

機関リポジトリ業務におけるITスキルの活用

芦北, 卓也
九州大学附属図書館

<https://hdl.handle.net/2324/7318920>

出版情報 : 2024-11-08. 九州大学
バージョン :
権利関係 : Creative Commons Attribution 4.0 International



機関リポジトリ業務における ITスキルの活用

九州大学附属図書館
eリソース課リポジトリ係 芦北卓也
<https://orcid.org/0000-0003-1853-7108>

きょうの内容

- はじめに
- JPCOARスキーマ対応(1)
メタデータ管理
- JPCOARスキーマ対応(2)
ハーベスト形式の移行
- リポジトリ登録インセンティブ対応
メタデータ一括登録
- おわりに

自己紹介

- 2001年 入職
- 2004～2007年 人事交流で福岡教育大学に勤務
- 2018年～現職

- 文学部卒 卒論はワープロで作成
- 卒業後はじめてパソコンを購入 Windows98?
 - Googleの普及期でなんでもGoogleで検索
- 最近はChatGPT等の生成AIが普及
 - AIが助言してくれる時代、まるでSF小説の世界



☐ 本文あり

コンテンツタイプ 部局

学内刊行物

紀要論文(31004)
博士論文(26277)
学術雑誌論文(9610)
その他(3440)
記事(2431)
会議発表論文(2230)
研究報告書(1046)
修士論文(936)
テクニカルレポート(891)
図書(部分)(517)
会議発表資料(444)
図書(238)
プレプリント(229)
データセット(193)

簡易検索

詳細検索

🔍 ✕

九州大学学術情報リポジトリ(QIR)

九州大学の知を世界に対して発信するオープンアクセスプラットフォームとして、九州大学の研究者によるさまざまな研究成果（学術雑誌論文、博士学位論文、研究紀要をはじめとする九大出版物など）を公開しています。

※一部のコンテンツは著作権等やプライバシーなどの理由により本文が非公開／学内限定公開になっています。

※九州大学学内刊行物のメタデータを公開、提供しています（抄録を除きCC0ライセンスを付与）。詳しくは「メタデータの提供」のページをご覧ください。

- オープンアクセスとは
- 九州大学学術情報リポジトリ運営指針
- 出版物一覧
- QIR統計
- 九大コレクション > 学位論文

<https://www.lib.kyushu-u.ac.jp/papers/>

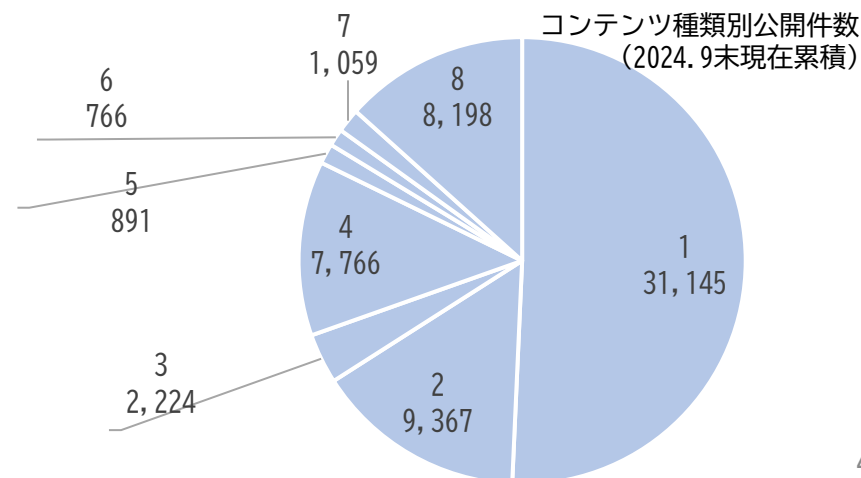
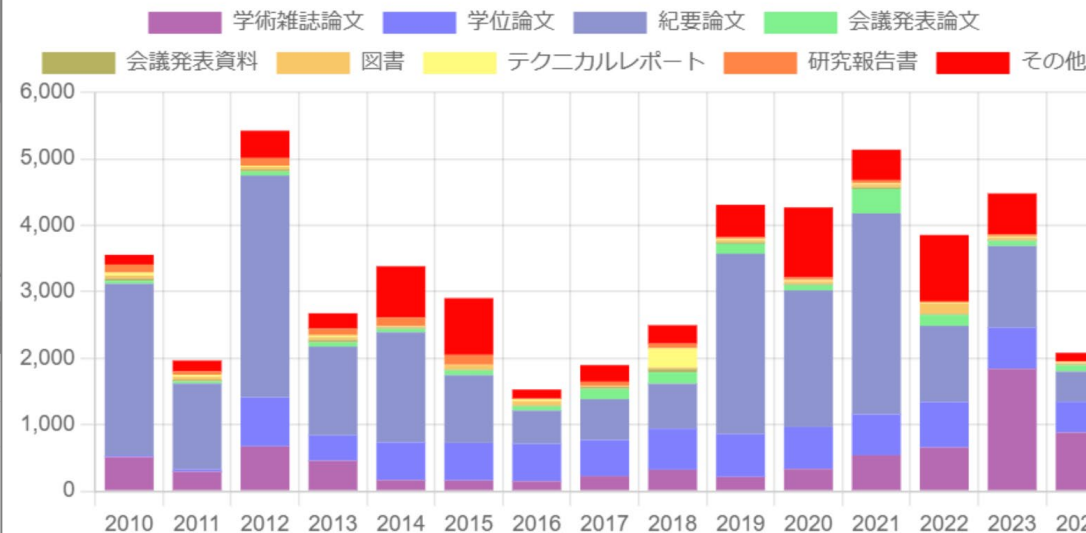
QIR統計

