7 Building Reproducible Corpora

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

7.1 Learning objectives

After completing this chapter, you will be able to:

- Deduplicate bibliographic records by DOI and by approximate title matching
- Resolve identifiers across databases (OpenAlex ID, DOI, PMID)
- Recognize author name disambiguation challenges and apply basic heuristics
- Standardize institutional affiliations using ROR identifiers
- Save a clean corpus in an efficient format (Parquet) for downstream analysis

7.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(arrow)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

7.3 Conceptual background

Raw bibliographic data is messy. Records from different sources may describe the same paper with different metadata, the same author under different name variants, and the same institution under dozens of spelling variations. Building a clean, reproducible corpus requires systematic attention to three problems:

Deduplication. When merging data from multiple sources, the same publication may appear multiple times. DOI-based deduplication is the gold standard, but not all records have DOIs. Fuzzy title matching provides a fallback but is error-prone for short or generic titles.

Author disambiguation. "J. Smith" could be dozens of different people. OpenAlex uses machine-learning-based author clustering to assign persistent author IDs, but errors persist — especially for common names and authors who change institutions (Priem et al. 2022).

Affiliation standardization. Institutional names appear in countless variants ("MIT", "Massachusetts Institute of Technology", "Mass. Inst. Tech."). The Research Organization Registry (ROR) provides a curated set of persistent identifiers for research organizations.

OpenAlex maps many affiliations to ROR IDs, but coverage is incomplete.

7.4 Worked example

7.4.1 Fetching raw data

```
works ← fetch_openalex(
  entity = "works",
  search = "bibliometrics",
  from_publication_date = "2021-01-01",
  to_publication_date = "2023-12-31",
  options = list(sample = 300, seed = 42)
)

cat(glue("Raw records: {nrow(works)}\n"))
```

```
#> Raw records: 300
```

7.4.2 DOI-based deduplication

```
works_deduped ← dedupe_by_doi(works)
cat(glue("After DOI dedup: {nrow(works_deduped)}\n"))
```

```
#> After DOI dedup: 300
```

```
cat(glue("Removed: {nrow(works) - nrow(works_deduped)} duplicates\n"))
```

```
#> Removed: 0 duplicates
```

7.4.3 Fuzzy title deduplication

For records without DOIs, we use approximate string matching.

```
no_doi ← works_deduped ▷ filter(is.na(doi))

if (nrow(no_doi) > 1) {
  title_lower ← tolower(no_doi$display_name)
  dist_matrix ← stringdist::stringdistmatrix(title_lower, method = "jw")
  potential_dupes ← which(as.matrix(dist_matrix) < 0.1 &
                          as.matrix(dist_matrix) > 0, arr.ind = TRUE)
  potential_dupes ← potential_dupes[potential_dupes[, 1] < potential_dupes[, 2], ,
drop = FALSE]
  cat(glue("Potential fuzzy duplicates (no DOI): {nrow(potential_dupes)} pairs\n"))
} else {
  cat("No records without DOI to fuzzy-match.\n")
}
```

```
#> Potential fuzzy duplicates (no DOI): 0 pairs
```

7.4.4 Affiliation extraction and ROR mapping

```
affiliations ← works_deduped ▷
  select(id, authorships) ▷
  unnest(authorships, names_sep = "_") ▷
  unnest(authorships_affiliations, names_sep = "_") ▷
  select(work_id = id,
         author_name = authorships_display_name,
         institution = authorships_affiliations_display_name) ▷
  filter(!is.na(institution))

top_institutions ← affiliations ▷
  count(institution, sort = TRUE) ▷
  head(15)

top_institutions
```

```
#> # A tibble: 15 × 2
#>    institution                                    n
#>    <chr>                                      <int>
#>  1 Changhai Hospital                             25
#>  2 Sumy State University                         10
#>  3 Third Hospital of Hebei Medical University    10
#>  4 First Affiliated Hospital of Jinan University  9
#>  5 Universidad de Extremadura                     9
#>  6 Universitat de València                        9
#>  7 University of Kansas Medical Center            9
#>  8 Aston University                               8
#>  9 Lanzhou University                             8
#> 10 Politecnico di Milano                          8
#> 11 Renmin Hospital of Wuhan University            8
#> 12 Universidad Rey Juan Carlos                    8
#> 13 University of the East                         8
#> 14 General Hospital of Shenyang Military Region   7
#> 15 Hunan University                               7
```

7.4.5 Saving as Parquet

```
corpus_clean ← works_deduped ▷
  select(id, display_name, publication_date, cited_by_count, doi, source_display_name,
abstract)

out_dir ← here::here("_freeze", "exports")
dir.create(out_dir, recursive = TRUE, showWarnings = FALSE)
out_path ← file.path(out_dir, "corpus_sample.parquet")
write_parquet(corpus_clean, out_path)
cat(glue("Saved {nrow(corpus_clean)} records to {out_path}\n"))
```

```
#> Saved 300 records to /home/runner/work/scientometrics-in-r/scientometrics-in-
r/_freeze/exports/corpus_sample.parquet
```

