SportMiner: A Comprehensive R Package for Sports Science Literature Mining and Topic Modeling

by Praveen D. Chougale and Usha Ananthakumar

Abstract SportMiner is an R package designed to facilitate systematic literature reviews and text mining in sports science research. It provides an integrated workflow from data collection through the Scopus API to advanced topic modeling and visualization. The package addresses the growing need for efficient tools to analyze the rapidly expanding body of sports science literature. Key features include automated literature retrieval, text preprocessing, multiple topic modeling algorithms (LDA, CTM, and STM), model comparison tools, and specialized visualizations including network analysis. SportMiner is designed for sports science researchers with varying levels of programming experience, offering both simplicity for basic analyses and flexibility for advanced users. The package is available on CRAN and actively maintained on GitHub.

1 Introduction

Sports science research has experienced exponential growth in recent decades, with thousands of papers published annually across diverse topics including exercise physiology, biomechanics, sports psychology, and performance analysis. This rapid expansion of literature presents both opportunities and challenges for researchers attempting to conduct comprehensive literature reviews and identify research trends.

Traditional manual literature review methods become increasingly impractical as the volume of publications grows. Systematic approaches to literature mining, including topic modeling and text mining techniques, offer powerful alternatives for identifying patterns, themes, and research gaps in large document collections. However, implementing these methods requires combining multiple R packages and APIs, presenting a barrier for many sports science researchers.

1.1 Existing Solutions and Limitations

Several R packages exist for bibliometric analysis and topic modeling, but none specifically address the needs of sports science researchers:

- **bibliometrix** (Aria and Cuccurullo, 2017) provides comprehensive bibliometric analysis but lacks domain-specific preprocessing and sports science-focused visualizations
- **topicmodels** (Grün and Hornik, 2011) offers topic modeling algorithms but requires extensive preprocessing and lacks integrated workflow
- **rscopus** (Muschelli, 2019) enables Scopus API access but provides no analysis capabilities
- **tidytext** (Silge and Robinson, 2016) facilitates text mining but requires substantial additional code for complete analyses

These packages require researchers to manually integrate multiple tools, write substantial custom code, and possess advanced R programming skills.

1.2 The SportMiner Solution

SportMiner fills this gap by providing an integrated, domain-aware workflow specifically designed for sports science literature analysis. The package combines:

1. **Seamless data collection** via the Scopus API with automatic rate limiting and error handling
2. **Domain-aware text preprocessing** optimized for sports science terminology
3. **Multiple topic modeling approaches** (LDA, CTM, STM) with automatic model comparison
4. **Specialized visualizations** including network analysis of research themes
5. **Reproducible workflows** with sensible defaults for common use cases

The target user is a sports science researcher with basic to intermediate R skills who needs to conduct systematic literature reviews or explore research trends. The package emphasizes ease of use while maintaining flexibility for advanced users.

2 Package Design and Architecture

2.1 Design Philosophy

SportMiner adheres to several key design principles:

Progressive disclosure: Common workflows require minimal code, while advanced options remain accessible through function parameters.

Fail-safe defaults: All functions include sensible default values based on sports science domain knowledge.

Informative feedback: Functions provide clear messages about processing steps and results.

Tidy integration: The package follows tidyverse principles and integrates seamlessly with ggplot2.

2.2 Dependency Considerations

SportMiner depends on well-established, actively maintained packages:

- **httr**: Robust HTTP requests for API access
- **dplyr** and **tidyr**: Data manipulation
- **tidytext**: Text mining operations
- **topicmodels**: LDA and CTM algorithms
- **stm**: Structural topic modeling
- **ggplot2**: Visualizations
- **igraph**: Network analysis

All dependencies are available on CRAN, minimizing installation issues. The package avoids deprecated or unmaintained dependencies.

2.3 Function Naming and Interface

All user-facing functions follow the `sm_*` prefix convention (SportMiner), making them easily discoverable through auto-completion. Function names use snake_case following tidyverse conventions:

- `sm_search_scopus()` - Search and retrieve literature
- `sm_preprocess_text()` - Clean and tokenize text
- `sm_create_dtm()` - Create document-term matrix
- `sm_fit_lda()` - Fit LDA topic model
- `sm_fit_ctm()` - Fit correlated topic model
- `sm_compare_models()` - Compare multiple models
- `sm_plot_topics()` - Visualize topic distributions
- `sm_create_network()` - Build co-occurrence networks

3 Typical Workflow

This section demonstrates the standard workflow for analyzing sports science literature using SportMiner.

