Report on Matching of Unpaywall and Web of Science

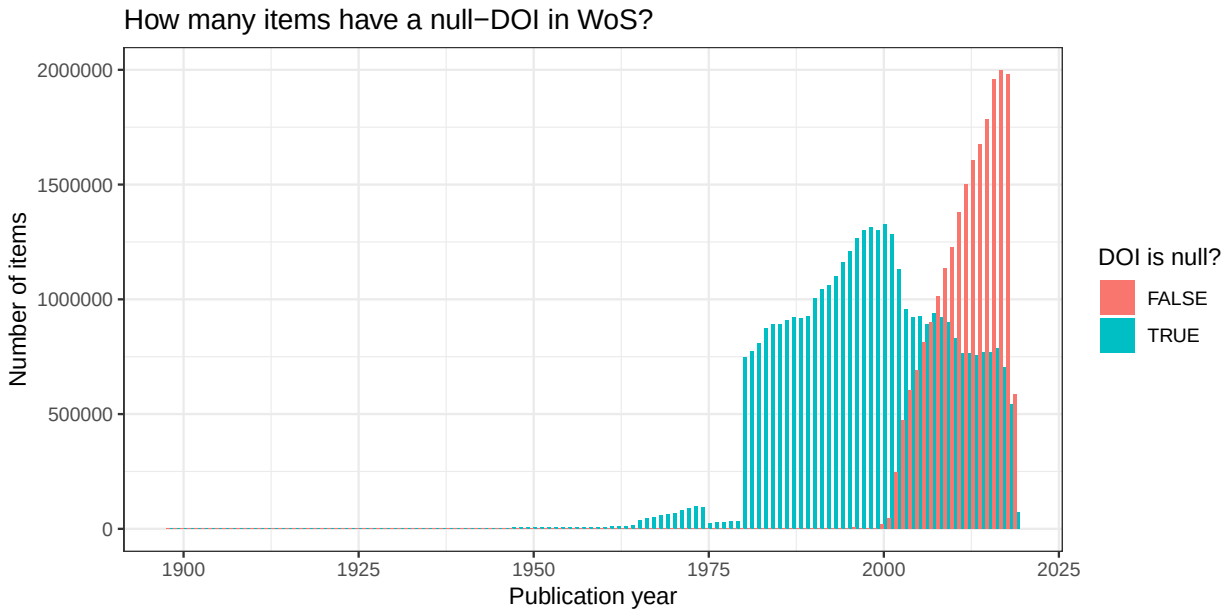
Summary

Combining data from multiple bibliometric databases requires accurate cross-matching of records. A potential way to match between databases is through the use of persistent identifiers, such as DOIs. However, in some cases DOI information is not available, which makes other matching methodologies necessary. Here we report a first attempt to match records between the Web of Science (WoS) in-house database of the German Competence Center for Bibliometrics (KB) and Unpaywall. Such a matching procedure is necessitated by the large amount of records in WoS missing DOI information. Our method relies on a relatively basic set of features available in both databases, including article titles, journal names, journal ISSNs, publication years, and author counts. Using these features, we applied a matching algorithm to a subset of Unpaywall and WoS data. Initial results are promising, with a matching precision of ~99 % and a recall of ~95 % when compared to a 'gold standard' of articles matched by DOIs. Further improvement to the methodology may be possible through incorporation of further article and/or author features; however, scaling our process to be applied to the full WoS and Unpaywall databases remains a challenge. The R-Markdown file this report is based on as well as all queries are available in a Github repository.

Motivation

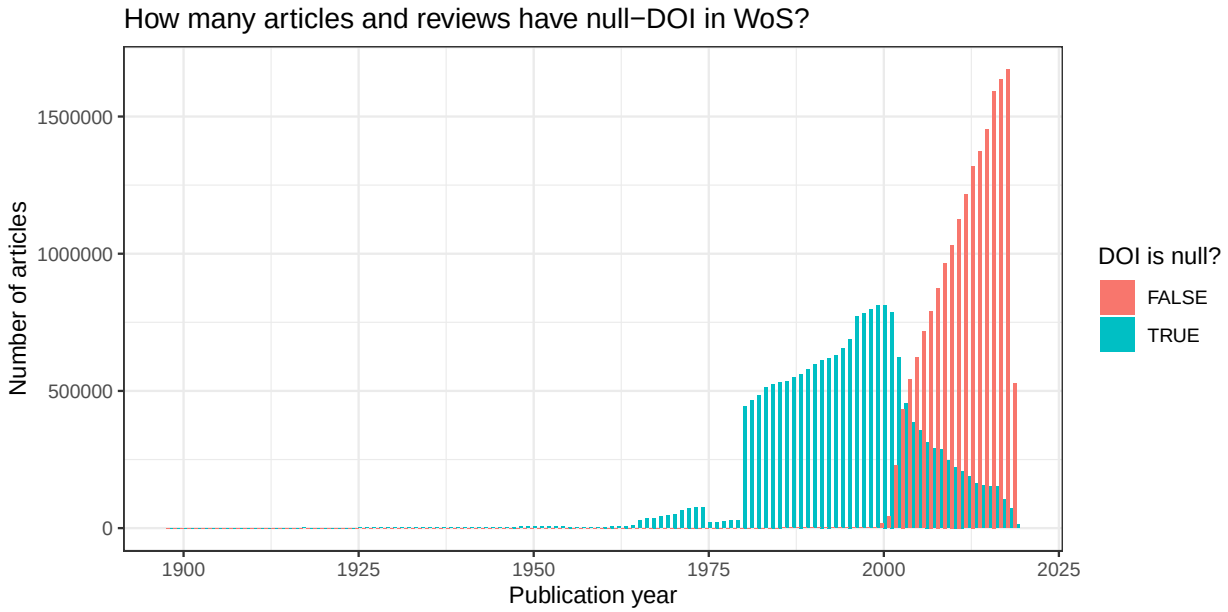
This report documents our approach to developing a procedure for the connection of items covered by the Web of Science (WoS) in-house database of the German Competence Center for Bibliometrics (KB), to information on their open access status contained in Unpaywall.

The most straight-forward way to perform matching between WoS and Unpaywall is based on digital object identifiers (DOI), i.e. persistent interoperable identifiers. However, DOI information in WoS is incomplete: not all records have a DOI stored in the database even though many of them are actually included in Unpaywall.