<https://www.lib.kyushu-u.ac.jp/ja/services/open/qir/statistics>

九州大学学術情報リポジトリ（QIR）は2006年4月14日に開設され、多くのコンテンツを公開しています。

累計コンテンツ件数：61,416 件（2024年9月30日現在）

新規公開コンテンツ件数の推移（年度別）

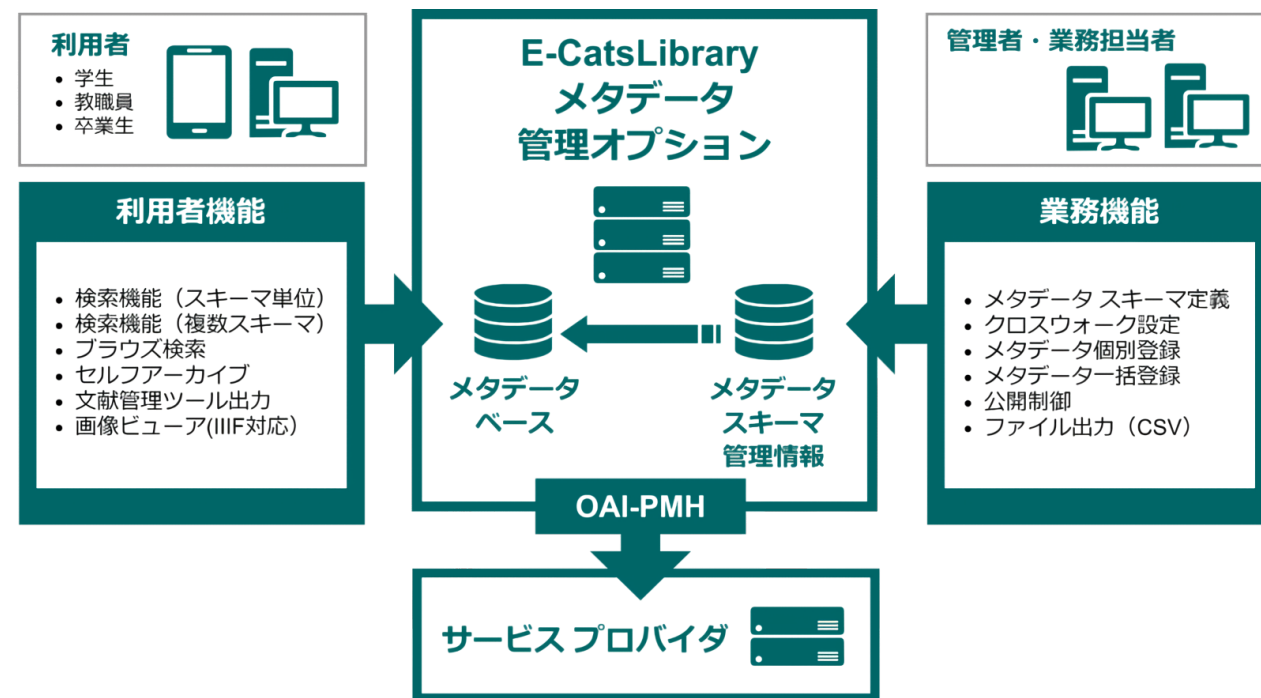


九州大学のリポジトリシステム

NECの図書館システム製品E-Cats Library v.7の メタデータ管理オプション

JAIRO Cloudは不使用
クラウド(SaaS)ではなくオンプレミス
安定運用中

- 目録業務や閲覧業務等、図書館サービス全般で同じE-Cats Libraryを使用
- 一部、**機関リポジトリ業務**や**eリソース系業務**、**デジタルアーカイブ系の業務**でメタデータ管理機能を使用



<https://www.cmsc.co.jp/metadata>

機関リポジトリのコンテンツを図書や雑誌、eジャーナルや
eブック、貴重資料画像等と統合検索可能(九大コレクション)

メタデータ管理画面

メタデータ管理詳細

検索画面 詳細画面

保存 フリー編集 終了

スキーマID papers メタデータID

管理情報

状態:登録中

公開レベル:表示する 公開日:2024.08.21

IIIF公開レベル:表示する

ハーベスト出力設定: ☐jpcoar ☒jpcoar_2.0 ☒junii2 ☒oai_dc

dc:title タイトル 1

xml:lang

dcterms:alternative その他のタイトル 1

xml:lang

jpcoar:creator 作成者 1 著者典拠

creatorType

jpcoar:nameIdentifier 作成者識別子 1

nameIdentifierScheme

nameIdentifierURI

jpcoar:creatorName 作成者姓名 1

xml:lang

nameType

jpcoar:familyName 作成者姓 1

xml:lang

jpcoar:givenName 作成者名 1

xml:lang

jpcoar:creatorAlternative 作成者別名 1

xml:lang

jpcoar:affiliation 作成者所属 1

JAIRO CloudにはDOIを用いたメタデータ補完機能があるが、E-Catsにはなく、入力作業が原始的

OA加速化事業によるシステム改修で改善予定(後述)

項目クリックで編集用のテキストボックスが開く

メタデータ管理詳細

検索画面 詳細画面

設定 終了

スキーマID papers メタデータID

メタデータ(フリー編集)

```
<dc:title xml:lang=""></dc:title>
<dcterms:alternative xml:lang=""></dcterms:alternative>
<jpcoar:creator creatorType="">
  <jpcoar:nameIdentifier nameIdentifierScheme="" nameIdentifierURI=""></jpcoar:nameIdentifier>
  <jpcoar:creatorName nameType="" xml:lang=""></jpcoar:creatorName>
  <jpcoar:familyName xml:lang=""></jpcoar:familyName>
  <jpcoar:givenName xml:lang=""></jpcoar:givenName>
  <jpcoar:creatorAlternative xml:lang=""></jpcoar:creatorAlternative>
  <jpcoar:affiliation>
    <jpcoar:nameIdentifier nameIdentifierScheme="" nameIdentifierURI=""></jpcoar:nameIdentifier>
    <jpcoar:affiliationName xml:lang=""></jpcoar:affiliationName>
  </jpcoar:affiliation>
</jpcoar:creator>
<jpcoar:contributor contributorType="">
  <jpcoar:nameIdentifier nameIdentifierScheme="" nameIdentifierURI=""></jpcoar:nameIdentifier>
  <jpcoar:contributorName nameType="" xml:lang=""></jpcoar:contributorName>
  <jpcoar:familyName xml:lang=""></jpcoar:familyName>
  <jpcoar:givenName xml:lang=""></jpcoar:givenName>
  <jpcoar:contributorAlternative xml:lang=""></jpcoar:contributorAlternative>
  <jpcoar:affiliation>
    <jpcoar:nameIdentifier nameIdentifierScheme="" nameIdentifierURI=""></jpcoar:nameIdentifier>
    <jpcoar:affiliationName xml:lang=""></jpcoar:affiliationName>
  </jpcoar:affiliation>
</jpcoar:contributor>
<dcterms:accessRights rdf:resource=""></dcterms:accessRights>
```

基本情報

状態 01 登録中

公開レベル 1 表示する

IIIF公開レベル 1 表示する

コメント

ハーベスト出力設定 ☐ jpcoar ☒ jpcoar_2.0 ☒ junii2 ☒ oai_dc

リソース種別

DOI登録フラグ

組織コード

リソースレコードID

リソース重複先レコードID

非公開理由

フリー編集機能によりメタデータをXML形式で編集することも可能

6

メタデータ管理詳細

検索画面 > 詳細画面

スキーマID papers メタデータID 7169360

「買える」図書館定番グッズを作る：九州大学附属図書館における商品開発記録 / 梶原, 瑠衣 国公立大学図書館協力委員会 2024-03-31

メタデータ カテゴリ(0件) 一次情報(1件)

編集 流用 取り下げ

文字サイズ:

公開

公開レベル:表示する 公開日:2024.03.05
IIIF公開レベル:表示する

投稿者: 梶原 瑠衣
山根 泰志

ハーベスト出力設定: ☐jpcoar_2.0 ☒junii2 ☒oai_dc

組織コード:附属図書館

登録者(複数可)とその所属(複数不可)

管理情報(部分)

dc:title タイトル 1
xml:lang ja
「買える」図書館定番グッズを作る：九州大学附属図書館における商品開発記録

dc:title タイトル 2
xml:lang en
Producing “Buyable” Standard Library Goods:Record of Product Development in Kyushu University Library

jpcoar:creator 作成者 1
creatorType
jpcoar:creatorName 作成者姓名 1
xml:lang ja
nameType
梶原, 瑠衣

jpcoar:creatorName 作成者姓名 2
xml:lang en
nameType
KAJIWARA, Rui

jpcoar:creatorName 作成者姓名 3
xml:lang ja-Kana
nameType
カジワラ, ルイ

jpcoar:affiliation 作成者所属 1
jpcoar:affiliationName 所属機関名 1
xml:lang ja
九州大学附属図書館図書館企画課企画係

jpcoar:affiliationName 所属機関名 2
xml:lang en
Planning and Public Relations Section, Library Planning Division, Kyushu University Library

メタデータの登録

メタデータ管理詳細

検索画面 > 詳細画面

スキーマID papers メタデータID 7169360

「買える」図書館定番グッズを作る：九州大学附属図書館における商品開発記録 / 梶原, 瑠衣 国公立大学図書館協力委員会 2024-03-31

メタデータ カテゴリ(0件) 一次情報(1件)

アップロード ※URLの場合、実ファイルはアップロードされません

対象 ☒ ファイル ☐ URL ☐ サムネイル

処理種別 ☒ 登録 ☐ 付替

ID ※指定された場合その上に登録します

ファイル ファイルを選択 選択されていません

公開レベル 1 表示する 公開日

ページ ~

説明

英語説明

備考

エンバーゴ対応として
公開予定日を設定可能

本文ファイル

ファイル

保存

表示順	ID	ファイル名	サイズ (kB)	登録日	公開レベル 公開日	説明 英語説明
	1139985	125_2166.pdf	2,937.20	2024.03.05	1 表示する 20240305	