3.1 Setting Up

```
# Install from CRAN
install.packages("SportMiner")

# Load package
library(SportMiner)

# Set Scopus API key (obtain from https://dev.elsevier.com/)
# Store in .Renviron file
Sys.setenv(SCOPUS_API_KEY = "your_api_key_here")
```

3.2 Step 1: Literature Retrieval

Search and retrieve papers from Scopus using domain-specific queries:

```
# Search for papers on mental fatigue in sports
papers <- sm_search_scopus(
  query = "mental fatigue AND (sport OR athlete OR exercise)",
  max_count = 500
)

# Returns data frame with title, abstract, authors, year, etc.
dim(papers)
```

The function handles pagination automatically and includes rate limiting to respect API constraints.

3.3 Step 2: Text Preprocessing

Clean and tokenize text data with sports science-appropriate preprocessing:

```
# Preprocess abstracts
processed <- sm_preprocess_text(
  data = papers,
  text_col = "abstract",
  min_word_length = 3,
  custom_stopwords = c("study", "research", "result")
)

# Returns word counts: doc_id, stem, n
head(processed)
```

3.4 Step 3: Document-Term Matrix Creation

Create a filtered document-term matrix suitable for topic modeling:

```
dtm <- sm_create_dtm(
  word_counts = processed,
  min_term_freq = 5,      # Term must appear in 5+ documents
  max_term_freq = 0.7     # Term in <70% of documents
)

# Dimensions: documents × terms
dim(dtm)
```

3.5 Step 4: Topic Modeling

Fit multiple topic models and compare performance:

```
# Fit LDA models with different topic counts
lda_5 <- sm_fit_lda(dtm, k = 5, seed = 42)
lda_10 <- sm_fit_lda(dtm, k = 10, seed = 42)
lda_15 <- sm_fit_lda(dtm, k = 15, seed = 42)

# Compare models
comparison <- sm_compare_models(
  list(k5 = lda_5, k10 = lda_10, k15 = lda_15)
)

print(comparison)
```

The `sm_compare_models()` function calculates perplexity and coherence metrics to guide model selection.

3.6 Step 5: Visualization

Create publication-ready visualizations:

```
# Plot top terms per topic
sm_plot_topics(lda_10, n_terms = 10)

# Visualize topic prevalence
sm_plot_topic_dist(lda_10, papers)

# Create co-occurrence network
network <- sm_create_network(processed, min_correlation = 0.3)
sm_plot_network(network)
```

All plots use ggplot2 and can be further customized using standard ggplot2 syntax.

3.7 Complete Workflow Example

A complete analysis from search to visualization:

```
library(SportMiner)

# 1. Retrieve papers
papers <- sm_search_scopus(
  "biomechanics AND running",
  max_count = 300
)

# 2. Preprocess
processed <- sm_preprocess_text(papers)

# 3. Create DTM
dtm <- sm_create_dtm(processed)

# 4. Fit model
model <- sm_fit_lda(dtm, k = 8, seed = 123)

# 5. Visualize
sm_plot_topics(model, n_terms = 15)
```

4 Implementation Details

4.1 Performance and Scalability

SportMiner is designed to handle typical sports science literature review scales (100-5000 documents):

Memory efficiency: Document-term matrices use the sparse matrix format from the slam package, reducing memory footprint by 90%+ compared to dense matrices.

Processing time: On a standard laptop (Intel i5, 8GB RAM): - 500 documents: ~2-3 minutes end-to-end - 2000 documents: ~8-10 minutes end-to-end - 5000 documents: ~20-25 minutes end-to-end

API rate limiting: Automatic delays between requests prevent Scopus API quota exhaustion.

4.2 Object-Oriented Design

SportMiner uses S3 methods for topic model objects:

```
# Generic plot method
plot(model)

# Generic summary method
summary(model)
```

```
# Print method with formatted output
print(model)
```

4.3 Error Handling and User Feedback

All functions include extensive error checking:

```
# Validates API key presence
sm_search_scopus(query)

# Checks for required columns
sm_preprocess_text(data, text_col = "abstract")

# Warns about filtering thresholds
sm_create_dtm(processed, min_term_freq = 100)
```

Functions use `message()` for informational output and `warning()`/`stop()` for issues, following R best practices.

4.4 Testing and Quality Assurance

SportMiner includes comprehensive unit tests using testthat:

- 95%+ code coverage
- Tests for all exported functions
- Tests for error conditions
- Tests for edge cases (empty data, API failures, etc.)

Continuous integration via GitHub Actions ensures tests pass on multiple R versions and operating systems.

