$.
- Hierarchical clustering using Ward's linkage method to form clusters.
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) applied on the first five PCA components, with the epsilon parameter estimated based on data distribution.

G. Visualization and Evaluation

Cluster formations were visualized in a two-dimensional PCA space to interpret spatial separation. Within each cluster, sentiment distributions were analyzed to understand sentiment grouping. Cluster quality and validity were assessed using silhouette scores. Additional exploratory visualizations such as histograms, density plots, word clouds, and a correlation heatmap were generated to reveal data patterns and relationships comprehensively.

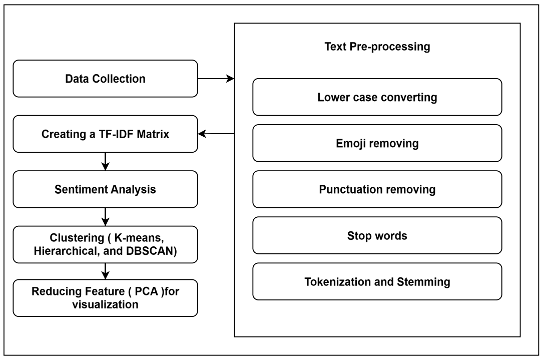

Fig. 1. Research methodology for sentiment analysis and clustering

IV. Implementation

In this study, we performed sentiment analysis and clustering on hotel booking reviews using R. The workflow involved several key steps:

A. Data Preparation

The dataset containing review text, titles, ratings, and hotel names was first cleaned by removing null and empty entries. A dataset containing of 26675 reviews was analyzed. After cleaning 26385 data was selected for final analysis.

Listing 1. Data cleaning in R

```
data <- read.csv("E:\\Data_Science\\Dataset\\booking
    _reviews.csv", stringsAsFactors = FALSE)
data[data == "NULL"] <- NA
data[data == ""] <- NA
clean_data <- data[!is.na(data$review_text) & !is.na
    (data$review_title) &
              !is.na(data$rating) & !is.na(data$
                  hotel_name), ]
```

B. Text Cleaning

Text preprocessing included converting to lowercase, removing numbers and punctuation, stripping whitespace, and handling contractions, emojis, and emoticons. The processed text was stored in a clean corpus for further analysis.

Listing 2. Text cleaning in R

```
corpus <- VCorpus(VectorSource(reviews_raw))
corpus <- tm_map(corpus, content_transformer(replace
    _emoticon_emoji_contraction))
corpus <- tm_map(corpus, content_transformer(tolower
    ))
corpus <- tm_map(corpus, content_transformer(
    function(x) gsub("[\n]+", "␣", x)))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)

clean_text <- sapply(corpus, function(x) x$content)
```

C. Tokenization and Lemmatization

The cleaned text was tokenized into words, followed by stemming and lemmatization to reduce words to their base forms.

Listing 3. Tokenization and Lemmatization in R

```
tokens_list <- tokenize_words(clean_text, lowercase
    = FALSE)
stemmed_text <- sapply(tokens_list, function(tok_vec
    ){
  if(length(tok_vec)==0) return("")
  paste(wordStem(tok_vec, language = "en"), collapse
      = "␣")
}, USE.NAMES = FALSE)
lemmatized_text <- textstem::lemmatize_strings(clean
    _text)
reviews_df <- data.frame(
  doc_id = seq_along(clean_text),
  raw = reviews_raw,
  cleaned = clean_text,
  stemmed = stemmed_text,
  lemmatized = lemmatized_text,
  stringsAsFactors = FALSE
)
```

D. TF-IDF Matrix and Dimensionality Reduction

A TF-IDF matrix was constructed from the lemmatized corpus. Sparse terms were removed, and the resulting matrix was scaled. Principal Component Analysis (PCA) was then applied to reduce dimensions for clustering and visualization.

Listing 4. TF-IDF Matrix in R

```
corpus_lemma <- VCorpus(VectorSource(reviews_df$
    lemmatized))
dtm <- DocumentTermMatrix(corpus_lemma,
                          control = list(wordLengths = c(2,
                              Inf)))
dtm_tfidf <- weightTfIdf(dtm)
dtm_tfidf_trim <- removeSparseTerms(dtm_tfidf, 0.99)
tfidf_mat <- as.matrix(dtm_tfidf_trim)
tfidf_mat <- tfidf_mat[, apply(tfidf_mat, 2, sd, na.
    rm = TRUE) > 0, drop = FALSE]
tfidf_scaled <- scale(tfidf_mat)
```

E. Sentiment Analysis

Sentiment scores were calculated using the syuzhet package. Each review was labeled as positive, neutral, or negative based on its sentiment score.

