Automatic detection of duplicate records in institutional repositories

Matteo Cancellieri matteo.cancellieri@open.ac.uk
Anton Zhuk anton.zhuk@open.ac.uk
Valerii Budko valerii.budko@open.ac.uk
Eka Chxaidze ekaterine.chkhaidze@open.ac.uk
Viktoriia Pavlenko viktoriia.pavlenko@open.ac.uk
Petr Knoth petr.knoth@open.ac.uk

The Open University

CORE and the OA landscape

CORE's mission is

to index all open access research worldwide and deliver unrestricted access for all.

We are here to support and advance the Open Access / Open Research movement

WE ARE

the world's **most used** collection of open access research papers from repositories

WE ARE

a **not-for-profit** scholarly infrastructure dedicated to the open access mission, **adopters of POSI** principles.

WE

provide solutions for content management, discovery and scalable machine access to research.

WE

serve the global network of repositories and journals by increasing discoverability and reuse of open access content.

Joining CORE benefits your institution

... and over 20 others are already our supporting and sustaining members

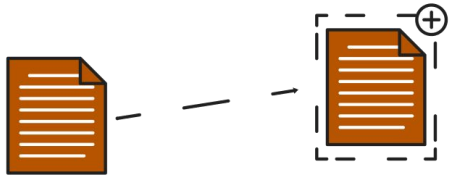
Become a member to demonstrate your support for Open Research

CORE

Duplicates in repositories

Is it a problem?

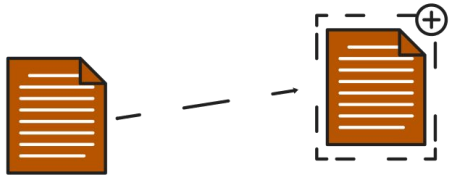
- Processing errors
- Multiple deposits by different authors
- Deposits by automated software



Duplicates in repositories

Is it a problem?

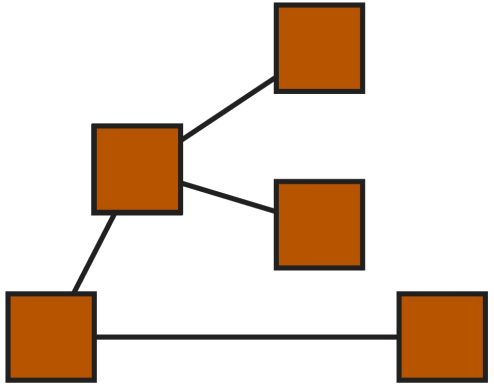
- Preprint vs. version of records
- Various deposit to satisfy various funders



Duplicates in repositories

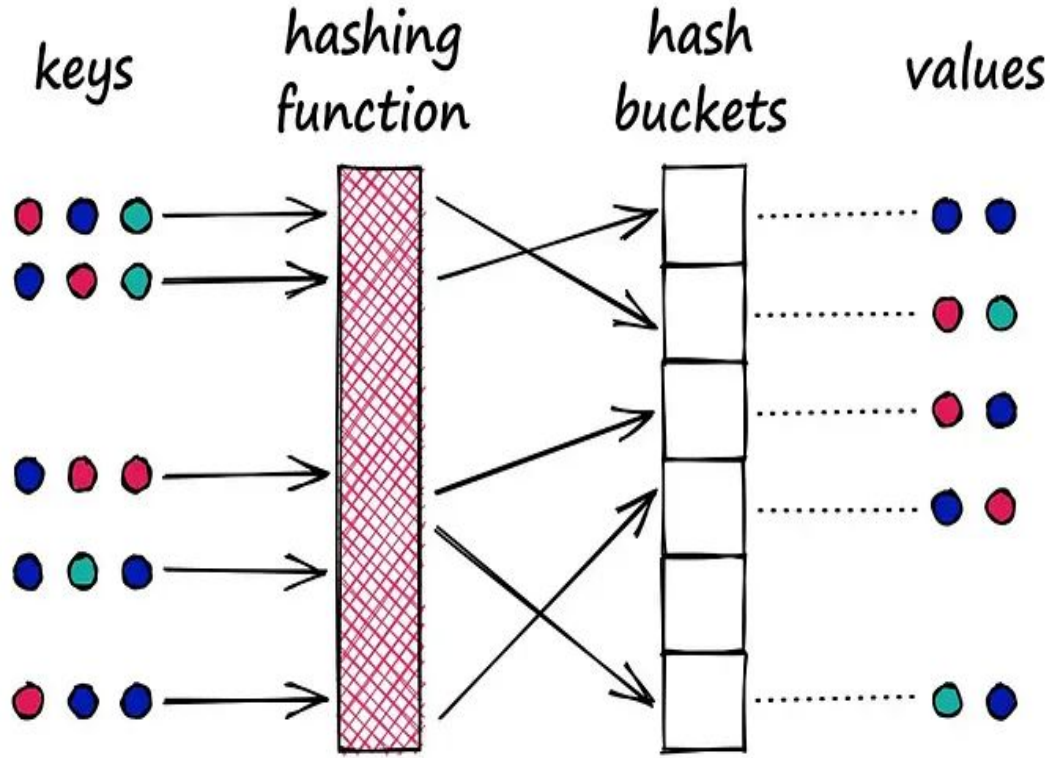
Is it a problem?