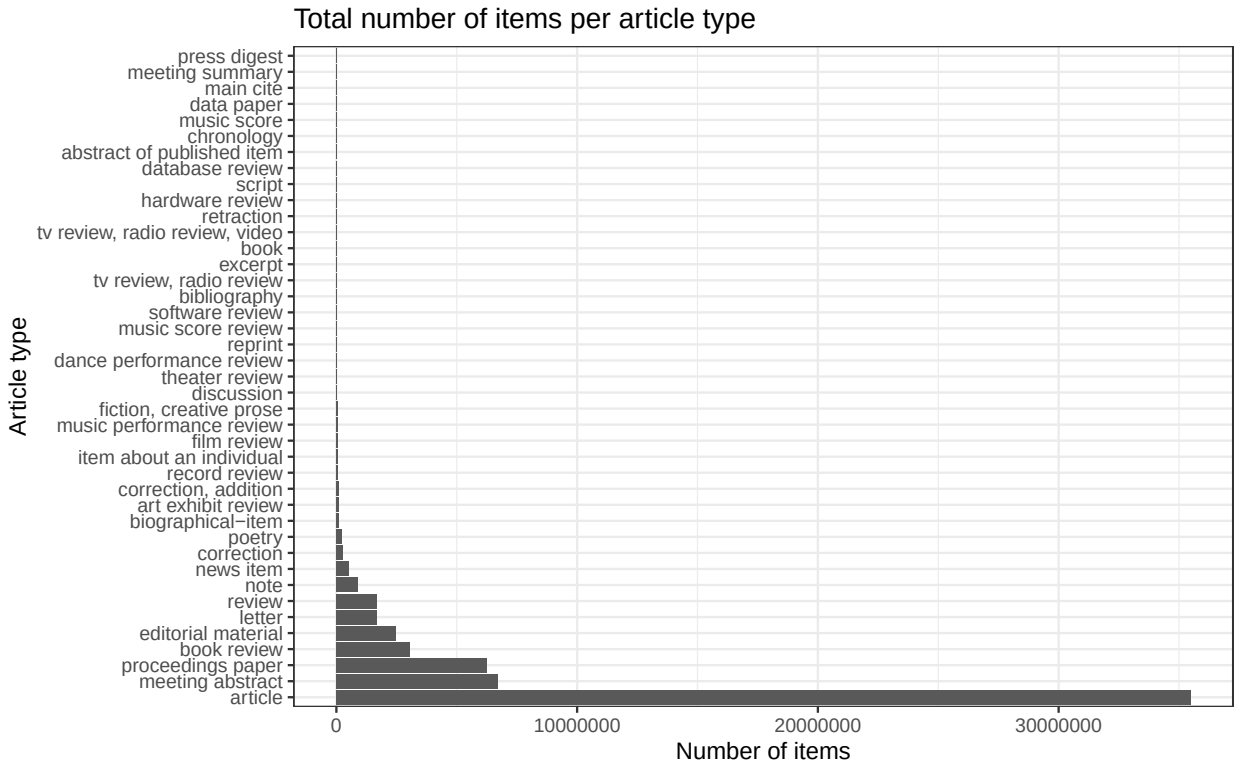
Looking at all 60,081,925 items in the 2019 version of the WoS-KB bibliometric database (`wos_b_2019`) we see that only 21,690,250 of them contain DOI information, corresponding to 36.1 %. The figure reveals that almost all of the items in WoS with a publication year before 2000 are missing any DOI information. Indeed, of 21,494,820 items older than 2000 in total, only 55,153 contain DOI information, corresponding to about 0.26 %. In more recent years, the number of items including DOI information has risen sharply. The coverage

is much better here, with 56.07 % of items published from 2000 onwards having DOI information. The decline in the number of items with and without DOI in the most recent 3 years is probably due to an indexing lag.

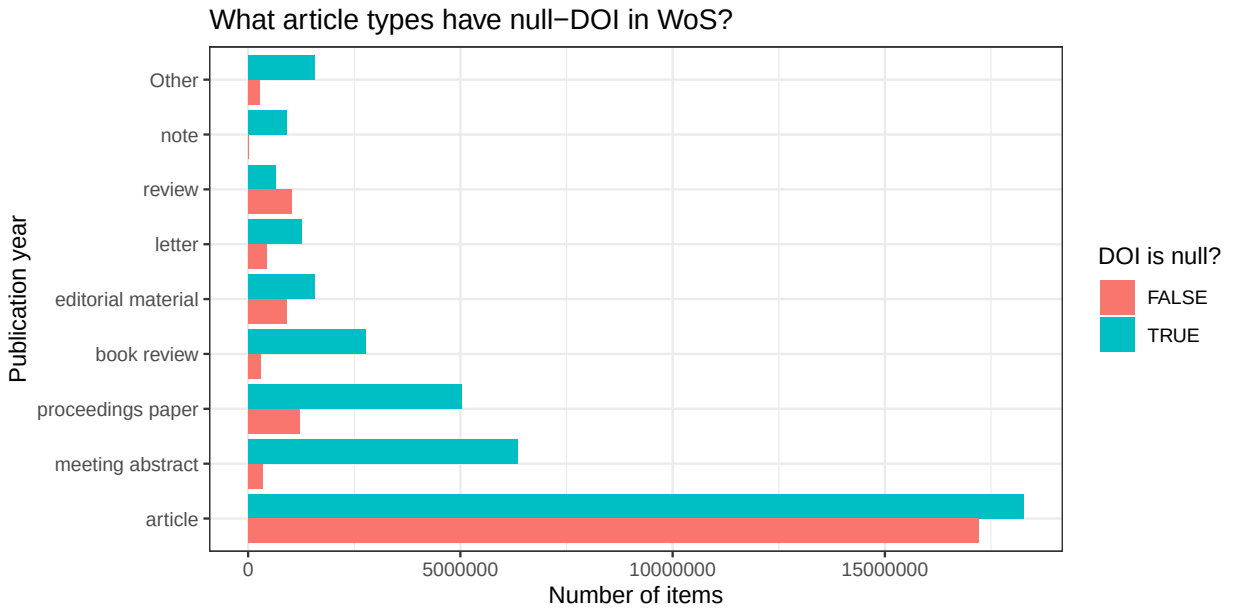


The observed trend is similar if we focus on the document types journal 'articles' and 'reviews' only. Overall, 49.04 % of all articles and reviews contain DOI information. Again, the percentage is extremely low for publication years before 2000 (0.27 %), and higher for recent years (75.24 %). Moreover, for these document types, we observe that the number of items missing DOI information decreases continuously from 2000 onwards, leading to a coverage of 94.03 % in 2017.

We now want to investigate more thoroughly how the proportion of items without DOI depends on the article type. In order to keep the figures readable, we first identify the most common article types and collate the remaining ones in the category `Other`. Looking at the number of items per article type in the following figures, we decide to keep all types with more than `100,000` items.



With this, we see the following behavior for the existence of a DOI per article type.



In this figure we see that missing DOIs are an issue particularly for proceedings papers, meeting abstracts, book reviews and marginal categories (`Other`). The proportion of journal articles and reviews without DOI is much lower, as the above results already indicated.