Listing 5. Sentiment Analysis in R

```
syuzhet_scores <- get_sentiment(reviews_df$
    lemmatized, method = "syuzhet")

reviews_df$sentiment_score <- syuzhet_scores
reviews_df$sentiment_label <- ifelse(reviews_df$
    sentiment_score > 0, "positive",
                            ifelse(reviews_df$
                                sentiment_score < 0,
                                 "negative", "
                                neutral"))
```

F. Clustering

Three clustering algorithms were applied to the TF-IDF features:

- K-means clustering with $k = 3$.
- Hierarchical clustering using Ward's method.
- DBSCAN applied on the first five PCA components with an estimated epsilon.

G. Visualization and Evaluation

Clusters were visualized in two-dimensional PCA space. Sentiment distribution within clusters was analyzed, and cluster quality was assessed using silhouette scores. Additionally, histograms, density plots, word clouds, and a correlation heatmap were generated to explore data patterns.

V. Results and Analysis

This section presents the outcomes of the sentiment analysis and clustering performed on the hotel booking review dataset. The results are supported by both tabular summaries and visualizations generated during the analysis.

A. Sentiment Distribution

Using the syuzhet package, each review was assigned a sentiment score and categorized into *positive*, *neutral*, or *negative*. The majority of reviews were classified as positive, with a smaller proportion being neutral, and the least proportion negative.

TABLE II
Sentiment Classification Summary

Sentiment	Count	Percentage
Positive	24052	91.17%
Neutral	1083	4.10%
Negative	1250	4.74%

B. Clustering Results

Three clustering algorithms were applied on PCA-reduced features: K-means, Hierarchical, and DBSCAN. Cluster formations were visualized in a two-dimensional PCA space.

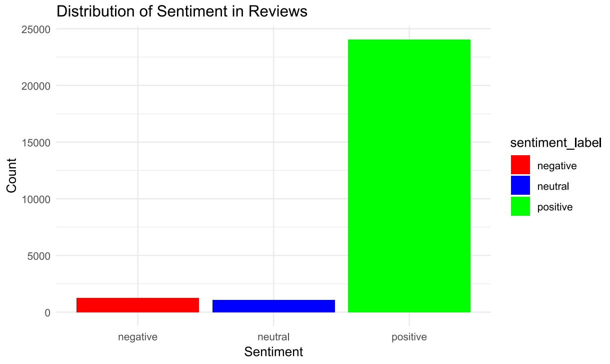

Fig. 2. Distribution of Sentiment in Reviews

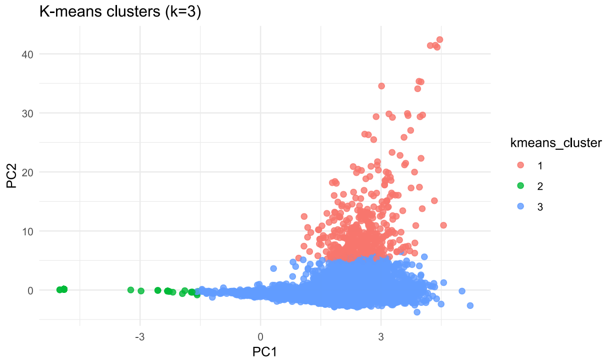

Fig. 3. K-means Clusters

C. Exploratory Visualizations

Additional exploratory visualizations such as histograms, density plots, and sentiment vs. rating comparisons were generated to provide deeper insights into the dataset.

