The Journal of Open Source Software

BibDedupe: An Open-Source Python Library for Bibliographic Record Deduplication

Gerit Wagner[1]

1 Otto-Friedrich Universität Bamberg

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 22 January 2024
Published: unpublished

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

DRAFT

Summary

BibDedupe is a Python library developed for bibliographic record deduplication in meta-analysis and research synthesis. It is constructed with a focus on four requirements: (1) **Zero false positives**: The primary objective is to prevent incorrectly merging distinct entries. This focus on zero false positives is crucial to ensure trustworthiness and prevent biased conclusions in the analysis. (2) **Reproducibility**: BibDedupe implements fixed rules to produce consistent results, in line with the scientific standard of reproducibility. (3) **Efficiency**: The library is also tuned for low false-negative rates and rapid processing, to ensure scalability of the duplicate identification process. (4) **Continuous evaluation and improvement**: It is continuously evaluated on over 160,000 records from 10 datasets to ensure its effectiveness, especially in follow-up refinements. Unlike general-purpose deduplication tools, BibDedupe is specifically designed for the unique requirements of bibliographic data in meta-analysis and research synthesis. In this context, BibDedupe aims to provide a Python library that significantly improves the effectiveness and efficiency of duplicate identification, potentially benefitting standalone review papers across scientific disciplines.

Statement of Need

Handling duplicates is a critical step in meta-analysis and research synthesis (Harrer et al., 2021), given that errors in this step can directly affect conclusions (Wood, 2008). Prior research has invested considerable efforts to evaluate duplicate identification software for bibliographic data (Binette & Steorts, 2022; Bramer et al., 2016; Koumarelas et al., 2020; Rathbone et al., 2015). While methodologists have repeatedly cautioned against the risk of treating identical studies independently when they are published in different papers (Fairfield et al., 2017; Senn, 2009), the risk of erroneously classifying papers as duplicates has arguably received less attention. However, once removed from the process, it is rarely possible to recover false positives, or to quantify and correct their effect on meta-analytic results. As such, preventing false positives is of critical importance, while false negatives can be detected and merged in the subsequent screening and analysis steps (McLoughlin, 2022).

Proprietary research software for duplicate identification often suffers from shortcomings related to the four requirements. Relying on tools like Endnote, Covidence, or Hubmeta requires compromises related to false positives, limited transparency of black-box algorithms, or the lack of peer-review and external validation. Moreover, the use of proprietary tools often incurs costs, and restricts the combination of research tools, because data is hard to access and programmatic interfaces are not offered.

General purpose deduplication libraries often lack the specificity needed for bibliographic data, requiring skills and excessive amounts of effort to develop and evaluate algorithms. For example, libraries such as the *Python Record Linkage Toolkit* (De Bruin, 2019) and *dedupe (io)* (Gregg & Eder, 2022) provide an arsenal of similarity measures, blocking rules, and utility functions.

As such, they provide a valuable basis to support the design of domain-specific duplicate identification tools, but they are rarely used directly by researchers conducting a meta-analysis (Nguyen et al., 2022). When developing a custom deduplication algorithm, its effectiveness can only be evaluated by creating an independently deduplicated dataset. More severely, developing an accurate algorithm require in-depth knowledge of publication practices and errors (meta data variation) typically introduced by academic databases, or other systems handling bibliographic metadata. Experience shows that minor changes potentially have significant effects on overall performance. Finally, machine-learning libraries, such as *dedupe (io)*, involve the learning of blocking rules and similarity functions from each dataset, and based on user input. This limits reproducibility, and the required manual efforts reduce efficiency.

Open-source research software for duplicate identification is scarce, and to-date, peer-reviewed software is non-existent in this area. In the Python ecosystem, the only library we found is *ASReview Datatools*, provided by the team behind the *ASReview* screening tool (Van De Schoot et al., 2021). Our evaluations show that this library introduces a considerable number of false positives, and cannot be used for meta-analyses. R users or Python users willing to switch the ecosystem, may use ASySD (Hair et al., 2023), a recently published R package with a Shiny web interface. The code of this package resembles BibDedupe, but it does not achieve zero-false-positives, uses a relatively small test dataset from medicine (n=1845) in the unit tests, and was not evaluated in the peer review process.

In conclusion, researchers are not well-served by currently available proprietary tools, or general purpose deduplication libraries. Effetive and peer-reviewed libraries are urgently needed for meta-analyses and research synthesis to facilitate researchers' trust and adoption of open-source libraries in the area of literature reviews.