Scholarly graph degrades.



Locality sensitive hashing

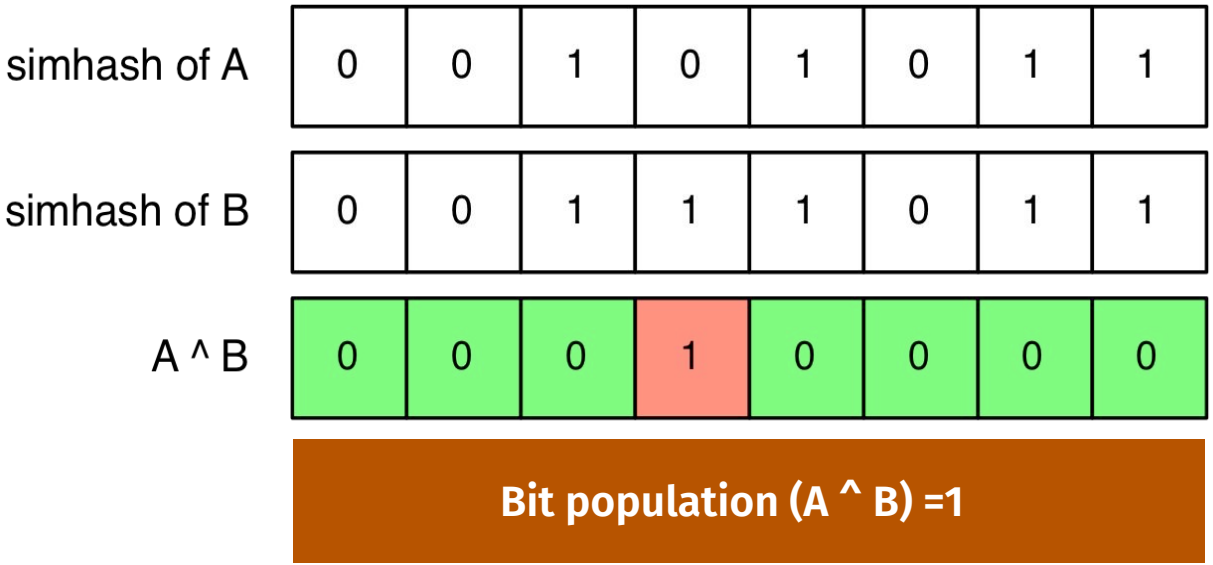
Simhash to be precise



CORE

Automated detection of near-duplicates

How can you do it at scale?



https://moz.com/devblog/near-duplicate-detection

Near duplicates in practice

Dataset from:

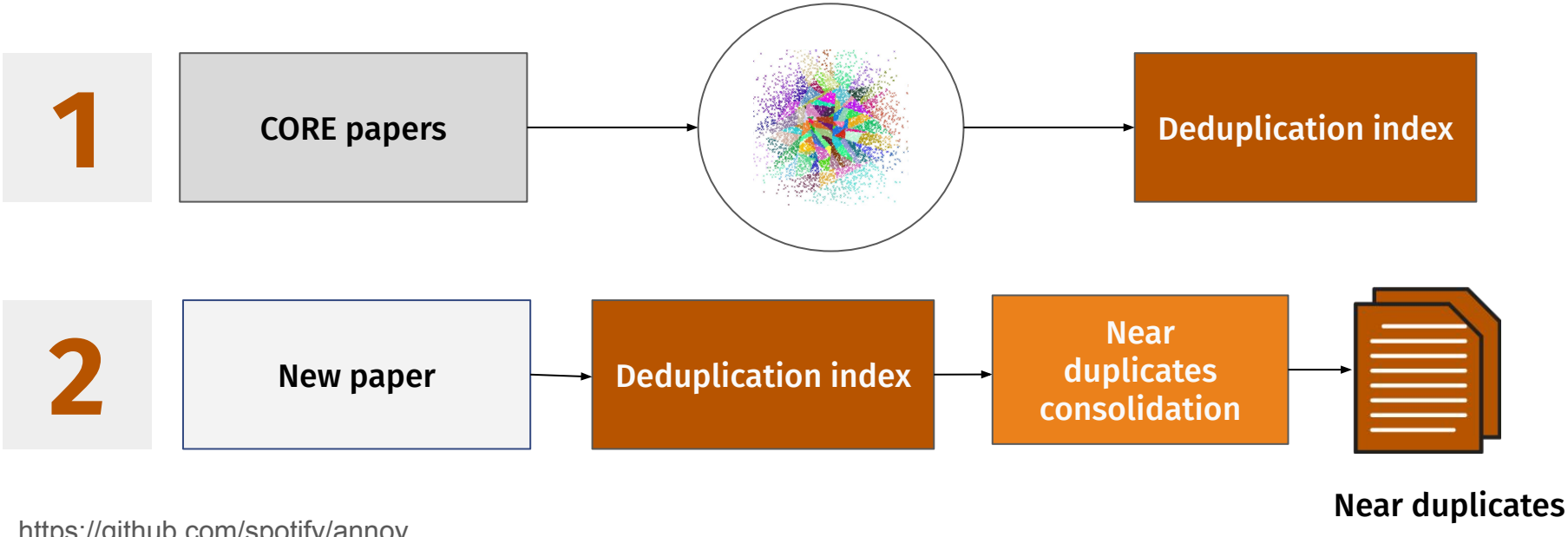
Deduplication of Scholarly Documents using Locality Sensitive Hashing and Word Embeddings

(Gyawali et al., LREC 2020)

CORE

Rules	Precision	Recall	Index duration	Search duration
Hamming distance on simhash 64 bit				
processed_title	0.83	0.77	48s	38s
processed_abstract	0.72	0.58	122s	112s
processed_title + processed_abstract	0.74	0.58	145s	131s
processed_title + processed_abstract 50	0.70	0.61	61s	53s
processed_title + processed_abstract 100	0.72	0.60	69s	60s
processed_title_initials	0.74	0.73	31s	23s
first_and_last_letters_processed_abstract + processed_title_initials	0.69	0.56	83s	75s
fullText	0.76	0.63	6201s	6162s
title+year	0.78	0.69	92s	87s
title + abstract + year	0.69	0.55	186s	178s
title + authorstring	0.58	0.55	144s	113s
title + first author	0.69	0.60	103s	87s
title + year + first author	0.68	0.57	102s	85s
Hamming distance on simhash 128 bit				
processed_title + processed_abstract	0.81	0.60	159s	154s
processed_title + processed_abstract_50	0.81	0.60	158s	153s
processed_title + processed_abstract_100	0.81	0.60	161s	147s