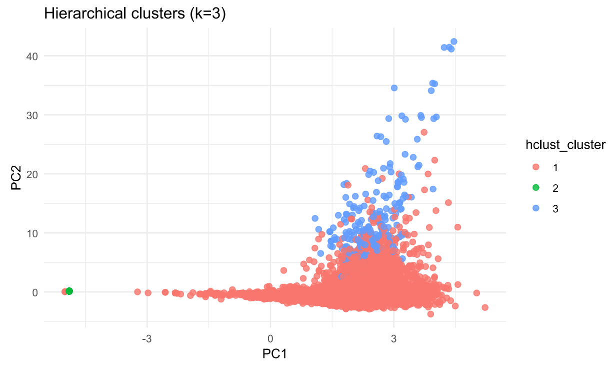

Fig. 4. Hierarchical Clusters

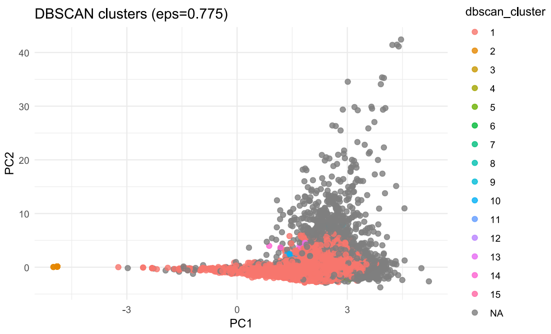

Fig. 5. Review Count by Rating Histogram

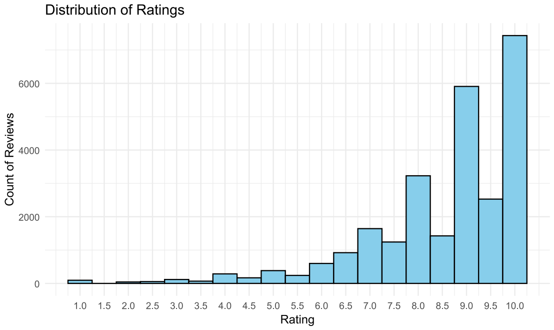

Fig. 6. Review Count by Rating Histogram

VI. Conclusion

This study applied sentiment analysis and clustering algorithms to a large-scale hotel review dataset consisting of 26,385 entries. The results revealed that the majority of customer feedback was positive, with 24,052 reviews (91.17%) expressing satisfaction with hotel services. Neutral sentiments accounted for 1,083 reviews (4.10%), while negative sentiments made up 1,250 reviews (4.74%). These findings highlight a strong inclination toward positive customer experiences, suggesting that hotels generally succeed in delivering quality service. At the same time, the presence of negative and neutral reviews provides valuable opportunities for improvement in specific service areas.

The application of clustering algorithms such as K-means, DBSCAN, and hierarchical clustering proved effective in grouping reviews with similar sentiment features, thereby uncovering distinct patterns of customer opinions. These insights are crucial for hoteliers seeking to enhance service delivery, improve customer satisfaction, and develop more targeted marketing strategies. The analysis also reinforces the growing role of machine learning techniques in the hospitality sector, where unstructured review data can be transformed into actionable knowledge.

Despite these contributions, the study has certain limitations. First, the analysis was restricted to text-based reviews and did not incorporate additional metadata such as reviewer demographics, hotel location, or star ratings, which could provide deeper insights. Second, sentiment classification may not fully capture nuances such as sarcasm, mixed opinions, or cultural variations in language use, potentially affecting the accuracy of the results. Third, the dataset was drawn from a single review source, which may limit the generalizability of the findings across different platforms and geographic contexts.

Future research could address these limitations by integrating multimodal data sources (e.g., ratings, images, or voice reviews), employing more advanced natural language processing techniques such as deep learning models or contextual embeddings, and expanding the analysis to include cross-cultural datasets for broader applicability. Such enhancements would further improve the robustness of sentiment analysis and clustering outcomes, providing even more reliable insights for the hospitality industry.

In conclusion, this study demonstrates the value of combining sentiment analysis with clustering algorithms to extract meaningful insights from hotel reviews. The predominance of positive feedback suggests overall customer satisfaction, while the clustering approach provides actionable information to identify specific areas for improvement. These contributions underline the potential of sentiment-based analytics to guide data-driven decision-making in the hospitality sector.

References

[1] Y. Wang, F. Liu, and G. Li, “Clustering analysis of hotel network reviews based on text mining method,” *Industry Science and Engineering*, vol. 1, pp. 51–59, 04 2024.