作成者ID : drupal 最終更新者ID : cenmetaadm 登録日 : 2024.03.05 更新日 : 2024.04.04

一次情報の登録

JAIRO Cloudと同様にCSV(Excel)形式のフォーマットで一括登録も可能

リポジトリ業務でふだん使っているツール

- Microsoft Excel
 - ピボットテーブル、XLOOKUP、VBA
 - Microsoft Access
 - ODBC、集計、クロス集計
 - テキストエディタ、Mery等
 - テキストデータの整形、正規表現
 - 生成AI
 - ChatGPT、Copilot(大学提供)
 - GakuNin RDM(NII提供)
 - データ解析機能、Python
 - Adobe Acrobat
 - PDFの加工、OCR処理
- ※今回の講演では説明の対象外

- はじめに
- JPCOARスキーマ対応(1)
メタデータ管理
- JPCOARスキーマ対応(2)
ハーベスト形式の移行
- リポジトリ登録インセンティブ対応
メタデータ一括登録
- おわりに

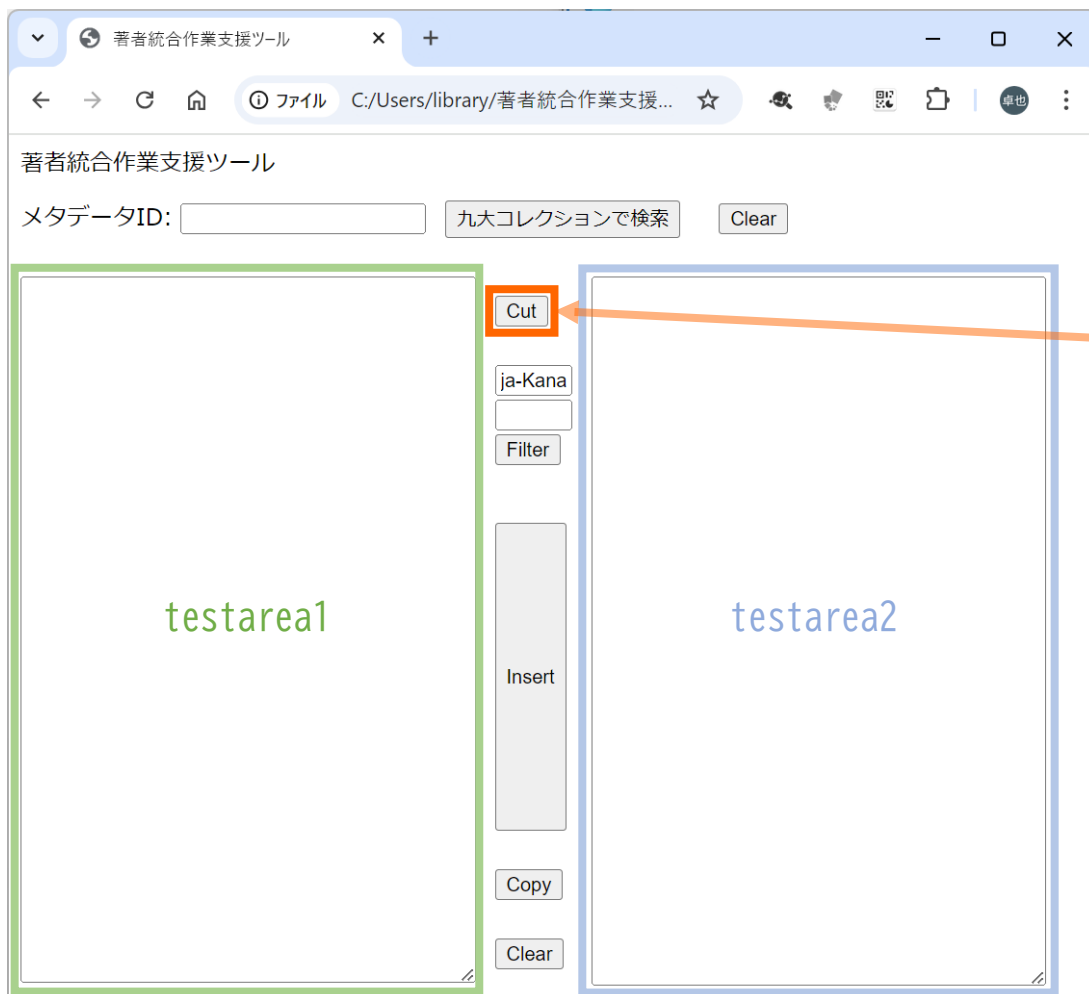
著者情報メタデータのメンテナンス

- 2017年 JPCOARスキーマに移行
→ フラットだったメタデータ構成が階層化 → 複雑化
- システム更新に伴うデータ移行で著者情報の一部に不具合発生



- アルバイト作業による修正を続けていたところ、Googleから不正データとみなされクローリング停止 → Google検索不可 → **ダウンロード数急減**
- 年内の早期解決をめざし課内総動員で分担作業
→ Excelでの一括修正を開始 → 781件はどうしても地道な手作業が必要
→ **JavaScriptでツールを作成し手作業の負担を軽減**

JavaScriptで作成したツール



1. 対象レコードのメタデータを左の枠内にコピー
2. 統合元(ja-Kana)の作成者タグ全体(<jpcoar:creator>~</jpcoar:creator>)を選択(複数可)して「Cut」ボタンをクリック
3. 「Filter」ボタンをクリック
4. 左の枠内にある、統合先作成者タグ内のja-Kanaを挿入する行内にカーソルを置き、「Insert」ボタンをクリック
5. 「Copy」ボタンをクリック

HTMLファイル中にスクリプトを書いてWebブラウザで実行
入力フォームや実行ボタンは通常のHTMLで記述

```
<div style="float:left">
  <textarea name="textarea1"></textarea>
</div>
<div style="float:left">
  <input type="button" value="Cut" onclick="clickBtn1()"><br />
</div>
<div>
  <textarea name="textarea2"></textarea>
</div>
```

関数の呼び出し

CutボタンをクリックするとclickBtn1()関数が実行

大きさや位置の定義は省略

スクリプトは<script>タグ内に記述

```
<script>
function clickBtn1() {
  var intStart = document.form1.textarea1.selectionStart;
  var intEnd = document.form1.textarea1.selectionEnd;
  var intLen = document.form1.textarea1.value.length;
  //左テキストエリアの選択範囲を右テキストエリアにコピー
  document.form1.textarea2.value += document.form1.textarea1.value.slice(intStart, intEnd);
  //左テキストエリアの選択範囲を削除
  document.form1.textarea1.value = document.form1.textarea1.value.slice(0, intStart) +
    document.form1.textarea1.value.slice(intEnd, intLen);
  //左テキストエリアの空行を削除
  var str = document.form1.textarea1.value.replace(/^\\n/gm, '');
  document.form1.textarea1.value = str;
}
</script>
```

関数の定義

左テキストエリア(textarea1)の選択範囲を
右テキストエリア(textarea2)にコピーするスクリプト

メタデータ修正作業時のSlack

 **Ashikita Takuya** 2020年12月10日 18:39
手作業リストの修正済みが100件に達しました。残り682件(87%)です。

 **Ashikita Takuya** 2020年12月11日 19:11
手作業リストの修正済みが200件に達しました。残り582件(75%)です。

 **Ashikita Takuya** 2020年12月11日 19:18
試しにツールを使わず5件ほど作業してみたところ、1件当たり5〜6分かかることが分かりました。相当しんどい作業で、これまでアルバイトさんたちに非人間的な仕事を強いていたことを改めて実感しました。