Large scale near duplicate checking in CORE

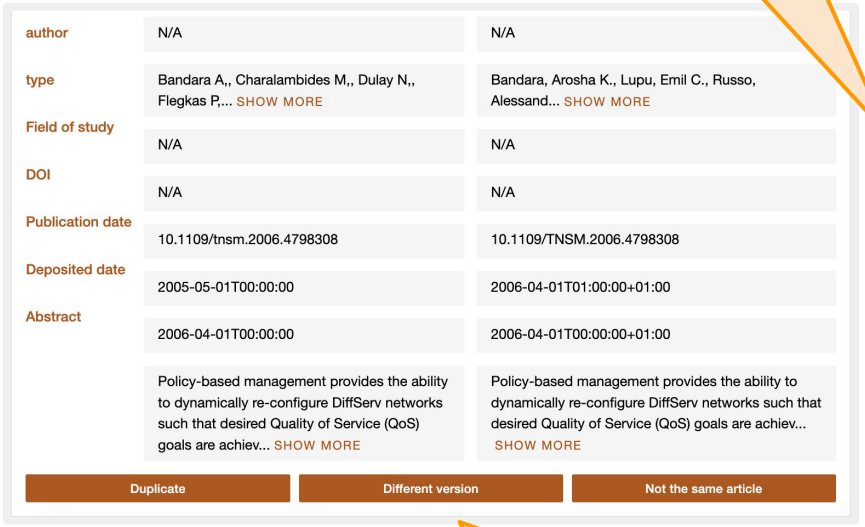


https://github.com/spotify/annoy

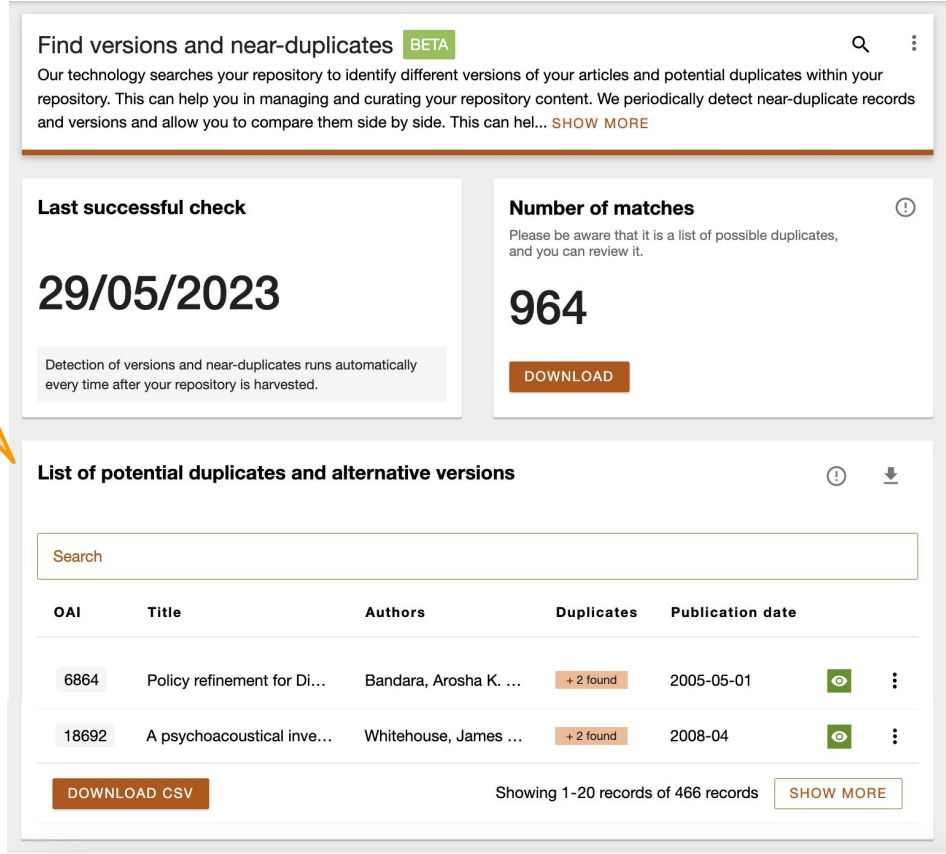
Finding version and near-duplicates

Versions/duplicates

List of possible duplicates

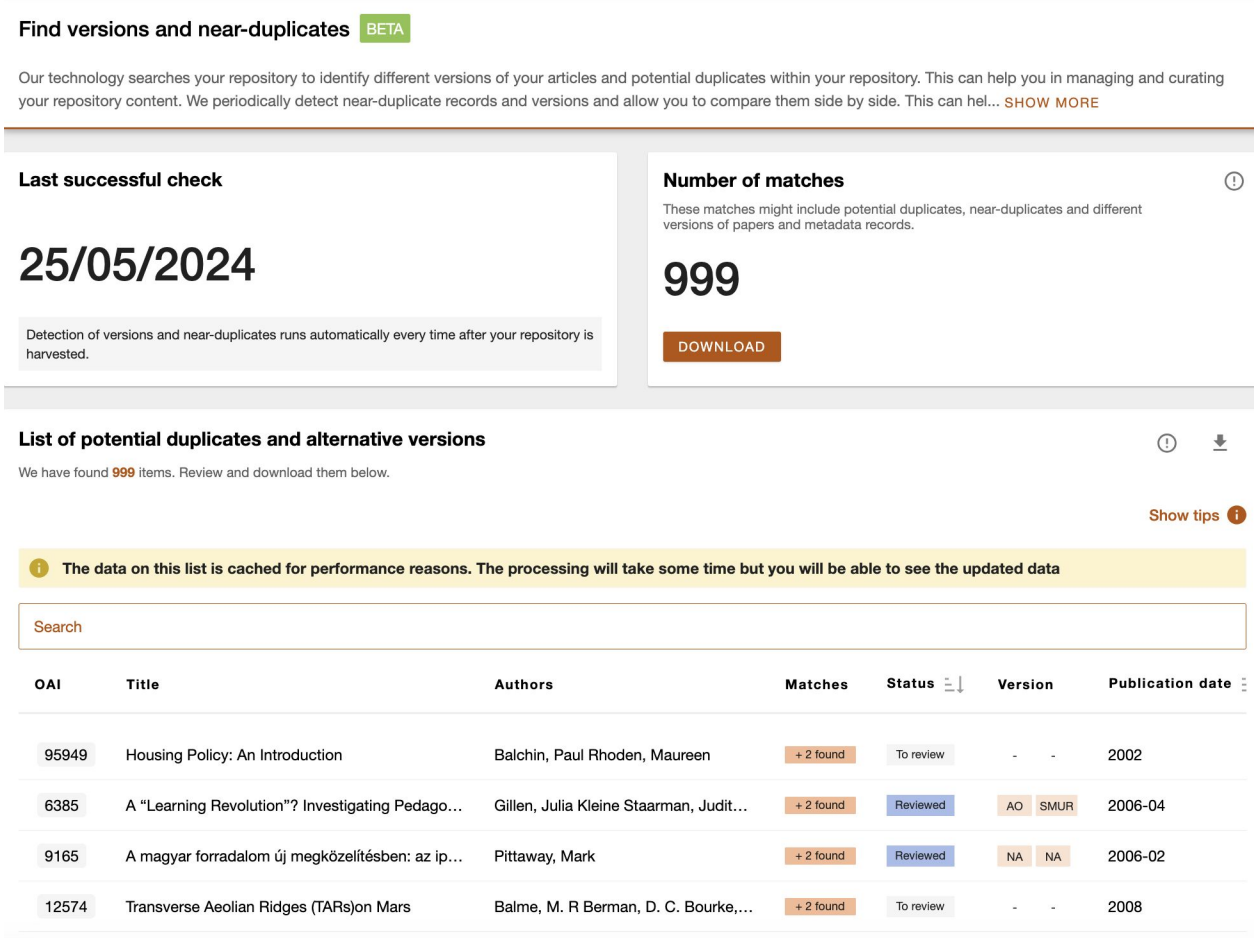


Comparison mode



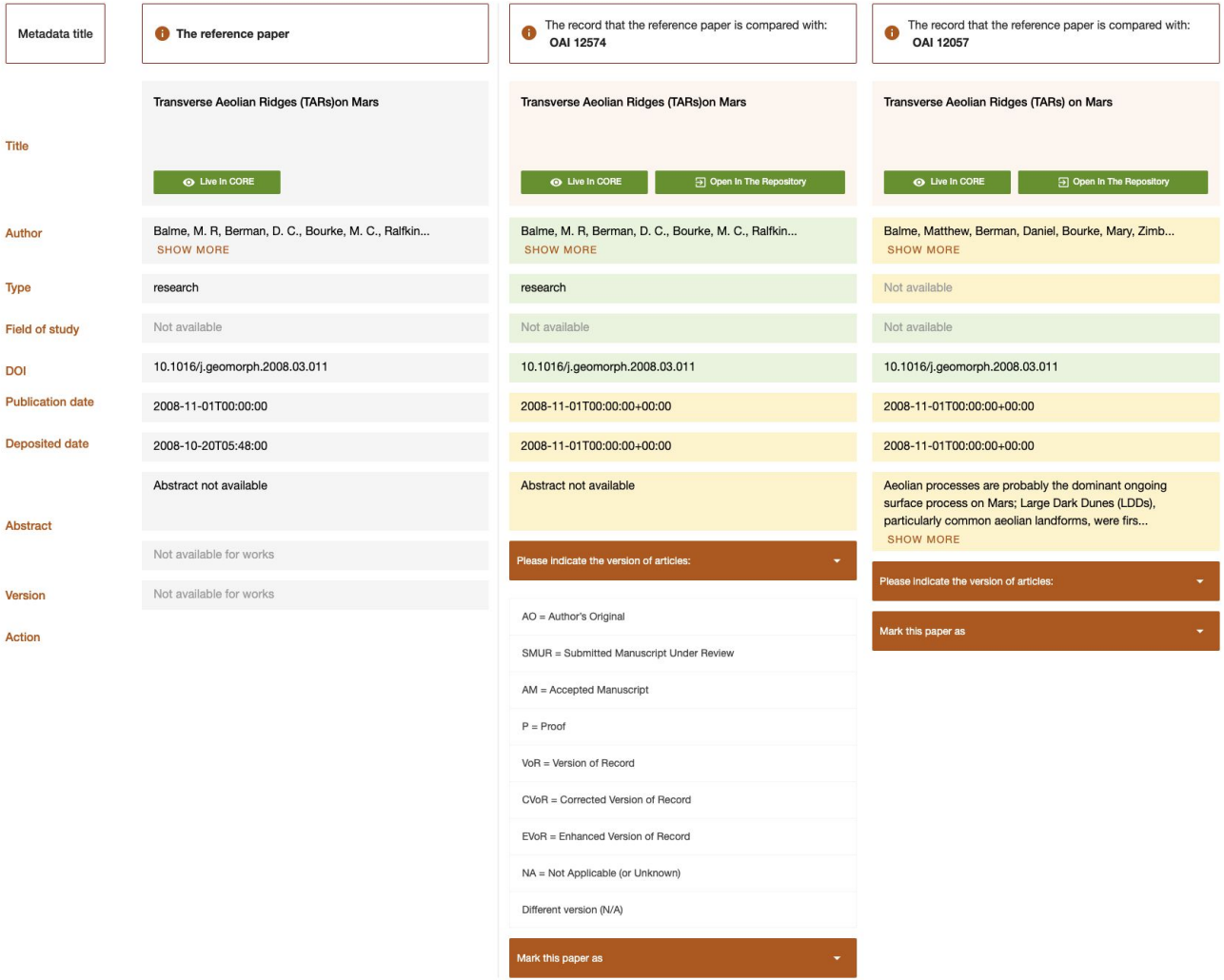
CORE

Finding version and near-duplicates



CORE

Finding version and near-duplicates



University of Chicago adopts CORE's article deduplication tool

We do not have a duplicate check built into our repository, so CORE's duplicate check is immensely helpful. While other repositories might already have duplicate checks, I don't know about their usage of ML to find duplicates. Based on my experience, I would guess that it isn't as accurate if it relies primarily on exact matches. CORE's duplicate check displays the "confidence" it has that a record is duplicated in the repository, which is also helpful when determining the final decision on whether a duplicate record...

Kirsten Vallee

Repository Services Manager at The University of Chicago

READ MORE

What's next?

- Integrate with repository software.
- Automatically find duplicates and "correct" them through your data
- Automatically predict versions of papers
- Expose the version information to the scholarly graph

Conclusions

We all benefit by having a deduplicated scholarly graph

CORE can help!

Integration with repository software is central in improving the experience

Thank you!

Matteo Cancellieri
matteo.cancellieri@open.ac.uk