7.4.6 Visualization

```
top_institutions ▷
  mutate(institution = fct_reorder(institution, n)) ▷
  ggplot(aes(x = n, y = institution)) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Publications", y = NULL) +
  theme_sci()
```

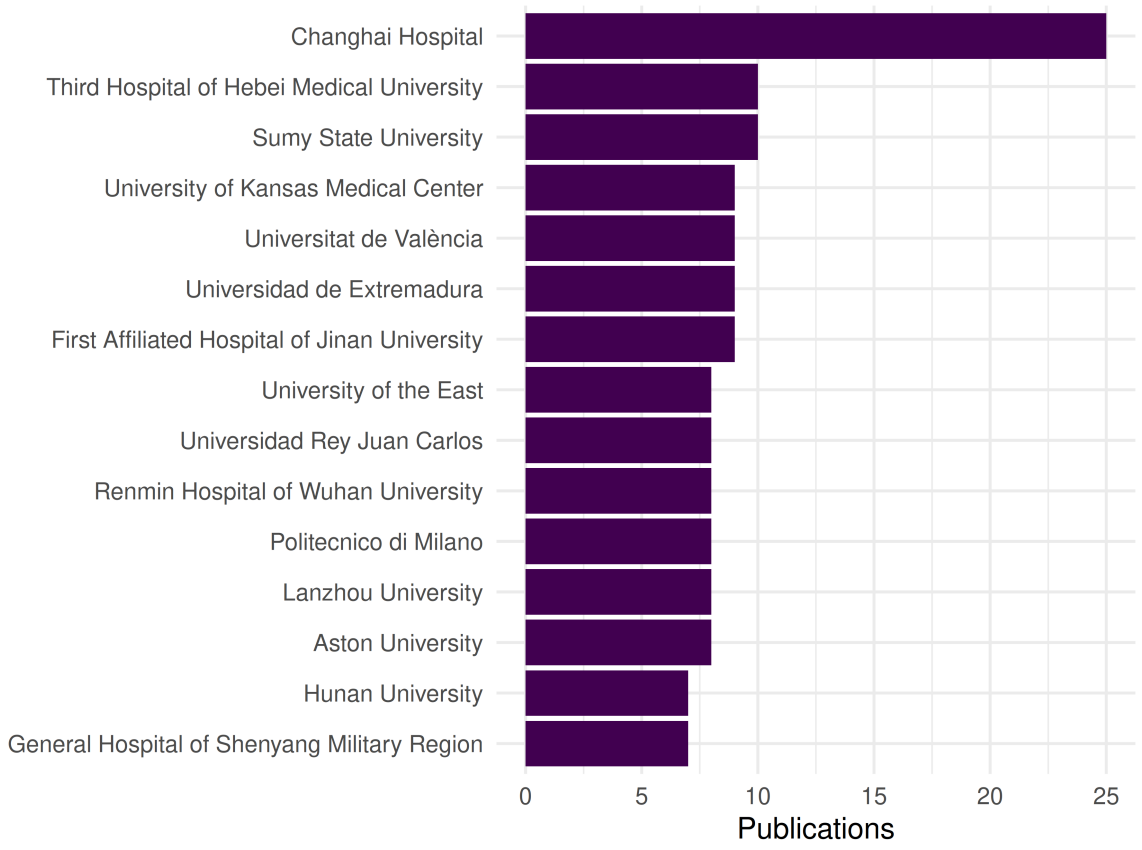


Figure 7.1: Top 15 institutions by publication count in the sample corpus.

7.5 Diagnostics and interpretation

After cleaning:

- **Deduplication rate**: A rate above 5% suggests either a multi-source merge or a query that returns overlapping result pages.
- **DOI coverage**: The proportion of records with DOIs determines how effective DOI-based deduplication can be.
- **Affiliation coverage**: Check what fraction of author records have institutional affiliations. Low coverage limits institutional analysis.
- **Parquet file size**: Parquet is columnar and compressed. A 10,000-record corpus typically compresses to under 1 MB.

7.6 Limitations and responsible use

7.7 Limitations and responsible use

- **Fuzzy matching is imperfect.** Jaro-Winkler similarity can produce both false positives (different papers with similar titles) and false negatives (same paper with different title versions). Always spot-check.
- **Author disambiguation remains unsolved.** No automated method achieves perfect accuracy. Report the disambiguation method used and its known error rate.
- **Affiliation data is noisy.** Even with ROR mapping, temporary affiliations, joint appointments, and name changes create ambiguity. Never use affiliation data as ground truth without validation (Hicks et al. 2015).