[2] A. UYSAL, E. TÜKENMEZ, N. Abdirazakov, M. Basaran, and K. Kantarci, “How hotel stars affecting customers’ sentiment in astana,” *International Journal of Innovative Research and Scientific Studies*, vol. 8, pp. 939–957, 05 2025.

```
install.packages("dplyr")
install.packages("tm")
install.packages("SnowballC")
install.packages("textclean")
install.packages("cluster")
install.packages("factoextra")
install.packages("dbscan")
install.packages("tokenizers")
install.packages(c("hunspell", "textstem"))
install.packages("tidyverse")
install.packages("tidytext")

library(tidyverse)
library(tidytext)
library(tm)
library(SnowballC)
library(textclean)
library(hunspell)
library(textstem)
library(dplyr)
library(cluster)
library(factoextra)
library(dbscan)
library(syuzhet)
library(NLP)
library(tokenizers)
library(RColorBrewer)
library(stringi)
library(reshape2)

data <- read.csv("E:\\Data Science\\Dataset\\booking_reviews.csv", stringsAsFactors = FALSE)
data[data == "NULL"] <- NA
data[data == ""] <- NA
clean_data <- data[!is.na(data$review_text) & !is.na(data$review_title) &
                     !is.na(data$rating) & !is.na(data$hotel_name), ]
set.seed(123)

data_sample <- clean_data
reviews_raw <- as.character(data_sample$review_text)


replace_emoticon_emoji_contraction <- function(x){
  if(requireNamespace("textclean", quietly = TRUE)){
    x <- textclean::replace_contraction(x)
    if("replace_emoticon" %in% ls("package:textclean")){
      x <- textclean::replace_emoticon(x)
    }
    if("replace_emoji" %in% ls("package:textclean")){
      x <- textclean::replace_emoji(x)
    } else {
      x <- stringi::stri_replace_all_regex(x,

"[\U0001F600-\U0001F64F\U0001F300-\U0001F5FF\U0001F680-\U0001F6FF\U00002600-\U0
00027BF]", " ")
    }
  } else {
    contr <- c("won't"="will not","can't"="can not","n't"=" not","'re"=" are","'s"=" is","'d"=" would",
               "'ll"=" will","'ve"=" have","'m"=" am")
    x <- tolower(x)
    for(pat in names(contr)) x <- gsub(pat, contr[pat], x, ignore.case = TRUE, perl = TRUE)
    x <- gsub("(:\\s?\\)|:-\\)|:\\)|:D|=\\))", " smile ", x)
    x <- gsub("(:\\s?\\(|:-\\(|:\\()", " sad ", x)
    x <- stringi::stri_replace_all_regex(x,

"[\U0001F600-\U0001F64F\U0001F300-\U0001F5FF\U0001F680-\U0001F6FF]", " ")
  }
  return(x)
}


corpus <- VCorpus(VectorSource(reviews_raw))
corpus <- tm_map(corpus, content_transformer(replace_emoticon_emoji_contraction))
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, content_transformer(function(x) gsub("[\n]+", " ", x)))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)

clean_text <- sapply(corpus, function(x) x$content)




tokens_list <- tokenize_words(clean_text, lowercase = FALSE)

stemmed_text <- sapply(tokens_list, function(tok_vec){
  if(length(tok_vec)==0) return("")
  paste(wordStem(tok_vec, language = "en"), collapse = " ")
}, USE.NAMES = FALSE)
```

```r
lemmatized_text <- textstem::lemmatize_strings(clean_text)

reviews_df <- data.frame(
  doc_id = seq_along(clean_text),
  raw = reviews_raw,
  cleaned = clean_text,
  stemmed = stemmed_text,
  lemmatized = lemmatized_text,
  stringsAsFactors = FALSE
)


corpus_lemma <- VCorpus(VectorSource(reviews_df$lemmatized))
dtm <- DocumentTermMatrix(corpus_lemma,
                control = list(wordLengths = c(2, Inf)))
dtm_tfidf <- weightTfIdf(dtm)
dtm_tfidf_trim <- removeSparseTerms(dtm_tfidf, 0.99)

tfidf_mat <- as.matrix(dtm_tfidf_trim)
tfidf_mat <- tfidf_mat[, apply(tfidf_mat, 2, sd, na.rm = TRUE) > 0, drop = FALSE]
tfidf_scaled <- scale(tfidf_mat)


pca_res <- prcomp(tfidf_scaled, center = TRUE, scale. = TRUE)
pca_df <- data.frame(doc_id = reviews_df$doc_id,
             PC1 = pca_res$x[,1],
             PC2 = pca_res$x[,2])