Data selection and acquistion

To develop and test our matching strategy we decided to focus on data from Unpaywall with publication year 2014 to reduce storage space needed (the whole Unpaywall data dump needs more than 100 GB). For the most recent years, indexing lags often cause problems and, as just mentioned, for earlier years the DOI coverage within WoS is much worse.

We further decided to focus exlusively on journal articles and reviews since these are the article types covered best in WoS and we consider them to be most relevant for the studies and scenarios where our matching strategy might be applied.

This means that from the Unpaywall data we only consider records with publication year 2014 that have the `genre` of `journal-article`.

We obtained our set of Unpaywall articles in a similar way to the method described in our blog post on open access evidence in Unpaywall. We will outline the main steps again below.

First, we downloaded the most recent Unpaywall data dump released in April 2019 from here and imported it into a local MongoDB database. The snapshot contains more than 100 million records and hence, the file is about 100 GB in size. To obtain a more manageable object to work with, we extracted the fields which are most relevant to our matching objective running extract_from_mongodb.txt on a bash shell.

We initially filtered for publication years between 2014 and 2016, extracted DOI, publication year, article type (`genre`), article title, and some author information, namely the number of authors. We also extracted the given and family names of the first and last author, but did not use it so far in the matching routine. We exported the resulting data as json file and loaded it as table into our GoogleBigQuery analytical environment (an access protected paid service), specifying a schema. There, we added information on the journal (ISSN and journal title) to the data from a previous export based on the local MongoDB database. We joined the journal related fields on DOI, which was possible since every Unpaywall entry contains DOI information. We then exported the table as three different .csv files, one for each publication year and imported the one corresponding to the publication year 2014 into the oracle environment of the KB.

Matching criteria

A number of features may be appropriate for matching articles between bibliographic databases. In the best case scenario, persistent identifiers such as DOIs can be used for direct matching. In the cases where persistent identifiers are missing, other characteristics of an article and its authorship may be used for matching, such as the article title, journal title or ISSNs. However, none of these features are 100 % unique (e.g. articles can share titles or journal names) and thus using them in isolation may lead to unreliable matches. A combination of these characteristics can produce more precise matching, i.e. records contained in two databases with a number of positive matching features have a high probability of referring to the same article.

In this report we focus on an initial subset of characteristics which may potentially be used for matching between WoS and Unpaywall, including publication years, article titles, journal names, ISSNs, and author counts. These criteria were selected primarily due to data availability and coverage in both databases, but are not exhaustive and may be supplemented with further characteristics in future (e.g. page numbers or first and last author names).

A description of the selected matching criteria follows. Note that for the following sections, items in WoS are limited only to ‘article’ and ‘review’ document types, and Unpaywall items to ‘journal-article’ types.

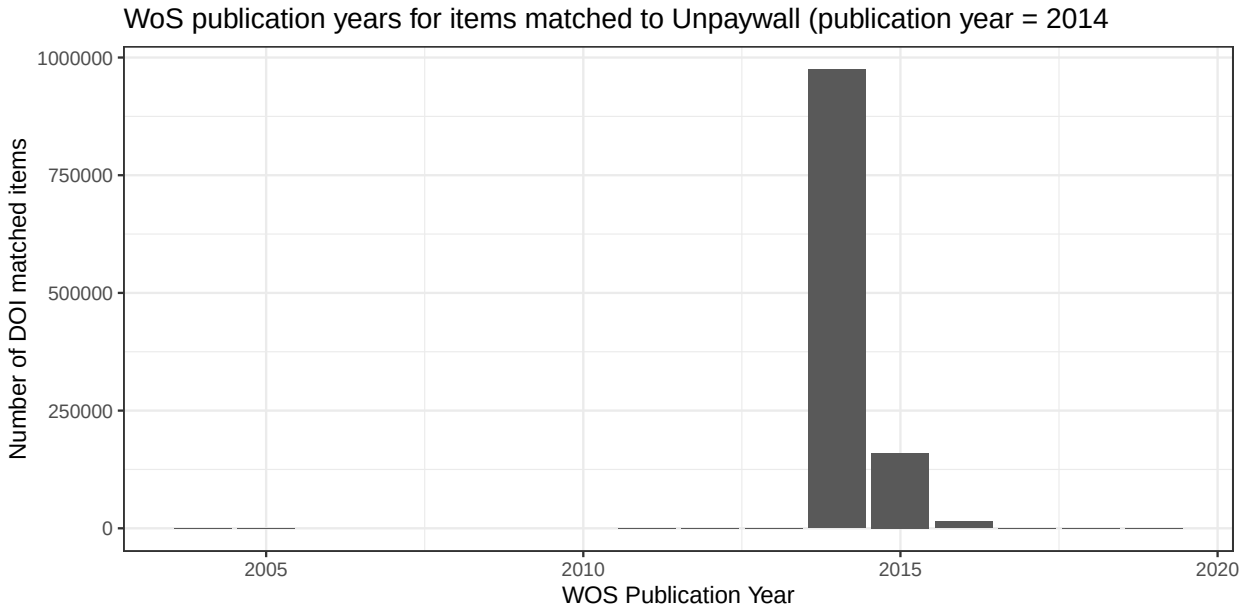
Publication years

An initial way to filter potentially matching records is via their registered publication years in the two datasets. However, the registered publication dates in WoS and Unpaywall sometimes differ (for example, because one takes the online publication date and the other the print publication date).

To understand how publication years vary between records contained in WoS and Unpaywall, we can consider the difference in publication years for all articles than can be directly matched with a DOI - in this way, we

can refer to the DOI-matched articles as a 'gold standard' against which matching criteria (to be used for matching items without DOIs in WoS) can be investigated.

Below we show the number of items that can be matched between our Unpaywall dataset (publication year = 2014) and all items in WoS, as a function of the WoS publication year:



The figure shows that most items have the same publication year, 2014, in WoS as in the Unpaywall dataset. However, a non-negligible proportion of items have publication years one or two years later in WoS than in Unpaywall. Based on this, we recommend to focus try to match items in Unpaywall with items from WoS having the same publication year, or one or two years later.