 1 

 **Fnday Ai** 2020年12月11日 19:47
以前試しに一日中作業したら、その後めまいを発症して2日間起き上がれませんでした。

 1 

 **Omura Takashi** 2020年12月15日 14:45
私の方で作業中だった1294件、完了しました。
Excelでやるのも結構目が疲れます。最近まぶたが痙攣するようになってしまったので、今後はのんびり休みます。

 2 

 **Ashikita Takuya** 2020年12月15日 19:18
室井さん、遠藤さんに協力していただいて、手作業分が782件中、350件すみしました。残り432件(55%)で、半分まであと少しです。（編集済み）

 1 

 **Hachi** 2020年12月15日 19:43
みなさん、無理なくやってくださいね～。目は大事！

作業時間短縮
概算で72時間→20時間

ツールの使用により
身体上の負担も軽減

 **Ashikita Takuya** 2020年12月17日 18:18
残り310件で、あと40%というところまで到達しました。年内解決を目指します。

 1  1 

 **Ashikita Takuya** 2020年12月22日 19:18
82%消化して、残り140件です。

 2 

 **Omura Takashi** 2020年12月23日 08:27
じつは、Googleへの完了メールはすでに書き終わっていてすでに出せる状態です。もう出しますかね？

 **Ashikita Takuya** 2020年12月23日 08:33
残りわずかなので、どうせなら完遂してから気持ちよく出したいです。

 1 

 **Ashikita Takuya** 2020年12月23日 17:10
今日はHachiさんに手伝っていただきました。残り50件、あと6.4%です。

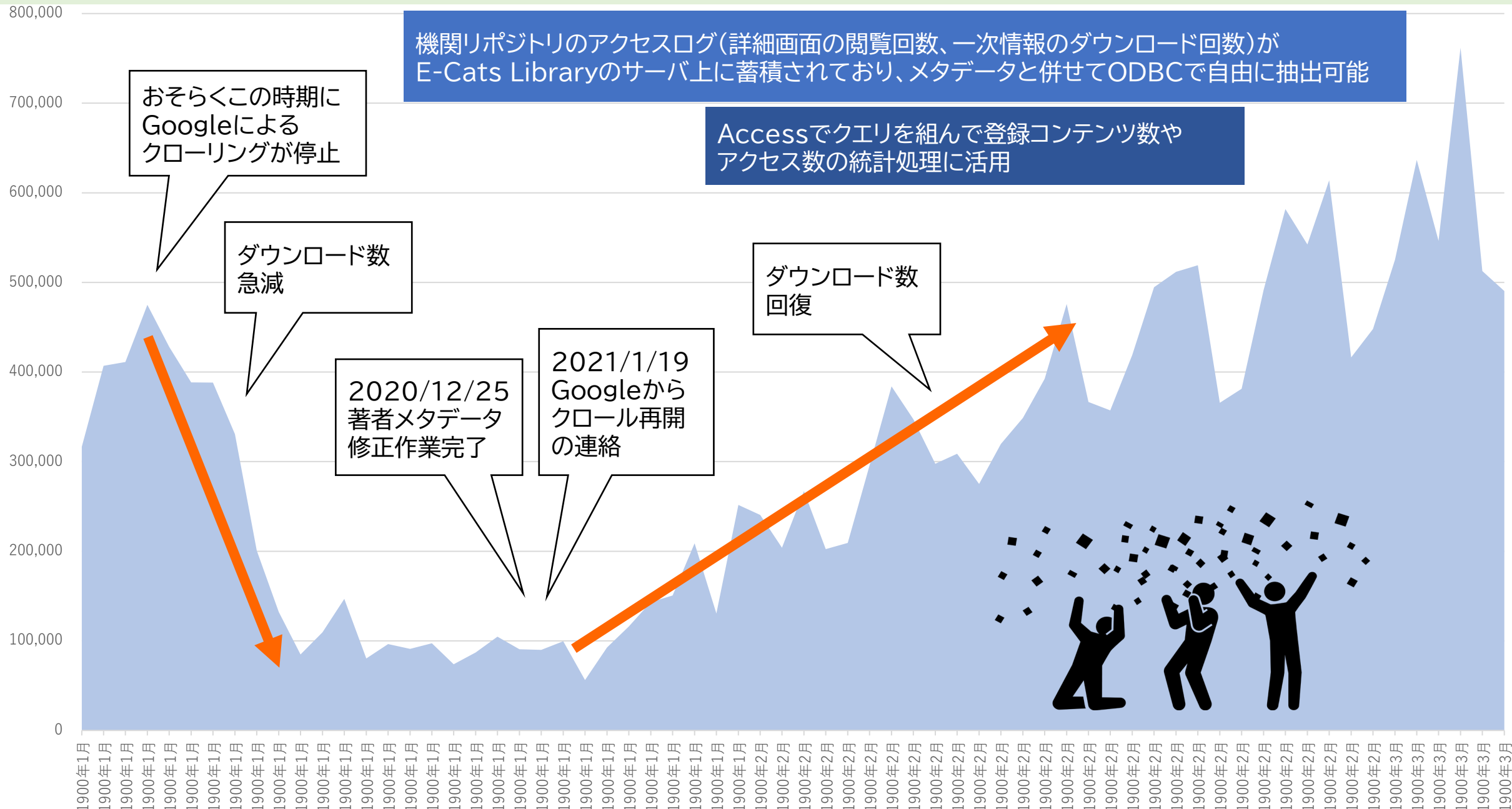
 1 

 **Ashikita Takuya** 2020年12月25日 09:51
Hachiさんから全件完了のお知らせがありました。皆さんご協力ありがとうございました。

 2  1  1 

年内早期解決の
目標を達成

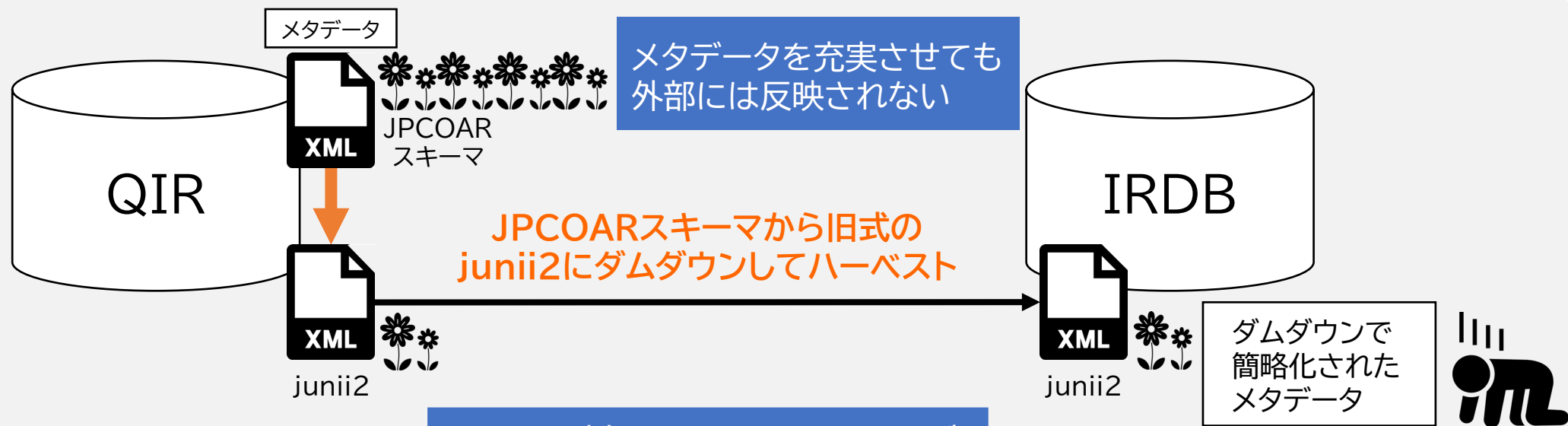
QIRダウンロード数の推移



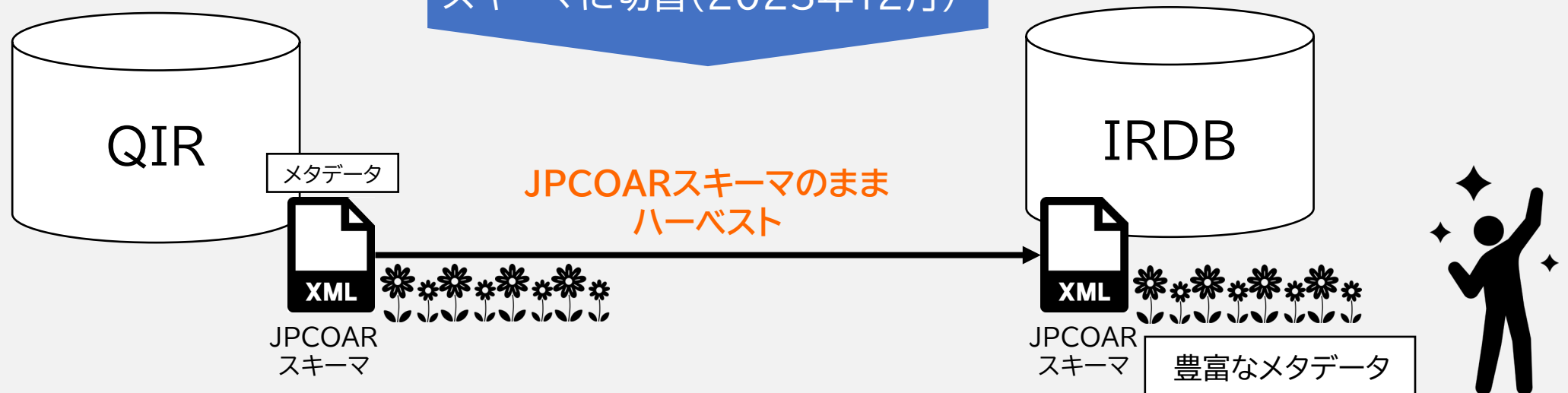
- はじめに
- JPCOARスキーマ対応(1)
メタデータ管理
- **JPCOARスキーマ対応(2)**
ハーベスト形式の移行
- リポジトリ登録インセンティブ対応
メタデータ一括登録
- おわりに

JPCOARスキーマ形式でのハーベスティング開始

以前



現在



ハーベスティングのテストで大量エラー

ハーベストログ送信先 qir@jimu.kyushu-u.ac.jp
次回ハーベスト起算日時 2023/09/06(水) - 10:56
前回ハーベスト開始日時 2023/09/06(水) - 10:56
前回ハーベスト終了日時 2023/09/06(水) - 11:44
前回ハーベスト対象データ件数 55702
前回ハーベスト登録件数 2778
前回ハーベスト更新件数 50067
前回ハーベスト削除件数 273
前回ハーベストデータ取得エラー 0
前回ハーベストステータス 正常
前回ハーベスト通知 前回ハーベスト開始日時: 2023-09-06 10:56:38
前回ハーベスト終了日時: 2023-09-06 11:44:11
ハーベスト対象データ件数: 55702 登録件数: 2778 更新件数: 50067 削除件数: 273 データ取得エラー: 0
レコードエラー件数: 2584 項目エラー件数: 9991 ワーニング件数: 7749 項目変換件数: 4

ハーベスト形式を切替えるためNIIのテスト環境で事前にJPCOARスキーマ形式でのハーベストテストを実施

55,702件中2,584件のメタデータで6,470件(19,410行)のレコードエラーが発生

レコードエラーを解消しないとIRDBにそのメタデータが登録されない

確認先URL: <https://t.irdb.nii.ac.jp/usercontents>

メタデータID

Gold OA 該当アイテムのURL = <https://doi.org/10.5109/10070>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:2324/10070

- レコードエラー: アイテムはIRDBに登録されていません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

エラーメッセージ

Gold OA 該当アイテムのURL = <https://doi.org/10.15017/10324>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:2324/10324

- レコードエラー: アイテムはIRDBに登録されていません。

[関連情報] JaLCに登録する場合、関連識別子がURIであれば、relationTypeは必須です。
[JaLC DOI 必須項目] relationType 規定値以外の値が設定されています。

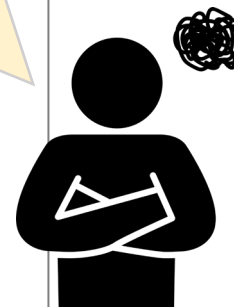
Gold OA 該当アイテムのURL = <https://doi.org/10.15017/10578>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:2324/10578

- レコードエラー: アイテムはIRDBに登録されていません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

エラーログが膨大すぎてどこから手を付けてよいかわからない……



正規表現を使ってハーベストログを整形

Gold OA 該当アイテムのURL = <https://doi.org/10.5109/10070>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:2324/10070

- レコードエラー：アイテムはIRDBに登録されていません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