Example usage

```python
import pandas as pd
from bib_dedupe.bib_dedupe import merge

# Load your bibliographic dataset into a pandas DataFrame
records_df = pd.read_csv("records.csv")

# Get the merged_df
merged_df = merge(records_df)
```

For advanced use cases, it is also possible to complete and customize each step individually

```python
 from bib_dedupe.bib_dedupe import prep, block, match, merge
 from bib_dedupe.bib_dedupe import export_maybe, import_maybe

# Preprocess records
records_df = prep(records_df)
# Block records
blocked_df = block(records_df)
# Identify matches
matched_df = match(blocked_df)
# Export maybe cases
export_maybe(matched_df)
# Import maybe cases
import_maybe(matched_df)
# Merge
merged_df = merge(records_df, matched_df=matched_df)
```

DRAFT

Implementation

We define duplicates as potentially differing bibliographic representations of the same real-world record (cf. Rathbone et al., 2015). This conceptual definition is operationalized as follows. The following are considered **duplicates**:

- Papers referring to the same record (per definition)
- Paper versions, including the author's original, submitted, accepted, proof, and corrected versions (NISO/ALPSP JAV Working Group and others, 2008)
- Papers that are continuously updated (e.g., versions of Cochrane reviews)
- Papers with different DOIs if they refer to the same record (e.g., redundantly registered DOIs for online and print versions)

The following are considered **non-duplicates**:

- A conference paper and its extended journal publication
- A journal paper and a reprint in another journal
- Papers reporting on the same study if they are published separately (e.g., involving different stages of the study such as pilots and protocols, or differences in outcomes, interventions, or populations)

These clarifications are necessary for the evaluation dataset, and for users to understand what will (not) be considered a duplicate. The rationale is that cases of duplicates are rarely or never cited as separate items in a reference section, while non-duplicates can in principle be cited separately. It is a different issue whether the corresponding research and administrative practices are considered questionable or ethical (e.g., salami publications, or registering multiple DOIs for the same paper).

To accurately identify and merge duplicates, BibDedupe implements the steps of preprocessing, blocking, rule-based matching, and merging. As seen in the usage example, each step can be adapted.

Preprocessing

Preprocessing involves an array of standardizations across fields, including replacement of special characters. For titles and journals, stop words are removed to give more weight to distinctive words in the similarity measures. For the author field, name particles are removed because they are often handled incorrectly in the data creation process. Additional notes and translations are removed from the title field. For translated journal names, the English version is used as a replacement.

Blocking

To avoid checking all possible combinations of papers, blocking selects the pairs that are likely to be duplicates. This is a common technique in deduplication where only records within the same block are compared for potential duplication.

BibDedupe relies on a comprehensive set of blocking rules to avoid false negatives in this step. After the set of blocking rules is applied, we remove pairs not sharing a minimum number of words in the titles, effectively reducing the number of pairs by 50-95% without losing true pairs. This leads to a more efficient matching step.

Matching

The matching function selects duplicates or potential duplicates from the list of blocked record pairs. Potential duplicates, also known as "maybe cases", are marked separately for manual verification. To achieve accurate and interpretable matching, we specified an array of

DRAFT

DRAFT

human-readable conditions, which are based on pre-calculated and context-specific similarities between fields.

The conditions and similarity functions account for bibliographic errors commonly introduced between duplicates. We summarize the key design decisions of BibDedupe, which differ from other approaches (notably ASySD):

- **Robust author similarities**: The most substantial format variation is observed in the author field, requiring robust similarity measures. This is particularly challenging for non-Western names, which are not supported well by current citation style conventions, or name-parsing software (see nameparser). Given that Chinese authors are leading in many research output and impact rankings (Brainard & Normile, 2022), this is a limitation. After testing multiple similarity measures, we found that the agreement between capital or beginning-of-word letters provided the most robust measure of author similarity, suggesting that common similarity measures like Jaro-Winkler are less appropriate in this case. We briefly illustrate this with an example of non-Western names that were erroneously abbreviated:

```
Author string 1: "Chen J. M.Gong X. Q.Zhong J. G.Chen S. C.Zhang G. Y."
Author string 2: "Jin-Ming C.Xiao-Qi G.Ji-Gen Z.Si-Cong C.Guo-Yuan Z."

Jaccard similarity          : 0.18
Cosine similarity           : 0.31
Jaro-Winkler similarity     : 0.64
First-letters similarity    : 1.0
```