Article titles

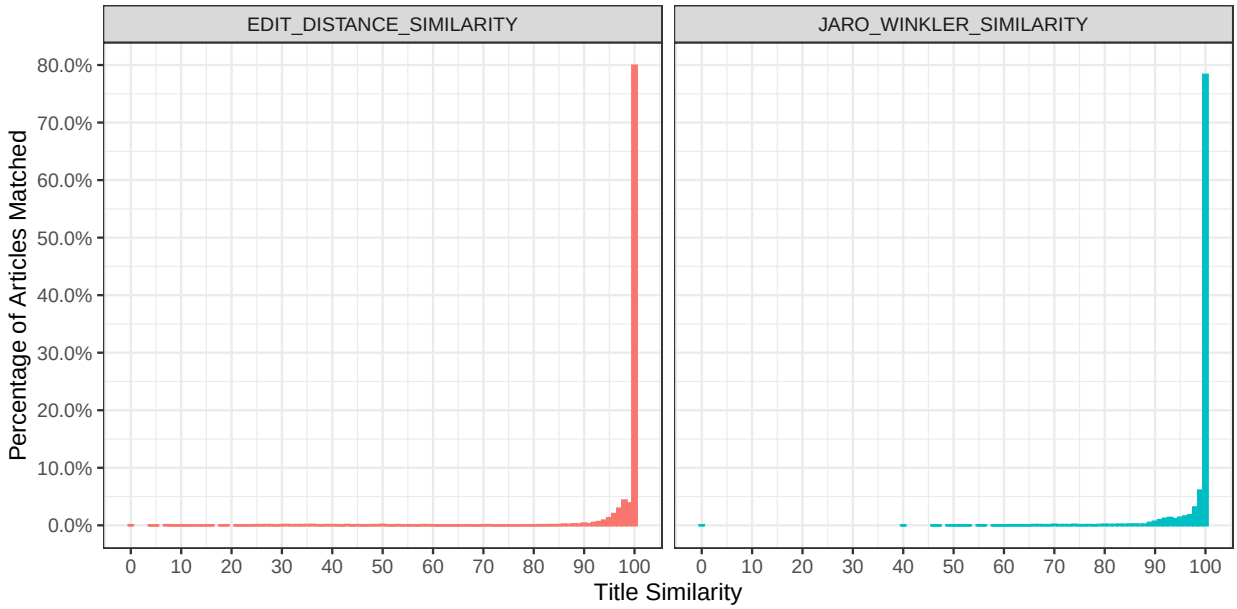
Article titles are potentially a good candidate for matching due to their high coverage: 100 % of 'article' and 'review' type documents in WoS, and 99.75 % of 'journal-article' type documents in Unpaywall have a title. However, when considering the proportion of unique article titles, 99.57 % of 'article' and 'review' titles in WoS are unique, compared to 95.19 of 'journal-articles' in Unpaywall, suggesting that relying on matching of titles alone would lead to a high number of false-positive matches.

As with the publication years, we used the set of DOI-matched articles as a 'gold standard', and investigated the percentage of articles that have exactly matching titles. We find that only 75.88 % of DOI-matched articles have exactly matching titles. Reasons for this may be due to differences in character encoding, handling of special character, or sentence cases (i.e. upper vs lower-case) of titles processed by WoS and Unpaywall.

An alternative to exact matching is to consider approximate string matching, which would allow titles to be matched even if they possessed minor differences in a small number of characters. Oracle provides two built-in functions as part of the `UTL_MATCH` package for approximate string matching: `EDIT_DISTANCE_SIMILARITY()` AND `JARO_WINKLER_SIMILARITY()`. Both functions provide a similarity score that is normalised such that 1 equates to perfect similarity and 0 to no similarity (or 100 % and 0 % when given as percentages, respectively). `EDIT_DISTANCE_SIMILARITY()` provides a similarity score based upon the Levenshtein distance between two strings, which counts the number of edits (insertions, deletions, or substitutions) needed to convert one string to the other. `JARO_WINKLER_SIMILARITY()` provides a similarity score based on the Jaro-Winkler distance, which measures the number of characters in common and number of transpositions.

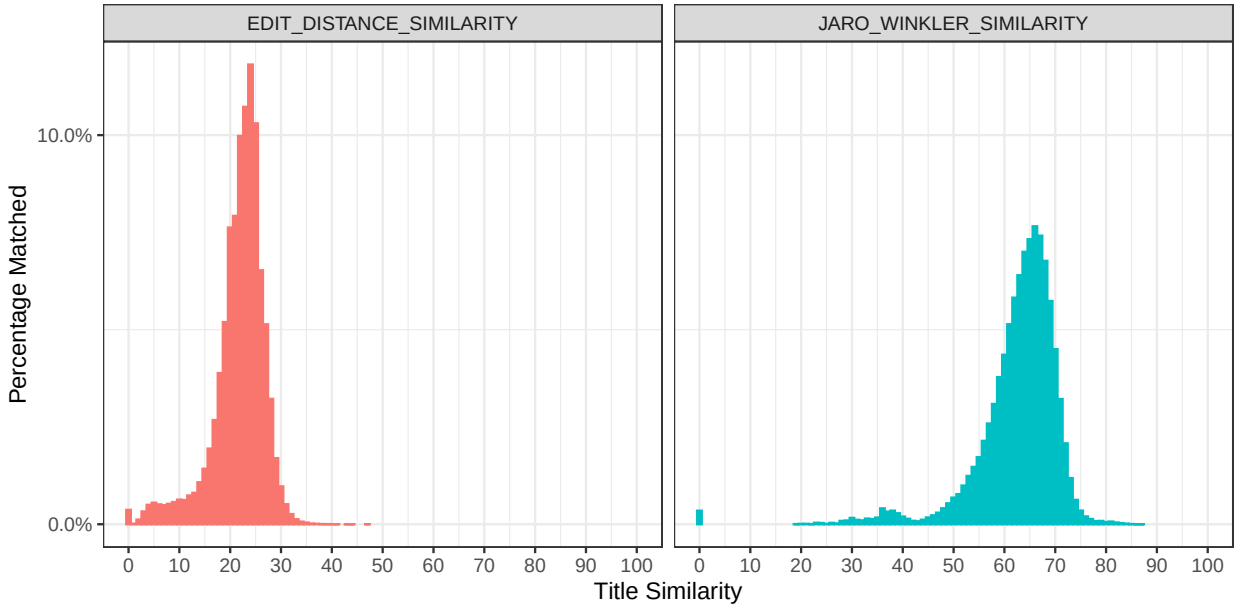
Below we plot a histogram of showing the proportion of matching titles in our DOI-matched set, as a function

of the two similarity measures. To further improve the matching efficiency, we transform the title strings to lower case (using the Oracle `LOWER` function), and remove whitespace from the start and end of the string (using the Oracle `TRIM` function). Note that for performance reasons, we use a small sample of the Unpaywall results set (~1%, using the Oracle `SAMPLE(1)` function) for this initial testing phase:



These plots show that `EDIT_DISTANCE_SIMILARITY()` and `JARO_WINKLER_SIMILARITY()` have similar distributions of similarity scores for titles on DOI-matched articles. Both distributions show that titles are consistently matched with similarities > 80 % (with a small percentage of outliers), suggesting this as a suitable threshold below which matched titles should be considered unreliable.