Gold OA 該当アイテムのURL = <https://doi.org/10.15017/10324>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:15017/10324

- レコードエラー：アイテムはIRDBに登録されていません。

[関連情報] JaLCに登録する場合、関連識別子がURIであれば、relationTypeは必須です。

[JaLC DOI 必須項目] relationType 規定値以外の値が設定されています。

Gold OA 該当アイテムのURL = <https://doi.org/10.15017/10578>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:15017/10578

- レコードエラー：アイテムはIRDBに登録されていません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

Gold OA 該当アイテムのURL = <https://doi.org/10.15017/10580>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:15017/10580

- レコードエラー：アイテムはIRDBに登録されていません。

[JaLC DOI 必須項目] creator Nameが見つかりません。

Gold OA 該当アイテムのURL = <https://doi.org/10.15017/10592>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:15017/10592

- レコードエラー：アイテムはIRDBに登録されていません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

Gold OA 該当アイテムのURL = <https://doi.org/10.15017/10594>

https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefix=jpcoar_2.0&identifier=oai:catalog.lib.kyushu-u.ac.jp:15017/10594

- レコードエラー：アイテムはIRDBに登録されていません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

テキストエディタの正規表現機能を使って
エラーログから不要部分をカット

- ・ 行頭のGold OAから最後の/までを削除
- ・ https://で始まる行を削除
- ・ 「● レコードエラー」で始まる行を削除
- ・ 空白行を削除
(さらにExcelで処理しやすくするために)
- ・ 行頭の全角スペース×2をタブに置換



10070

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

10324

[関連情報] JaLCに登録する場合、関連識別子がURIであれば、relationTypeは必須です。

[JaLC DOI 必須項目] relationType 規定値以外の値が設定されています。

10578

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

10580

[JaLC DOI 必須項目] creator Nameが見つかりません。

10592

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

10594

[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

テキストエディタによる正規表現を使った置換処理

無題-1 * - Mery

ファイル(E) 編集(E) 検索(S) 表示(V) マクロ(M) ツール(T) ウィンドウ(W) ヘルプ(H)

無題-1 *

1 ■ 該当アイテムのURL = <https://doi.org/10.5109/10070>↓
2 https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefi
3 ↓
4 ☐ ● レコードエラー：アイテムはIRDBに登録さ
5 ☐ [JaLC DOI 必須項目] creatorName 言語属性
6 ☐ [JaLC
7 ↓
8 ■ 該当アイテムのURL = <https://doi.org/10.15017/1058>
9 https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefi
10 ↓
11 ☐ ● レコードエラー：アイテムはIRDBに登録されていま
12 ☐ [関連情報] JaLCに登録する場合、関連識別
13 ☐ [JaLC DOI 必須項目] relationType 規定値
14 ↓
15 ■ 該当アイテムのURL = <https://doi.org/10.15017/1058>
16 https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefi
17 ↓
18 ☐ ● レコードエラー：アイテムはIRDBに登録されていま
19 ☐ [JaLC DOI 必須項目] creatorName 言語属性が見つ
20 ↓
21 ■ 該当アイテムのURL = <https://doi.org/10.15017/1058>
22 https://catalog.lib.kyushu-u.ac.jp/mmd/mmd_api/oai-pmh/?verb=GetRecord&metadataPrefi
23 ↓
24 ☐ ● レコードエラー：アイテムはIRDBに登録されていません。↓
25 ☐ [JaLC DOI 必須項目] creator Nameが見つかりません。↓
26 ↓

テキストエディタMeryの場合

検索→置換

置換

検索する文字列(I):
^■.*/

置換後の文字列(P):

☐ 大文字と小文字を区別する(C)
☒ 正規表現を使用する(X)
☐ 単語のみ検索する(W)
☐ 選択した範囲のみ(S)
☐ 終了したら閉じる(L)

