Reproducible Evidence Synthesis

How to lit review

---

yacob.haddou@glasgow.ac.uk

https://valiriel.github.io

.../valiriel

https://orcid.org/0000-0003-0557-0339

Disclaimer & context

- I have not invented this, just reshaped it to fit my needs
- This is part of my PhD initial review, **first time** I do an "official" review

Zhi Foo et al. **(2021) A practical guide to question formation, systematic searching and study screening for literature reviews in ecology and evolution. Methods Ecol. Evol. DOI: 10.1111/2041-210X.13654**

Grames et al. **(2019) An automated approach to identifying search terms for systematic reviews using keyword co-occurrence networks. Methods Ecol. Evol. DOI: 10.1111/2041-210X.13268**

Start somewhere...

- Come up with your unambiguous research question, clearly defined.
- Get an idea of the *high citation* articles on your topic.
- **Seek out any review already done on your field !**
- Avoid overlaps with what has already been synthesized before by clarifying the timeframe, aim and search string of past reviews.
- Now come up with a naive search question.

Naive search question

1. Come up with a series of keywords that represent your question.
    - PICO method (Population, Intervention, Control, Outcome)
    - Think of it as Study system, Predictor variables and Response
    - Consider lateral/synonyms and vertical/specific/overarching terms
    - Check with peers that it all makes sense
2. Use boolean operator **(AND, OR, ...)** to put the terms together
3. Use truncation/stemming **(*)** to capture word with common root and plurals, be cautious though

Example naive search question

After having a look at past publications/reviews and what said above I came up with this series of term.

Not interested in a specific response or explanatory so kept it wide for anything including these terms.

("extinction debt/" OR "species credit/" OR "colonization credit/" OR "colonisation credit/" OR "immigration credit/" OR "ecosystem function debt/" OR "ecosystem service debt/" OR "ecosystem function credit/" OR "ecosystem service credit/" OR "ecological lags" OR "biodiversity lag/" OR "species lags" OR "landscape legacy")

substitute / with * (markdown does not interpret otherwise)

Time to do a search

- Pick some literature databases and query them.
    - Web of Science, BASE, SCOPUS, ScienceDirect etc...
- **Do** have a look at how many entries are returned, in the many 1000s you are probably asking something too broad and no one has time to go through all of that, honestly.
- Export the entries as .ris files

Install some packages

---

https://rmetaverse.github.io/

```
remotes::install_github("rmetaverse/metaverse")
remotes::install_github("elizagrames/litsearchr", ref = "main")
```

Packages aimed at evidence syntheses and meta-analyses

Import naive search results

```
naive_import <- import_results(directory = "lit_data/00_naive_lit_data/naive_search_dat
    verbose = TRUE)
print(paste0("imported n. ", nrow(naive_import)))
```

910 records from my naive search

De-duplicate

Based on exact match and string similarity of title

```
naive_deduplicated <- remove_duplicates(naive_import, field = "title", method = "exact"
print(paste0("deduplicated_exact n. ", nrow(naive_deduplicated)))

naive_deduplicated <- remove_duplicates(naive_deduplicated, field = "title", method = "
print(paste0("deduplicated_string n. ", nrow(naive_deduplicated)))
```

810 and then 799 remaining, 111 records gone

Extract keywords

```
rake_keywords <-
  extract_terms(text = paste(naive_deduplicated$title, naive_deduplicated$abstract,
                             naive_deduplicated$keywords),
                method = "fakerake",
                min_freq = 10, # at least 10 times
                ngrams = TRUE, min_n = 2, # minimum 2 words keywords
                language = "English",
                stopwords = get_stopwords("English")) %>% as_tibble()

# remove terms that are too broad, save to csv (ver1)
# and then save for re-import (ver2)
write.csv(rake_keywords, file = "lit_data/00_naive_lit_data/naive_search_keywords_ver1.
```

Remove broad keywords and co-occurence network

```
#' obsolete or too broad terms manually removed, such as keywords related to specific
#' habitat or taxonomic groups, broad words ('biodiversity', population size) and unrel
rake_keywords <- read.csv("lit_data/00_naive_lit_data/naive_search_keywords_ver2.csv")[
    -1]

#' create document-feature matrix
naive_dfm <- create_dfm(elements = paste(naive_deduplicated$title, naive_deduplicated$a
    features = rake_keywords)

#' create a keyword co-occurrence network
naive_graph <- create_network(search_dfm = naive_dfm, min_studies = 5, min_occ = 5)

#' evaluate keyword importance and clean
nodes_rank <- strength(naive_graph)

terms_rank <- tibble(term = names(nodes_rank), strength = nodes_rank, row.names = NULL)
    mutate(rank = rank(strength, ties.method = "min")) %>%
    arrange(strength)
```

Enhance,enhance

- Note that the final keyword order might not always be so useful, words such us biodiversity could be high ranking but not necessarily good.

Manually inspect the list and see if you find any keywords that could fit in your original naive string, if yes include them and remember to truncate.

Final search string

("extinction debt/" OR "species credit/" OR "colonization credit/" OR "colonisation credit/" OR "immigration credit/" OR "ecosystem function debt/" OR "ecosystem service debt/" OR "ecosystem function credit/" OR "ecosystem service credit/" OR "ecological lags" OR "biodiversity lag/" OR "species lag/" OR "landscape legac/")

+

"community lag/" OR "emigration debt/" OR "delayed species extinction/" OR "species debt/" OR "delayed species colonisation/" OR "delayed species colonization/"

Loop one more time through the above steps

1. Search the lit databases with your final string
    - 966 entries retrieved
2. De-duplicate
    - 830 articles remaining

Manual screening (part 1)

```
# save a copy of raw data, 830 articles
write.csv(deduplicated, file = "lit_data/02_title_abstract_screening/00_lit_not_screene

# screen titles
revtools::screen_titles(deduplicated)
# manually move saved file from main dir to
# lit_data/02_title_abstract_screening as 01_lit_title_screened.csv
data <- read.csv("lit_data/02_title_abstract_screening/01_lit_title_screened.csv")

# retain selection from above, remove excluded
data <- data %>%
    filter(screened_titles != "excluded")
table(data$screened_titles)
table(data$source_type)
table(data$year)

# screen abstract
screen_abstracts(data %>%
```

Manual screening (part 2)

```
# import again
data2 <- read.csv("lit_data/02_title_abstract_screening/02_lit_abstract_screened.csv")

# retain selection from above, remove excluded
data2 <- data2 %>%
    filter(screened_abstracts != "excluded")
table(data2$screened_abstracts)
table(data2$source_type)
table(data2$year)

# retain peer-reviewed articles
data2 <- data2 %>%
    filter(source_type == "JOUR")

# see trend through time
plot(names(table(data2$year)), as.numeric(table(data2$year)), type = "l")

#' Go through each article and import into reference manager for reading
```

Manual screening (part 3)

Some reference numbers:

- 830 to begin with
- 289 left after screening titles
- 208 left after screening abstracts
- 196 left after keeping only peer-reviewed material
- 190 after sourcing and final clean-up

Make it reproducible and more

Show R code, GitHub, Zotero

Reading time

Constructed a series of variables and criteria to quantitatively assess each article and then report

Show csv

_ yacob.haddou@glasgow.ac.uk

_ https://github.com/valiriel

_ https://orcid.org/0000-0003-0557-0339

_ https://www.researchgate.net/profile/Yacob-Haddou

slides powered by xaringan https://github.com/yihui/xaringan