An additional method to assess the efficacy of the different similarity matching methods is to assess matching similarity on randomly matched articles, i.e. assuming that random titles should not be the same:



These plots show that the `EDIT_DISTANCE_SIMILARITY()` and `JARO_WINKLER_SIMILARITY()` produce much lower similarity scores when used to compare random titles, than when comparing titles of doi-matched

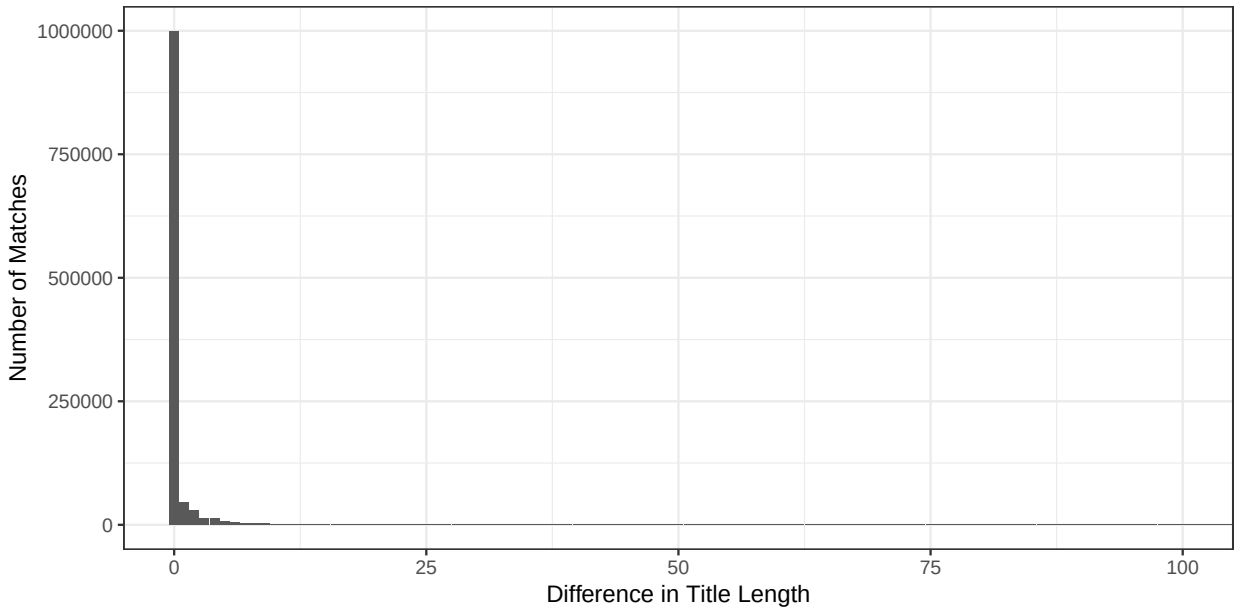
articles. These results suggest that matching via string similarity functions can be an effective method to increase the number of matches whilst retaining high precision. Notably, we can see different distribution patterns between the two similarity measures - it appears that `EDIT_DISTANCE_SIMILARITY()` produces much lower similarity scores (median = 23 %) with less overlap with the distribution of DOI-matched articles, compared to `JARO_WINKLER_SIMILARITY()` (median = 64 %).

We can thus interpret that `EDIT_DISTANCE_SIMILARITY()` is a more appropriate measure for matching titles. For matching between datasets we thus recommend to implement the `EDIT_DISTANCE_SIMILARITY()` function, with a similarity threshold of 80 %, for matching of article titles.

Article title length

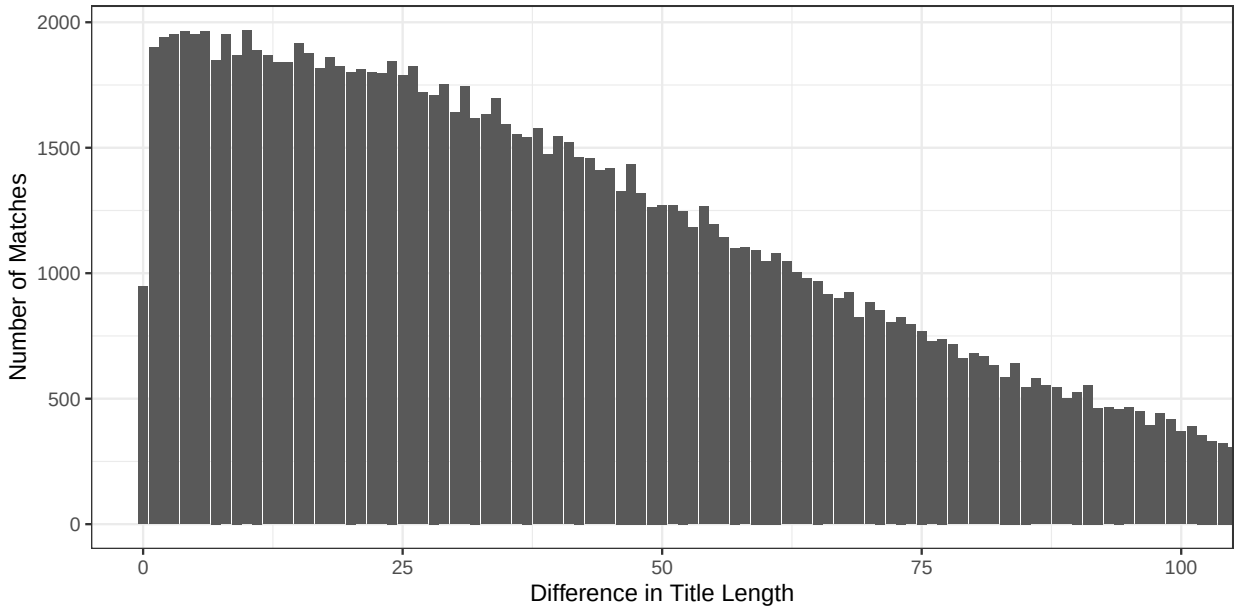
In the above section we matched titles using Oracle built-in functions for approximate string matching. However, such string similarity algorithms are notoriously slow on large datasets, as they require a matrix of rows to be checked against every other row in the dataset (e.g. if we were to compare 10,000 Unpaywall records with 10,000 WoS records, we would need to perform 100,000,000 string similarity comparisons). A target for scaling such similarity functions is to reduce the number of pairwise comparisons between titles. One potential way to do this is by first considering differences in the title length, where titles with greatly differing lengths are unlikely to belong to the same record.

First we compare the difference in title lengths of Unpaywall and WoS items in our DOI-matched dataset:



For our From this, we can see that the vast majority of titles do not differ significantly in length: 97.57 % of titles differ by less than 10 characters in length.