次を検索(F)
置換(R)
すべて置換(A)
閉じる

行頭の■半角スペースから
行中の最後の/までの
任意の文字列にマッチ

置換後の文字列を指定せず
正規表現にチェックを入れて
「置換」をクリックすると
マッチングした部分が削除

「すべて置換」だと
ファイル全体で削除

よく使う記号

- ^ 行頭
- \$ 行末
- . 改行を除く任意の文字
- * 直前の文字が1個以上
- + 直前の文字が0個以上
- \n 改行
- \t タブ

Excelを使ってハーベストログを整形

ピボットテーブルを使えるようにExcelで整形

Book2 - Excel

検索

挿入 削除 書式 並べ替えとフィルター 検索と選択

セル

検索(E)... 置換(R)... ジャンプ(G)... 条件を選択してジャンプ(S)... 数式(U) コメント(M) 条件付き書式(C) 定数(N) データの入力規則(Y) オブジェクトの選択(Q) オブジェクトの選択と表示(B)

B列を挿入し
B1にA1の値を
コピーしておく

B2

=IF(A2="",B1,A2)

左のセル(A2)が空欄かどうかチェック

- 空欄の場合
→ 上のセル(B1)の値をセット
- 空欄でない場合
→ A2の値をセット

計算式をB列の最後までコピー

	A	B	C
1	10070	10070	
2		10070	[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
3			[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
4	10324		
5			[関連情報] JaLCに登録する場合、関連識別子がURIであれば、relationTypeは必須です。
6			[JaLC DOI 必須項目] relationType 規定値以外の値が設定されて
7	10578		
8			[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
9	10580		
10			[JaLC DOI 必須項目] creator Nameが見つかりません。
11	10592		
12			[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
13	10594		
14			[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

Excel固有の機能（条件を選択してジャンプ）でも可能

選択オプション

選択

- ☐ コメント(C)
- ☐ 定数(Q)
- ☐ 数式(E)
- ☐ 数値(U)
- ☐ 文字(X)
- ☐ 論理値(G)
- ☐ エラー値(E)
- ☒ 空白セル(K)
- ☐ アクティブセル領域(B)
- ☐ アクティブセルの配列(A)
- ☐ オブジェクト(B)
- ☐ アクティブ行との相違(W)
- ☐ アクティブ列との相違(M)
- ☐ 参照元(P)
- ☐ 参照先(D)
- ☒ 1レベルのみ(I)
- ☐ すべてのレベル(L)
- ☐ 最後のセル(S)
- ☐ 可視セル(V)
- ☐ 条件付き書式(I)
- ☐ データの入力規則(Y)
- ☒ すべて(L)
- ☐ 同じ入力規則(E)

OK キャンセル

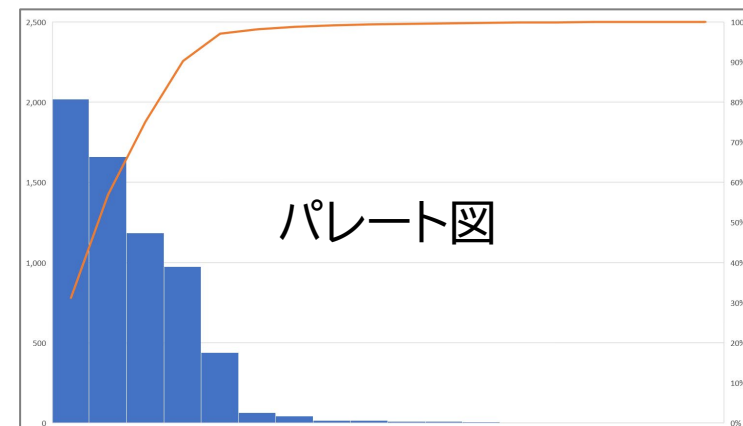
	A	B	C
1	10070	10070	
2		10070	[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
3		10070	[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
4	10324	10324	
5		10324	[関連情報] JaLCに登録する場合、関連識別子がURIであれば、relationTypeは必須です。
6		10324	[JaLC DOI 必須項目] relationType 規定値以外の値が設定されて
7	10578	10578	
8		10578	[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
9	10580	10580	
10		10580	[JaLC DOI 必須項目] creator Nameが見つかりません。
11	10592	10592	
12		10592	[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。
13	10594	10594	
14		10594	[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。

ピボットテーブルを使ってエラー傾向を分析

C31		
	A	B
1	エラーログに基づいて作成したピボットテーブル	
2		
3	行ラベル	個数 / メタデータID
4	[JaLC DOI 必須項目] Abstractの言語 en がありません	7
5	[JaLC DOI 必須項目] affiliationName 言語属性が見つかりません。	2020
6	[JaLC DOI 必須項目] creator Nameが見つかりません。	66
7	[JaLC DOI 必須項目] creatorName 言語属性が見つかりません。	1184
8	[JaLC DOI 必須項目] datacite:date または dcndl:dateGranted が必要です。	45
9	[JaLC DOI 必須項目] dcndl:dateGranted がありません。	441
10	[JaLC DOI 必須項目] file がありません。	10
11	[JaLC DOI 必須項目] funderName en 以外の値が設定されています。	10
12	[JaLC DOI 必須項目] Publisher en 以外の値が設定されています。	3
13	[JaLC DOI 必須項目] publisher がありません。	17
14	[JaLC DOI 必須項目] relationType 規定値以外の値が設定されています。	976
15	[JaLC DOI 必須項目] 同一言語のAbstractが2件以上あります。	18
16	[jalcc] DO付与済みのアイテムから、DOIが削除されています。(10.15017/16905)	1
17	[タイトル] 言語が重複しています。(ja)	3
18	[関連情報] DataCiteに登録する場合、関連識別子がURIであれば、relationTypeは必須です。	5
19	[関連情報] JaLCに登録する場合、関連識別子がDOIであれば、relationTypeは必須です。	2
20	[関連情報] JaLCに登録する場合、関連識別子がURIであれば、relationTypeは必須です。	1661
21	[作成者] 作成者が設定されていません。	1
22	総計	6470

- エラーメッセージは18種類
- 発生件数が多いエラーと少ないエラーがある
- 言語属性(en/ja等)の設定ミスが多い

エラーの傾向を分析
→メンテナンス作業の戦略を検討



とりあえず
件数の多い言語属性
系のエラーから
片づけるか……



JPCOARスキーマ対応: ハーベスト形式の移行

- 作業表を作り、係内で分担してメタデータ修正作業
- 2023年12月19日 NIIによるテストハーベストでレコードエラー0件
翌12月20日 本番環境でのハーベスト実施
 - ハーベスト形式をjunii2からJPCOARスキーマv.2.0に移行
 - 2017年の段階で内部的にはJPCOARスキーマを使っていたため
5年越しの悲願達成
- (非公式情報) IRDBでのJPCOARスキーマv.2.0ハーベスト
九州大学がNIMSに続き全国2例目
- JAIRO Cloud(Weko3)のJPCOARスキーマv.2.0対応
2024年10月時点で現在進行中

- はじめに
- JPCOARスキーマ対応(1)
メタデータ管理
- JPCOARスキーマ対応(2)
ハーベスト形式の移行
- **リポジトリ登録インセンティブ対応**
メタデータ一括登録
- おわりに

リポジトリ登録インセンティブ対応

- リポジトリ登録したことがある在職教員
 - 紀要論文等、代表者による一括登録を除くとわずかに10%未満
- 大学改革推進経費制度によるインセンティブ化を検討
 - リポジトリ登録者の割合に応じて部局経費を増額配分
 - 50%で+1%
 - 70%で+2%
- 案に対してGold OA志向の一部部局から不満
 - **CCライセンス付きGold OA論文**を教員の登録作業の負担なく**図書館で一括登録**することで合意