5 Use Case: Mental Fatigue in Sports

This extended example demonstrates SportMiner's capabilities for addressing a research question: "What are the main research themes in mental fatigue and sports performance?"

```
library(SportMiner)
library(dplyr)

# Define comprehensive search query
query <- paste(
  "mental fatigue OR cognitive fatigue OR ego depletion",
  "AND",
  "(sport OR athlete OR exercise OR performance)",
  "AND",
  "(decision OR accuracy OR reaction OR endurance)"
)

# Retrieve papers (2015-2023)
papers <- sm_search_scopus(
  query = query,
  max_count = 1000,
  year_start = 2015,
  year_end = 2023
)

cat("Retrieved", nrow(papers), "papers\n")

# Preprocess with domain-specific stopwords
sport_stopwords <- c(
  "study", "research", "participant", "result",
```

```
  "conclusion", "method", "background", "objective"
)

processed <- sm_preprocess_text(
  papers,
  text_col = "abstract",
  custom_stopwords = sport_stopwords
)

# Create DTM with conservative filtering
dtm <- sm_create_dtm(
  processed,
  min_term_freq = 10,
  max_term_freq = 0.6
)

# Fit multiple models
models <- list()
for (k in c(6, 8, 10, 12)) {
  models[[paste0("k", k)]] <- sm_fit_lda(dtm, k = k, seed = 42)
}

# Compare and select best model
comparison <- sm_compare_models(models)
best_k <- comparison$k[which.min(comparison$perplexity)]

final_model <- models[[paste0("k", best_k)]]

# Visualize topics
p1 <- sm_plot_topics(final_model, n_terms = 12)
print(p1)

# Examine topic prevalence by year
papers$year <- as.numeric(format(papers$publication_date, "%Y"))
sm_plot_topic_trends(final_model, papers, time_var = "year")

# Create semantic network
network <- sm_create_network(
  processed,
  min_correlation = 0.4,
  top_n_terms = 50
)

sm_plot_network(
  network,
  layout = "fr",
  node_size = "degree",
  label_size = 3
)
```

This analysis reveals distinct research themes including cognitive load, decision-making under fatigue, endurance performance, and recovery strategies.

6 Discussion

6.1 Advantages Over Existing Tools

SportMiner offers several advantages:

1. **Integrated workflow**: No need to combine multiple packages manually
2. **Domain awareness**: Preprocessing and defaults optimized for sports science
3. **Ease of use**: Minimal code for complete analyses
4. **Reproducibility**: Seed parameters and version control ensure consistent results

5. **Extensibility**: Functions return standard objects that work with other R packages

6.2 Limitations

Current limitations include:

- Only supports Scopus API (PubMed support planned)
- Topic modeling limited to LDA, CTM, and STM
- Large datasets (>10,000 documents) may require extended processing time
- Requires API key for Scopus access

6.3 Future Development

Planned enhancements include:

- Additional data sources (PubMed, Web of Science)
- Interactive visualizations using plotly
- Shiny app for GUI-based analysis
- Advanced sentiment analysis for sports science text
- Multilingual support

7 Conclusion

SportMiner addresses a critical need in sports science research by providing an accessible, integrated toolset for literature mining and topic modeling. The package reduces the technical barriers to conducting systematic literature reviews, enabling researchers to focus on interpreting results rather than wrestling with code.

By combining Scopus API access, text preprocessing, multiple topic modeling algorithms, and specialized visualizations in a single package, SportMiner streamlines the entire literature analysis workflow. The package's design prioritizes ease of use while maintaining flexibility for advanced analyses.

SportMiner is actively maintained and welcomes community contributions through its GitHub repository (https://github.com/praveenmaths89/SportMiner). Users can report bugs, request features, or contribute code following standard open-source practices.

8 Acknowledgments

We thank the R community for developing the excellent packages upon which SportMiner is built, particularly the authors of tidytext, topicmodels, and stm.

9 References

References

M. Aria and C. Cuccurullo. bibliometrix: An r-tool for comprehensive science mapping analysis. *Journal of Informetrics*, 11(4):959–975, 2017. [p1]

B. Grün and K. Hornik. topicmodels: An r package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011. [p1]

J. Muschelli. *rscopus: Scopus Database 'API' Interface*, 2019. URL https://CRAN.R-project.org/package=rscopus. R package version 0.6.6. [p1]

J. Silge and D. Robinson. tidytext: Text mining and analysis using tidy data principles in r. *Journal of Open Source Software*, 1(3):37, 2016. [p1]

Praveen D. Chougale
IIT Bombay
Koita Centre for Digital Health, IIT Bombay, Mumbai, India
ORCiD: 0000-0002-5262-4726
praveenmaths89@gmail.com

Usha Ananthakumar
IIT Bombay
Shailesh J. Mehta School of Management, IIT Bombay, Mumbai, India
ORCiD: 0000-0003-1983-2168
usha@som.iitb.ac.in