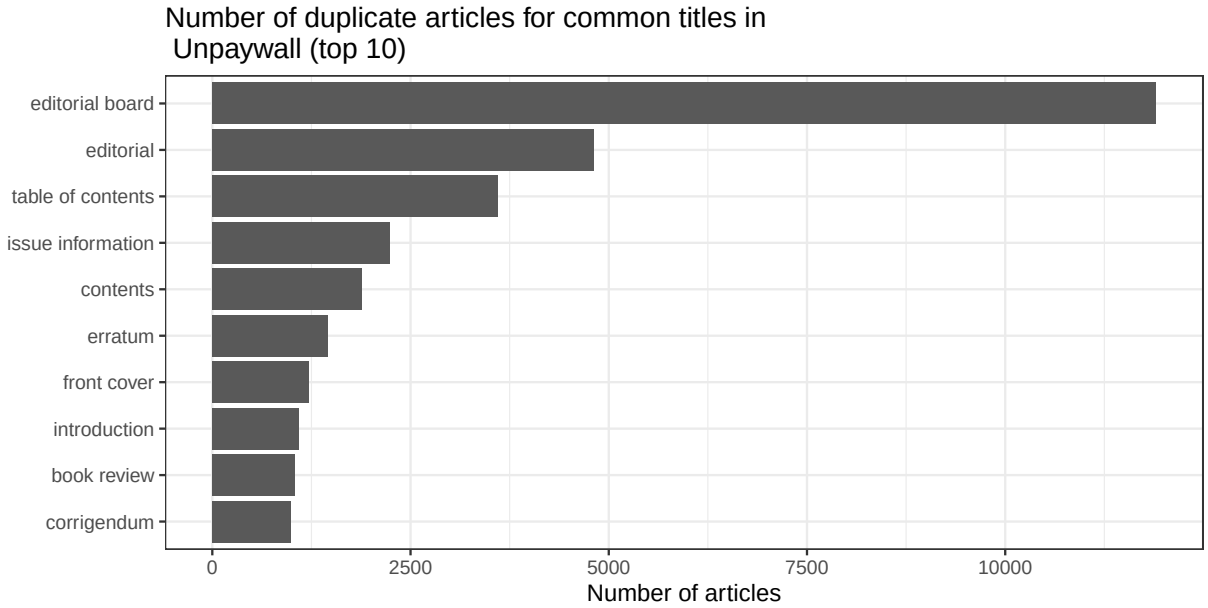
Another way to understand this is to compare the difference in length between titles of articles that are randomly matched, as, unlike our DOI-matched articles, we would expect randomly matched articles to have very different title lengths:

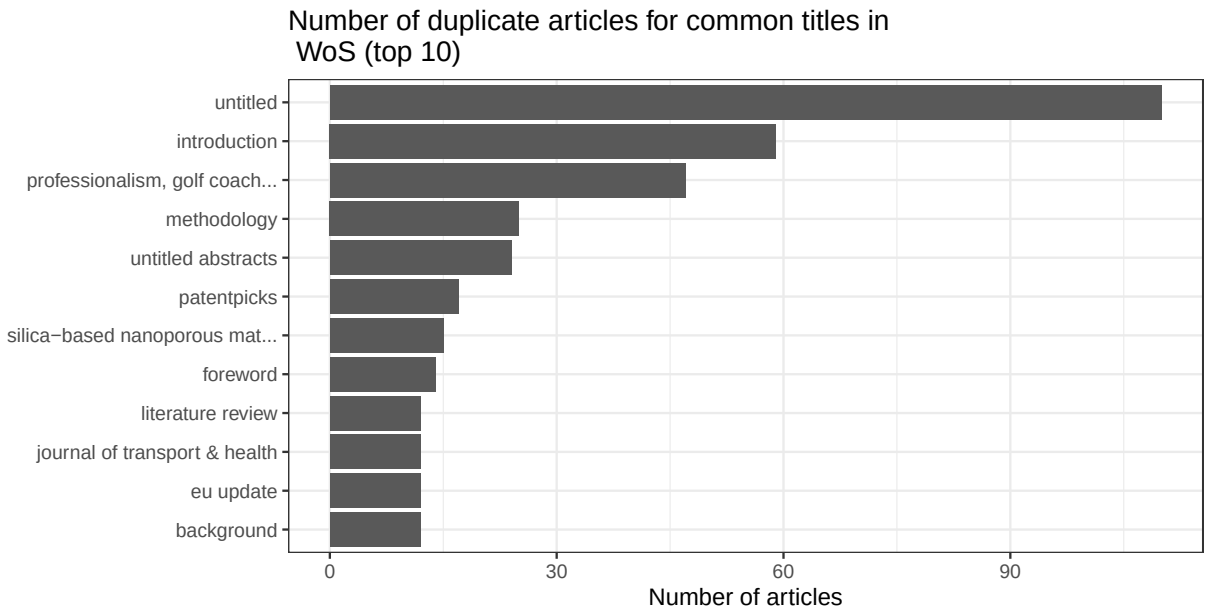


The results show that there exists a wide variability in the length of titles when randomly matched, thus, by only comparing titles with similar lengths, we could exclude a large number of pairwise similarity comparisons between article titles. For example, excluding articles with differences in title lengths over 10 characters, would exclude 86.61 % of the pairwise comparisons required, greatly improving performance.

Article title duplicates

A secondary factor related to titles is that of potential duplicates, i.e. two distinct records within the same database that share the same title. In some cases, title names can be shared by a large number of items (e.g. "editorial board"):





Due to these duplicate titles which cannot be definitively matched via title matching methods, our recommendation is to remove all items with duplicate titles from our datasets, and only match on records containing unique titles.

During these title matching processes, we also noticed a number of examples where items might match due to high similarity of titles, when they represent related (but not the same) documents. For example, an article may have a corrigendum associated with it, with the title "Corrigendum: [original article title]". Clearly, these two records do not represent the same document, but may match due to the similarity between titles, and matching of other associated metadata (e.g. the corrigendum may be in the same journal and have the same authors as the original document). We inspected cases where we found related articles with similar titles, and defined a manual blacklist of expressions which should not be contained at the start of a title.

Journal Titles

Journal titles present another possible facet for matching. First we checked the coverage of journal titles in datasets, finding that 100 % of items in WoS, and 100 % of items in Unpaywall are associated with a journal title.

We then checked the percentage of articles in our DOI-matched sample, that also possess the exact same journal title. We applied the Oracle functions `LOWER` and `TRIM` to the journal titles before matching, to improve efficiency. Despite a high level of coverage of titles in both databases, only 80.46 % of journal titles match exactly, for articles that can be matched together via DOIs. The reason for this is either minor changes in journal titles between datasets due to character encoding issues, or inconsistent reporting of journal titles between databases (e.g "PNAS" versus "Proceedings of the National Academy of Sciences").

ISSNs

A potential alternative to matching on journal titles is to match on the journal ISSN. As before, we first checked the coverage of ISSNs within each dataset, finding that 99.28 % of items in WoS, and 99.95 % of items in Unpaywall possess ISSN information.