CCライセンス付きGold OA論文のメタデータ一括登録

作業手順

1. 登録可能な論文を特定1

- 教員業績データベース(Q-RADeRS)から論文のDOIを取得
- 重複DOIやリポジトリ登録済みDOIを除外

2. 登録可能な論文を特定2

- Unpaywallから論文のライセンスとOAステータスを取得
- 対象外の論文を除外

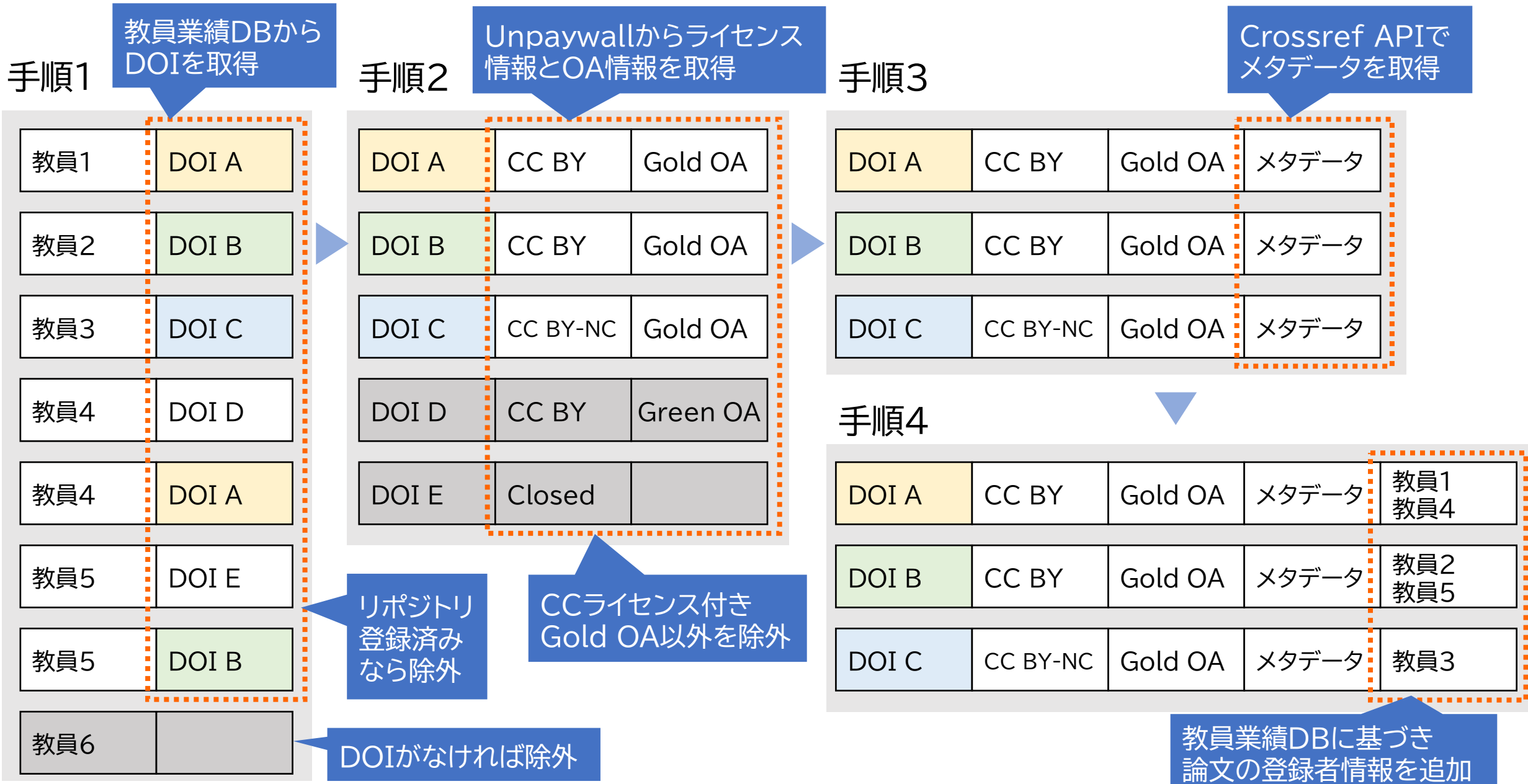
3. Crossref APIで論文のメタデータを取得

- 生成AI(ChatGPT)でPythonコードを作成
- GakuNin RDMでPythonコードを実行

4. 論文メタデータをリポジトリシステムにインポート

- 論文メタデータをExcel形式の一括登録用フォーマットに加工
- 登録者情報として著者の教員IDを紐づけ

CCライセンス付きGold OA論文のメタデータ一括登録



手順1: VBAを使ってDOIデータを正規化

Sub 選択範囲のDOI正規化()

Dim selectedRange As Range

Dim cell As Range

' アクティブなシートから選択範囲を取得

Set selectedRange = Selection

' 選択範囲内の各セルに対して処理を実行

For Each cell In selectedRange

' 1. セル内改行を削除

cell.Value = Replace(cell.Value, vbNewLine, " ")

' 2. セル内の全角文字を半角文字に変換

cell.Value = WorksheetFunction.Trim(Application.WorksheetFunction.Clean(cell.Value))

' 3. 行頭及び行末のスペースを削除

cell.Value = Trim(cell.Value)

' 4. 行頭の"doi:", "doi: ", "doi "を大文字・小文字を区別せずに削除

cell.Value = Replace(cell.Value, "doi:", "", , , vbTextCompare)

cell.Value = Replace(cell.Value, "doi: ", "", , , vbTextCompare)

cell.Value = Replace(cell.Value, "doi ", "", , , vbTextCompare)

' 5. 行末の"."を削除

If Right(cell.Value, 1) = "." Then

cell.Value = Left(cell.Value, Len(cell.Value) - 1)

End If

' 6. 行頭の"https://doi.org/", "http://doi.org/", "https://dx.doi.org/", "http://dx.doi.org/", "doi.org/", "org/"を削除

cell.Value = Replace(cell.Value, "https://doi.org/", "", , , vbTextCompare)

cell.Value = Replace(cell.Value, "http://doi.org/", "", , , vbTextCompare)

cell.Value = Replace(cell.Value, "https://dx.doi.org/", "", , , vbTextCompare)

cell.Value = Replace(cell.Value, "http://dx.doi.org/", "", , , vbTextCompare)

cell.Value = Replace(cell.Value, "doi.org/", "", , , vbTextCompare)

cell.Value = Replace(cell.Value, "org/", "", , , vbTextCompare)

' 7. 値に"/"が含まれないセルを空値にする

If InStr(1, cell.Value, "/") = 0 Then

cell.ClearContents

End If

Next cell

End Sub

改行コードが
含まれている

全角文字で
入力されている

空白スペースが含まれている

見出しが追加されている

ピリオドが入っている

フォーマットが
バラバラ

そもそもDOI
ではない

教員業績データベースに教員が論文のDOIを入力しているが、フォーマットが一定していないためVBAで正規化

VBA (Visual Basic Application)はExcelやWord等、マイクロソフト製品で使われるプログラミング言語

定型作業の自動化が可能

- DOI正規化VBA.xlsm
※VBAコードを記述したファイル
- DOIサンプルデータ.xlsx
※作業用ファイル

両方のファイルを開き、作業用ファイル上でA列の任意のDOIを選択→開発→マクロ→「選択範囲のDOI正規化」を実行

VBAコードを記述したファイル上で開発→Visual Basic
で実際のコード内容を確認、編集可能

手順2: Unpaywall



Simple Query Tool

If you want to check a few articles to see if they are Open Access and don't want to mess with the [REST API](#), you're in the right place. You can check up to 1,000 DOIs at once using this tool.

We'll run the list through our API for you and give you a report on the Open Access status of each DOI (note: this tool caches API response, so you may see slight differences between data here and in the API for recently-changed articles). In a few minutes, you'll get an email with the results, as up to three attachments:

- A CSV file that lets you easily import results into a spreadsheet like Excel.
- A [JSON Lines](#) file that shows what you'd get if you called our API once for each DOI.
- An Excel spreadsheet.

Don't forget to check the [schema documentation](#), which includes definitions that will help you interpret the the result files.