7.8 Common pitfalls

7.9 Common pitfalls

- **Deduplicating before merging.** Always merge first, then deduplicate. Deduplicating each source independently misses cross-source duplicates.
- **Trusting DOIs blindly.** Some records have incorrect DOIs (typos, test DOIs, placeholder values). Validate a sample.
- **Ignoring records without DOIs.** Discarding them biases the corpus toward recent, well-indexed publications.
- **Not documenting cleaning steps.** Every deduplication or name-standardization step should be logged for reproducibility.

7.10 Exercises

1. **DOI coverage by year.** Compute the DOI coverage rate by publication year in the sample. Is there a trend?

2. **Name variants.** Find authors in the sample who appear under multiple name spellings. What heuristics could you use to merge them?

3. **ROR lookup.** Use the ROR API to resolve the top 5 institutions in your corpus to their official ROR identifiers.

7.11 Solutions

Solutions are provided in E.

7.12 Further reading

- Priem et al. (2022) — OpenAlex author disambiguation and institutional mapping.
- Aria and Cuccurullo (2017) — `bibliometrix` data cleaning utilities.
- Hicks et al. (2015) — The Leiden Manifesto; principles 4 and 5 emphasise data transparency and verifiability.

7.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.26.so;  LAPACK
version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C           LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8    LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] arrow_26.0.1.1     bibliometrix_5.5.0 RefManageR_1.4.0   bib2df_1.1.2.0
#>  [5] rcrossref_1.2.1    gt_1.4.0           tidytext_0.4.3     glue_1.8.1
#>  [9] openalexR_3.1.2    lubridate_1.9.5    forcats_1.0.1      stringr_1.6.0
#> [13] dplyr_1.2.1        purrr_1.2.2        readr_2.2.0        tidyr_1.3.2
#> [17] tibble_3.3.1       ggplot2_4.0.4      tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] gridExtra_2.3.1        httr2_1.3.0            readxl_1.6.0
#>   [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>   [7] compiler_4.6.1         vctrs_0.7.3            httpcode_0.3.0
#>  [10] pkgconfig_2.0.3        fastmap_1.2.0          labeling_0.4.3
#>  [13] ca_0.72                utf8_1.2.6             promises_1.5.0
#>  [16] rmarkdown_2.32         tzdb_0.5.0             bit_4.6.0
#>  [19] xfun_0.62              cachem_1.1.0           jsonlite_2.0.0
#>  [22] SnowballC_0.7.1        later_1.4.8            parallel_4.6.1
#>  [25] stopwords_2.3          R6_2.6.1               bslib_0.12.0
#>  [28] stringi_1.8.9          RColorBrewer_1.1-3     cellranger_1.1.0
#>  [31] jquerylib_0.1.4        assertthat_0.2.1       Rcpp_1.1.2
#>  [34] bookdown_0.48          knitr_1.52             triebeard_0.4.1
#>  [37] base64enc_0.1-6        rentrez_1.2.4          httpuv_1.6.17
#>  [40] Matrix_1.7-5           igraph_2.3.4           timechange_0.4.0
#>  [43] tidyselect_1.2.1       stringdist_0.9.17      pubmedR_1.1.0
#>  [46] rstudioapi_0.19.0      yaml_2.3.12            viridis_0.6.5
#>  [49] miniUI_0.1.2           humaniformat_0.6.0     curl_8.1.0
#>  [52] qpdf_1.4.1             lattice_0.22-9         plyr_1.8.9
#>  [55] shiny_1.15.0           withr_3.0.3            S7_1.0.0
#>  [58] askpass_1.2.1          evaluate_1.0.5         zip_3.0.2
#>  [61] xml2_1.6.0             shinycssloaders_1.1.0  pillar_1.11.1
#>  [64] janeaustenr_1.0.0      DT_0.34.0              plotly_4.12.1
#>  [67] generics_0.1.4         rprojroot_2.1.1        hms_1.1.4
#>  [70] scales_1.4.0           xtable_1.8-8           contentanalysis_1.1.1
#>  [73] tools_4.6.1            data.table_1.18.6.1    tokenizers_0.3.0
#>  [76] openxlsx_4.2.9         pdftools_3.9.1         XML_3.99-0.25
#>  [79] fs_2.1.0               visNetwork_2.1.4       grid_4.6.1
#>  [82] bibtex_0.5.3           urltools_1.7.3.1       rscopus_0.9.0
#>  [85] dimensionsR_0.0.3      bibliometrixData_0.3.0 cli_3.6.6
#>  [88] viridisLite_0.4.3      downlit_0.4.5          gtable_0.3.6
#>  [91] sass_0.5.0             digest_0.6.39          ggrepel_0.9.8
#>  [94] crul_1.6.0             htmlwidgets_1.6.4      farver_2.1.2
#>  [97] memoise_2.0.1          htmltools_0.5.9        lifecycle_1.0.5
```

```
#> [100] httr_1.4.9             here_1.0.2             mime_0.13
#> [103] bit64_4.8.6
```