We then calculated the percentage of DOI-matched items that have a matching ISSN. Note that some records in Unpaywall have multiple ISSNs associated with them, whilst WoS records are only associated with a single ISSN. We thus match dependent on whether ANY of the ISSNs associated with an item in Unpaywall are the same as the ISSN for an item in WoS. Note that ISSNs in Unpaywall are parsed to a separate table (documented in the following section on data preprocessing). We found that 97.43 % of DOI-matched articles

also have a matching ISSN, a higher rate than for journal titles alone. To provide the greatest coverage possible, we therefore recommend to conduct matching based on the ISSN OR the journal title.

Author counts

The final feature that we investigate here as a potential matching option is for author counts, i.e. matched articles should have the same number of authors. Author count data is provided within the KB database directly, whilst author counts for Unpaywall are calculated from the length of the "z_authors" field (see above section on "Data selection and acquistion").

First we checked the coverage of author counts in both datasets, finding that 100 % of WoS items, and 93.7 % of Unpaywall items had author counts.

We then checked the number of DOI-matched articles that also have exactly matching number of authors, finding that 98.96 % of articles that can be matched by DOIs also have matching numbers of authors. This high percentage means that we recommend to also use author counts as a matching feature.

Data preprocessing

For matching of records between WoS and Unpaywall, a necessary first step is to clean and normalise the data, both to improve matching efficiency and performance of SQL queries. For example, when matching on the DOI field, it is necessary that both DOIs are in the same letter case, and do not contain any superfluous characters (e.g. leading and trailing whitespaces), otherwise matches may be missed.

For both Unpaywall and WoS we applied the following cleaning and normalisation procedures:

- DOIs: Converted to lowercase (Oracle `LOWER()` function) and trimmed leading and trailing whitespace (Oracle `TRIM()` function)
- Article titles: Converted to lowercase and trimmed leading and trailing whitespace
- Journal titles: Converted to lowercase and trimmed leading and trailing whitespace

As mentioned before, we restricted the analysis to document types `journal-article` (Unpaywall) or `article` and `review` (WoS). From the imported Unpaywall table, and the WoS-KB database, we retrieved the fields we found to be potentially useful in the previous section, namely:

- DOI
- Publication year
- Article title
- Length of the article title
- Author count
- Journal name
- Journal ISSNs

As shown in the previous section on potential matching criteria, both datasets contain articles titles which occur in multiple rows. We decided to exclude all of these titles from our matching candidates since we want to think of an article's title as unique for this publication. We also exluded all titles starting with one of the keywords contained in this list, e.g. "correction", "erratum", or "addendum" since they would otherwise wrongly be matched with the original articles.

In order to speed up matching we also created an index for the DOI column in both tables.

Matching algorithm

The first step of our matching routine is simply using DOIs where they exist and are present in both databases. The more interesting part is the matching of articles without DOI information based on the criteria discussed in the previous section. As mentioned before, we only consider `journal article` document types from Unpaywall and `article` and `review` document types from WoS. We only try to match articles where the publication year from WoS is the same as or up to two years after the one in Unpaywall. Since all

Unpaywall articles we investigated in this analysis are from 2014, we perform the matching on WoS articles with publication year between 2014 and 2016.

Since the comparison of titles to determine their similarity measure is costly, we try to narrow down the subset of possible matching candidates as far as possible using the following criteria:

- pubyear: As mentioned, before we require the publication year in WoS to be the same as or up to two years after the Unpaywall publication year. This is already implemented in the preprocessing of the two matching tables.
- journal information: We restrict matching candidates to records with the same ISSN or (exactly) the same journal title. When multiple ISSNs are associated with an article (e.g. of the print and electronic version of the journal), we compare to all of them.
- author count: We compare only articles with coinciding number of authors.
- length of article title: We compare only articles where the lengths of the article titles differ no more than 10 characters.

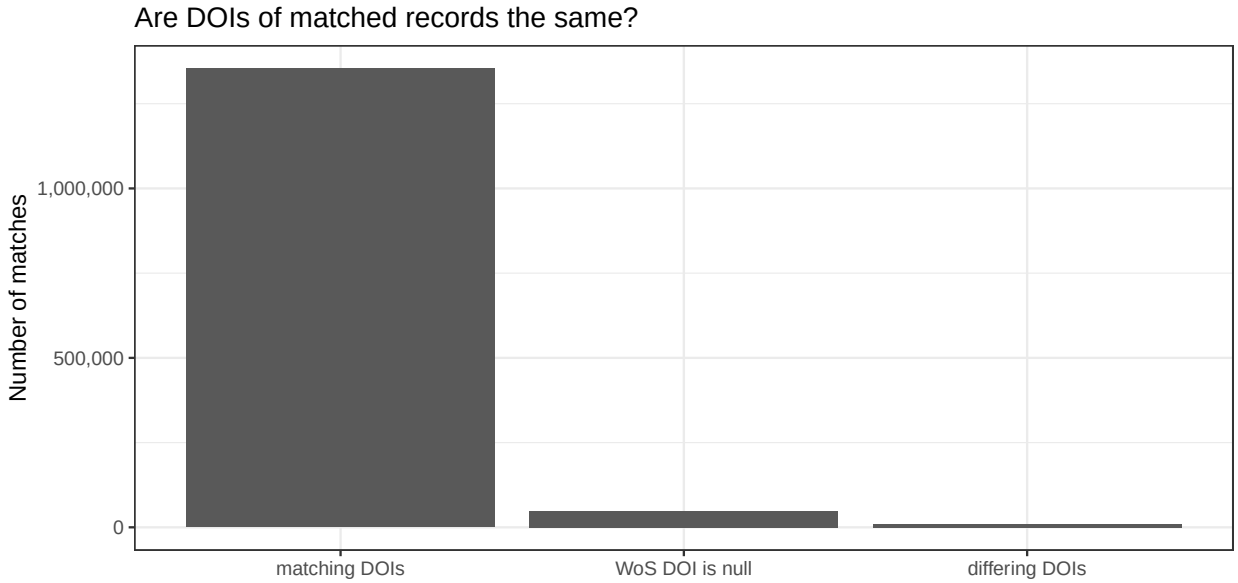
Then, we classify as matched articles the ones which have a title similarity (edit distance) of more than 80 percent.

This results in the query stored as create_upw14_wos_matching_results.sql to obtain a list of matchings.