- **Sensitive title similarities**: For titles, similarity measures must be sensitive to minor differences between non-duplicates, as exemplified in so-called *salami-publications* or publications consisting of multiple parts. In these cases, titles are almost identical, and general similarity measures yield values close to 1, i.e., they are not sensitive enough to differences that are significant in the context of bibliographic data. BibDedupe implements a similarity function that is sensitive to differences in numbers (e.g., part 1 vs. part 2), populations (e.g., men vs. women, in vivo vs. in vitro, cats vs. rats), interventions (e.g., effect of X vs. effect of Y), and outcomes (e.g., effect on X vs. effect on Y).
- **Translations of container titles**: Given the nested data structure, in which papers are contained in journals, proceedings, or other containers, accurate matching is required for the field of container titles. To accomplish this, BibDedupe uses a list of approx. 1,300 translated journal names as replacements in the preprocessing step, effectively increasing the average Jaro-Winkler similarity between journals and their translated titles from 0.45 to 1.0. This leads to a substantial improvement in false negatives.
- **Handling missing values**: While values author, title, and container_title fields are rarely missing, there can be missing values in the other fields, such as the volume, DOI, or abstract. Similarity measures typically return insufficient results when only one value is missing. For instance, when one paper contains a DOI and the other does not, the similarity would be zero, as it would be the case for different DOIs. We distinguish these cases based on a `non_contradictory()` function, which is robust against missing values, and indicates whether non-missing values differ between records.

We note that global IDs (like DOIs) contribute to duplicate identification, but neither are identical DOIs considered a sufficient condition for a duplicate, nor are distinct DOIs considered a sufficient condition for non-duplicates. This is confirmed by the data. For the iterative tuning, we designed diagnostic utilities to assess which conditions match for selected (FP/FN) cases.

DRAFT

Merging

Upon merging a set of records, BibDedupe keeps track of the original IDs in the *origin* field. Compared to the common approach of deleting n-1 records from the set of duplicates, this approach has three distinct advantages: (1) **validation**: together with the original dataset, it allows users to validate whether duplicate decisions are accurate, (2) **undo**: it is possible to restore selected cases where erroneous duplicates were merged, and (3) **evaluation**: it enables subsequent use of datasets to evaluate and tune duplicate detection algorithms.

The merging function uses heuristics to select the most appropriate fields from duplicate records instead of selecting all fields from one record regardless of field-level quality. For instance, proper capitalization is preferred when one record has author or title fields in all-caps, and DOIs are selected when other DOI fields are empty.

Evaluation

To evaluate BibDedupe, we collected 10 datasets comprising over 160,000 records and 34,900 duplicates (Hair et al., 2023; Rathbone et al., 2015; Wagner et al., 2021). This is, to the best of our knowledge, the most comprehensive evaluation of bibliographic duplicate detection algorithms to date. We completed over 3,000 iterations to evaluate and improve BibDedupe based on these datasets.

The efforts involved tuning the preprocessing, blocking, and matching steps, vetting different similarity measures, and validating the false positives and negatives based on the definition of (non)-duplicates. We carefully reviewed the conditions to combine and generalize narrowly defined cases. In addition, we implemented unit tests to ensure consistency, and understand how changes in the code affect each step. Runtime was optimized by implementing and evaluating different approaches to parallel processing, such as processing NumPy-arrays vs. splitting dataframes horizontally. As a result, the depression dataset with approx. 80,000 records is processed in under 10 minutes with 8 CPUs.

Package	FP	TP	FN	TN	FP rate	Specificity	Sensitivity	F1
BibDedupe	0	35035	227	125537	0	1.0	0.99	0.997
asreview	5596	29749	5469	119972	0.04	0.96	0.84	0.84

Given that ASySD and other tools were evaluated on a subset of the datasets contained in our evaluation (Hair et al., 2023), it is instructive to re-print the results for comparison.

Package	FP	TP	FN	TN	FP rate	Specificity	Sensitivity	F1
Endnote	11	12466	3657	77977	0.0	1.0	0.77	0.87
Human	54	14693	1430	77934	0.0	1.0	0.91	0.95
ASySD	80	15664	459	77908	0.0	1.0	0.97	0.98
SRA-DM	1735	12794	3329	76253	0.02	0.98	0.79	0.83

Ongoing improvements

BibDedupe provides duplicate identification functionality, which performs with zero false positives on a dataset comprising over 160,000 records. It builds on carefully crafted rules and high-quality training data to ensure effectiveness, transparency, and reproducibility. The evaluation runs automatically and provides a solid foundation for continuous improvements and additions of datasets. We intend to incorporate additional datasets and continue refining the rules and procedures.