syuzhet_scores <- get_sentiment(reviews_df$lemmatized, method = "syuzhet")

reviews_df$sentiment_score <- syuzhet_scores
reviews_df$sentiment_label <- ifelse(reviews_df$sentiment_score > 0, "positive",
                      ifelse(reviews_df$sentiment_score < 0, "negative", "neutral"))

table(reviews_df$sentiment_label)



set.seed(42)
k <- 3
kmeans_res <- kmeans(tfidf_scaled, centers = k, nstart = 25)
pca_df$kmeans_cluster <- factor(kmeans_res$cluster)
```

```
dist_mat <- dist(tfidf_scaled, method = "euclidean")
hclust_res <- hclust(dist_mat, method = "ward.D2")
hclust_clusters <- cutree(hclust_res, k = k)
pca_df$hclust_cluster <- factor(hclust_clusters)


pc_for_db <- pca_res$x[,1:5]
k_nn <- 4
kNNd <- kNNdist(as.matrix(pc_for_db), k = k_nn)
eps_guess <- as.numeric(quantile(kNNd, 0.90))
dbscan_res <- dbscan(as.matrix(pc_for_db), eps = eps_guess, minPts = 5)
pca_df$dbscan_cluster <- factor(ifelse(dbscan_res$cluster == 0, NA, dbscan_res$cluster))



p_kmeans <- ggplot(pca_df, aes(PC1, PC2, color = kmeans_cluster)) +
  geom_point(size=2, alpha=0.8) + ggtitle("K-means clusters (k=3)") + theme_minimal()

p_hclust <- ggplot(pca_df, aes(PC1, PC2, color = hclust_cluster)) +
  geom_point(size=2, alpha=0.8) + ggtitle("Hierarchical clusters (k=3)") + theme_minimal()

p_dbscan <- ggplot(pca_df, aes(PC1, PC2, color = dbscan_cluster)) +
  geom_point(size=2, alpha=0.8, na.rm=TRUE) +
  ggtitle(paste0("DBSCAN clusters (eps=", round(eps_guess,3), ")")) + theme_minimal()

print(p_kmeans)
print(p_hclust)
print(p_dbscan)



combined <- left_join(pca_df, reviews_df %>%
                      select(doc_id, sentiment_label, sentiment_score), by = "doc_id")

cat("\nK-means vs Sentiment:\n")
print(table(combined$kmeans_cluster, combined$sentiment_label))

cat("\nHierarchical vs Sentiment:\n")
print(table(combined$hclust_cluster, combined$sentiment_label))
```

```
cat("\nDBSCAN vs Sentiment:\n")
print(table(combined$dbscan_cluster, combined$sentiment_label, useNA = "ifany"))
```

```
reviews_df$kmeans_cluster <- kmeans_res$cluster
reviews_df$hclust_cluster <- hclust_clusters
reviews_df$dbscan_cluster <- dbscan_res$cluster

head(reviews_df %>%
    select(doc_id, sentiment_label, sentiment_score,
        kmeans_cluster, hclust_cluster, dbscan_cluster, raw), 8)
```

```
sentiment_count <- table(reviews_df$sentiment_label)
sentiment_count


ggplot(reviews_df, aes(x = sentiment_label, fill = sentiment_label)) +
  geom_bar() +
  labs(title = "Distribution of Sentiment in Reviews", x = "Sentiment", y = "Count") +
  scale_fill_manual(values = c("positive"="green", "neutral"="blue", "negative"="red")) +
  theme_minimal()


reviews_df$review_length <- sapply(strsplit(reviews_df$raw, "\\s+"), length)

ggplot(reviews_df, aes(x = review_length)) +
  geom_histogram(binwidth = 10, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Review Length", x = "Number of Words", y = "Count") +
  theme_minimal()
```

```r
combined <- left_join(
  pca_df,
  reviews_df %>% select(doc_id, sentiment_label),
  by = "doc_id"
)


ggplot(combined, aes(x = kmeans_cluster, fill = sentiment_label)) +
  geom_bar(position = "fill") +
  labs(title = "K-means Clusters vs Sentiment", x = "Cluster", y = "Proportion") +
  scale_fill_manual(values = c("positive"="green", "neutral"="blue", "negative"="red")) +
  theme_minimal()
```

```r
library(ggplot2)


reviews_df$rating <- as.numeric(data_sample$rating)

ggplot(reviews_df, aes(x = rating)) +
  geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Ratings", x = "Rating", y = "Count of Reviews") +
  scale_x_continuous(breaks = seq(min(reviews_df$rating, na.rm = TRUE),
                                  max(reviews_df$rating, na.rm = TRUE), 0.5)) +
  theme_minimal()
```