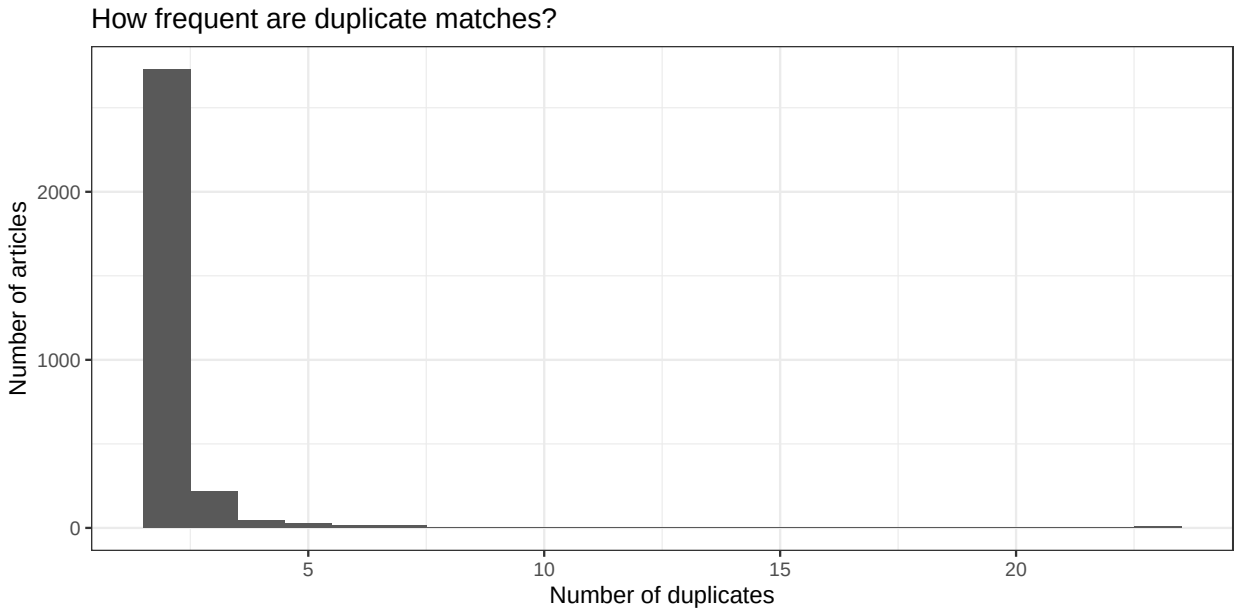
To evaluate the performance of our matching algorithm, we also run the matching algorithm on the two matching tables without using any DOI information (see the query create_upw_14_wos_matching_res_eval.sql).

Results

Using the above described algorithm, a total of 1,410,995 matched records were obtained. 1,353,911 were matched by DOI, corresponding to 95.95 %. Of the remaining 57,084 matches, for 47,307, DOI was missing in WoS and 9,777 had differing DOIs (i.e. the WoS DOI and the Unpaywall DOI were different for matched records). The following figure illustrates these numbers:



Some articles in Unpaywall were matched with multiple WoS records. A superficial check of these records revealed that for some of them, the same DOI was registered with differing items key and title in WoS. For others, the titles were almost identical with just one word or number changed (for example, “part 1” and “part 2”). They mostly appeared in the same journal with identical author count and were thus not recognized as different articles by the algorithm. The following figure shows the frequency of duplicates.



We additionally took a sample of 100 matched articles, which did not have any DOI information from WoS but were still identified with a corresponding article in Unpaywall. We manually checked all corresponding Unpaywall DOIs and found that all of them resolved to the correct article. It seems that at least for our sample the DOIs were just not registered with WoS.

Evaluation of matching routine

To evaluate the performance of our matching algorithm we considered the set of DOI-matched articles as the ground truth, that is we assume that records with matching DOIs are referring to the same article, and articles without a DOI match in the other dataset are only registered in one of the databases. Therefore, in the analysis, we will consider only articles that have DOI information also in WoS, since they could potentially be matched by DOI and exclude all articles with null DOI.

Comparing the number of articles that were matched by our algorithm in the second approach with the ones matched by DOI, we can calculate precision and recall. We use distinct counting here, since we assume multiple records with the same DOI to be rather a problem of duplicate database entries than of incorrect matching.

We find that using purely DOI, a total number of 1,353,856 articles could be matched. Our algorithm found 1,292,901 in total, of which 1,283,123 have coinciding DOIs. This corresponds to a recall of 94.78 % and a precision of 99.24 %. This means that even without using DOI information, almost 95 % percent of articles that could be matched by DOI can be matched correctly at a mismatch-rate of less than 1 %.

Discussion and Recommendations

In general, the results are quite promising. The algorithm is able to retrieve a large number of records even without using DOI information. For the more recent years with a better DOI coverage, it still finds additional matches for the entries with missing DOI. The amount of mismatches is quite low. However, there are some critical points that need to be considered and addressed in some way for an implementation of the algorithm for the purposes of the KB.

Firstly, the proposed algorithm can not distinguish well between articles with very similar titles published in the same journal and year with the same author count. For example, titles that only differ in a number (part 1, part 2, and so on) or a year or a single word are typically matched, leading to several uncorrect additional matches. Since the number of these occurances is quite low, we decided not to take further steps to exclude them. It might be possible to reduce the number of mismatches further by using more matching criteria,

like author names or to choose a very high threshold for the title similarity at the risk of loosing some other (correct) matches.

A second point to consider is that we decided to exclude all records that do not have unique titles in the preprocessing step. This might be too restrictive. Excluding only articles that coincide in the title and all other matching criteria will likely lead to an even higher number of matched articles. Alternatively, instead of excluding duplicate entries, one of them could be chosen.

In our algorithm, we require exactly matching author counts. For the Unpaywall dataset we caculated the author count ourselves from the nested field that lists all authors. For the WoS, however, we used the preimplemented author count from the items table without running any tests how reliable this number is (apart from comparing author counts with the ones from Unpwaywall for the DOI-matched records). Calculating the author counts from the available author information might yield better results.

Running the matching algorithm (without using DOI information) took little over 4 hours on the scriptserver. When using also DOI information, it is slightly faster, but also takes several hours to complete. For this report we only considered one year of Unpaywall records and three years of WoS records. Since the matching algorithm uses a restriction on certain publication years, we expect the runtime to scale more or less linearly if more publication years are included. Depending on the study one wants to conduct and the amount of data that should be included, this might not be feasible. Hence, for a single study, that focuses on more recent years, matching articles purely based on doi might be the best option.

To obtain more matches, especially for years before 2000, where the DOI coverage is unsatisfying, it might be advised to implement a more sophisticated algorithm. In our case the final matching criterion was a high title similarity. We use several other arguments to reduce the number of titles that have to be compared, since that is the most costly step in terms of run time. We found the following steps to be helpful:

- Publication year: Compare Unpaywall entries to WoS entries with the same publication year or up to two years later
- Journal: Compare articles with exactly the same journal title or matching ISSN
- Author count: We allowed only exactly matching author counts.
- Title length: We allowed for title lengths to differ up to 10 characters but not more.

For the integration of open access information into the KB infrastructure, a matching would most likely not have to be performed by each user individually, but could be done once after every upload of a new Unpaywall dump. Hence, an algorithm based on the one we developed for this report could be implemented to create a matching table, for example as part of the yearly released bibliometric database. Applicants can then base their studies and analyses on this table and do not have to perform a matching indivdually anymore.