DRAFT

References

Binette, O., & Steorts, R. C. (2022). (Almost) all of entity resolution. *Science Advances*, *8*(12), eabi8021. https://doi.org/10.1126/sciadv.abi8021

Brainard, J., & Normile, D. (2022). China rises to first place in most cited papers. *Science*, *377*(6608), 799. https://doi.org/10.1126/science.ade4585

Bramer, W. M., Giustini, D., Jonge, G. B. de, Holland, L., & Bekhuis, T. (2016). Deduplication of database search results for systematic reviews in EndNote. *Journal of the Medical Library Association*, *104*(3), 240. https://doi.org/10.3163/1536-5050.104.3.014

De Bruin, J. (2019). *Python record linkage toolkit: A toolkit for record linkage and duplicate detection in python* (Version v0.14). Zenodo. https://doi.org/10.5281/zenodo.3559043

Fairfield, C. J., Harrison, E. M., & Wigmore, S. J. (2017). Duplicate publication bias weakens the validity of meta-analysis of immunosuppression after transplantation. *World Journal of Gastroenterology*, *23*(39), 7198. https://doi.org/10.3748/wjg.v23.i39.7198

Gregg, F., & Eder, D. (2022). *dedupe* (Version 2.0.11). https://github.com/dedupeio/dedupe

Hair, K., Bahor, Z., Macleod, M., Liao, J., & Sena, E. S. (2023). The automated systematic search deduplicator (ASySD): A rapid, open-source, interoperable tool to remove duplicate citations in biomedical systematic reviews. *BMC Biology*, *21*(1), 189. https://doi.org/10.1186/s12915-023-01686-z

Harrer, M., Cuijpers, P., Furukawa, T., & Ebert, D. (2021). *Doing meta-analysis with r: A hands-on guide*. Chapman; Hall/CRC.

Koumarelas, I., Jiang, L., & Naumann, F. (2020). Data preparation for duplicate detection. *Journal of Data and Information Quality*, *12*(3), 1–24. https://doi.org/10.1145/3377878

McLoughlin, R. (2022, March 7). *Improving our deduplication process*. Covidence. https://www.covidence.org/blog/improving-our-deduplication-process/

Nguyen, P.-Y., Kanukula, R., McKenzie, J. E., Alqaidoom, Z., Brennan, S. E., Haddaway, N. R., Hamilton, D. G., Karunananthan, S., McDonald, S., Moher, D., & others. (2022). Changing patterns in reporting and sharing of review data in systematic reviews with meta-analysis of the effects of interventions: Cross sectional meta-research study. *Bmj*, *379*. https://doi.org/10.1136/bmj-2022-072428

NISO/ALPSP JAV Working Group and others. (2008). *NISO-RP-8-2008, journal article versions (JAV): recommendations*.

Rathbone, J., Carter, M., Hoffmann, T., & Glasziou, P. (2015). Better duplicate detection for systematic reviewers: Evaluation of systematic review assistant-deduplication module. *Systematic Reviews*, *4*, 1–6. https://doi.org/10.1186/2046-4053-4-6

Senn, S. J. (2009). Overstating the evidence–double counting in meta-analysis and related problems. *BMC Medical Research Methodology*, *9*, 1–7. https://doi.org/10.1186/1471-2288-9-10

Van De Schoot, R., De Bruin, J., Schram, R., Zahedi, P., De Boer, J., Weijdema, F., Kramer, B., Huijts, M., Hoogerwerf, M., Ferdinands, G., & others. (2021). An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence*, *3*(2), 125–133. https://doi.org/10.1038/s42256-020-00287-7

Wagner, G., Prester, J., & Paré, G. (2021). Exploring the boundaries and processes of digital platforms for knowledge work: A review of information systems research. *The Journal of Strategic Information Systems*, *30*(4), 101694. https://doi.org/10.1016/j.jsis.2021.101694

Wood, J. (2008). Methodology for dealing with duplicate study effects in a meta-analysis. *Organizational Research Methods*, *11*(1), 79–95. https://doi.org/10.1177/1094428106296638

DRAFT