Submit DOIs

Paste DOIs here, one per line

DOIを一度に1,000件まで指定可能

Select Result Formats

☒ CSV ☒ JSONL ☐ Excel (.xlsx)

Your email

指定したメールアドレスにUnpaywallが収集する論文のオープンアクセス情報を送信

SUBMIT YOUR DOIS

<https://unpaywall.org/products/simple-query-tool>

UnpaywallはWebブラウザのプラグインを通じて出版社サイトや論文データベースのWebページから該当論文の本文PDFへのリンクを無料で提供するサービス

サービス提供ため収集された論文のライセンス情報やOA情報がSimple Query ToolやAPIでも提供されている

データフォーマット

<https://unpaywall.org/data-format>

実際のデータ取得例

	OAかどうか	OAの種類	ライセンス
doi	is_oa	oa_status	license
10.1001/jamanetworkopen.2022.45081	TRUE	gold	cc-by-nc-nd
10.1002/1873-3468.14417	FALSE	closed	
10.1002/2475-8876.12282	TRUE	gold	cc-by
10.1002/2475-8876.12293	TRUE	gold	cc-by
10.1002/2475-8876.12299	TRUE	gold	cc-by
10.1002/2475-8876.12312	TRUE	gold	cc-by-nc-nd
10.1002/9783527830190.ch5	FALSE	closed	
10.1002/adfm.202201772	TRUE	green	cc-by-nc-nd
10.1002/adfm.202203003	FALSE	closed	
10.1002/adfm.202204352	FALSE	closed	

ChatGPTでUnpaywall API用のPythonコードを作成

Unpaywall APIでDOIのリストから該当論文の次の値を取得するPythonコードを教えてください。

1. "is_oa" (boolean)
2. "oa_status" (string)
3. "best_oa_location" (object|null) に含まれる "license" (string|null) および "oa_date" (string|null)

DOIのリストは"doi_list.txt"から取得してください。

取得した論文の情報は"unpaywall.tsv"に出力してください。

"unpaywall.tsv"の内容は"doi", "is_oa", "oa_status", "license", "oa_date"としてください。

endpointの例は以下の通りです。

https://api.unpaywall.org/v2/10.1038/nature12373?email=YOUR_EMAIL

最後にDOI数と実行時間を返してください。

Google ColaboratoryでPythonコードを実行

The screenshot shows the Google Colaboratory web interface. At the top, the title bar says 'Untitled0.ipynb' with a star icon. Below it, a menu bar includes 'ファイル' (File), '編集' (Edit), '表示' (View), '挿入' (Insert), 'ランタイム' (Runtime), 'ツール' (Tools), and 'ヘルプ' (Help). A status bar indicates 'すべての変更を保存しました' (All changes saved).

On the left, the 'ファイル' (File) sidebar shows a file explorer with folders like 'sample_data' and files like 'doi_list.txt' and 'unpaywall.tsv'. Annotations point to the 'アップロードしたDOIのリスト' (List of uploaded DOIs) and the 'ファイルのアップロードボタン' (File upload button).

The main code cell contains Python code for fetching DOI information from the Unpaywall API. Annotations highlight the '実行ボタン' (Run button) and the 'メールアドレス' (Email address) placeholder. A large blue box in the center says 'ChatGPTで作成したPythonコード' (Python code created by ChatGPT).

On the right, a preview of the output file 'unpaywall.tsv' is shown, displaying a table of DOI data. An annotation points to it saying '出力したファイルの内容' (Content of the output file).

At the bottom right, a box titled '実行手順' (Execution steps) lists the following steps:

1. Googleアカウントでログイン
<https://colab.google/>
2. 「New Notebook」を選択
3. DOIのリストをアップロード
4. コードセルにPythonコードをコピー
5. 実行ボタンをクリック

```
import requests
import csv
import time
from datetime import datetime

# Replace this with your email
YOUR_EMAIL = 'qir@jimu.kyushu-u.ac.jp'

# Replace this with doi list name
filename = 'doi_list.txt'

# Collects the requested information for a specific DOI
def get_oa_info(doi):
    response = requests.get(f'https://api.unpaywall.org/v2/{doi}?email={YOUR_EMAIL}')
    data = response.json()

    is_oa = data.get('is_oa')
    oa_status = data.get('oa_status')

    best_oa_location = data.get('best_oa_location')
    if best_oa_location:
        license = best_oa_location.get('license')
        oa_date = best_oa_location.get('oa_date')
    else:
        license = None
        oa_date = None

    return [doi, is_oa, oa_status, license, oa_date]

# Writes the collected data to the specified .tsv file
def write_to_tsv(data, file_name):
    with open(file_name, 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerow(['doi', 'is_oa', 'oa_status', 'license', 'oa_date'])
        for row in data:
            writer.writerow(row)
```


手順3: Crossref API

Webブラウザで「[https://api.crossref.org/works/\[DOI\]](https://api.crossref.org/works/[DOI])」にアクセスするとJSON形式で該当論文のメタデータを取得可能

```
{
  "status": "ok",
  "message-type": "work",
  "message-version": "1.0.0",
  "message": {
    "indexed": {
      "date-parts": [
        [
          2023,
          8,
          21
        ]
      ],
      "date-time": "2023-08-21T04:45:19Z",
      "timestamp": 1692593119790
    },
    "reference-count": 31,
    "publisher": "Wiley",
    "issue": "4",
    "license": [
      {
        "start": {
          "date-parts": [
            [
              2022,
              9,
              4
            ]
          ],
          "date-time": "2022-09-04T00:00:00Z",
          "timestamp": 1662249600000
        },
        "content-version": "vor",
        "delay-in-days": 0,
        "URL": "http://creativecommons.org/licenses/by/4.0/"
      }
    ]
  }
}
```

```
  "funder": [
    {
      "DOI": "10.13039/501100001691",
      "name": "Japan Society for the Promotion of Science",
      "doi-asserted-by": "publisher",
      "award": [
        "JP20K21035"
      ],
      "id": [
        {
          "id": "10.13039/501100001691",
          "id-type": "DOI",
          "asserted-by": "publisher"
        }
      ]
    }
  ],
  "content-domain": {
    "domain": [
      "onlinelibrary.wiley.com"
    ],
    "crossmark-restriction": true
  },
  "short-container-title": [
    "Japan Architectural Review"
  ],
  "published-print": {
    "date-parts": [
      [
        2022,
        10
      ]
    ]
  }
}
```

```
  "abstract": "<jats:title>Abstract</jats:title><jats:p>Numerous studies have been conducted using mesoscale numerical weather prediction models to analyze the thermal and wind environments of urban areas around the world. However, weather predictions are highly sensitive to initial conditions. This is the so-called “butterfly effect.” In this study, the effects of initial and boundary conditions upon the reproduction accuracy for wind environments (particularly for the sea breeze) over the Fukuoka metropolitan area were investigated by varying (i) the objective analysis data used for the initial and boundary conditions and (ii) the length of the spin-up calculation period. The results of the simulation were compared with surface observations and Doppler LIDAR observations of the airflow field. The accuracy is thought to decrease under an increase in the length of the spin-up calculation period; however, this was not the case in this study. In terms of the differences in the objective analysis data, no clear trend was observed from the comparisons with surface observations; however, slight differences were observed when comparing with the Doppler LIDAR. The lagged average forecasting method (which reduces the uncertainty of the initial conditions by shifting the calculation start time) was also adopted; the ensemble-averaged results showed a good correspondence with the observations.</jats:p>",
  "DOI": "10.1002/2475-8876.12282",
  "type": "journal-article",
  "created": {
    "date-parts": [
      [
        2022,
        9,
        4
      ]
    ],
    "date-time": "2022-09-04T13:30:06Z",
    "timestamp": 1662298206000
  }
}
```

Crossref APIでメタデータを取得できれば 機関リポジトリへの論文登録が可能

"title": [

"Investigation of the uncertainty of initial and boundary conditions in the hindcasts of flow fields over urban areas using a mesoscale numerical weather prediction model"

"author": [

```
{
  "ORCID": "http://orcid.org/0000-0002-2092-9621",
  "authenticated-orcid": false,
  "given": "Yoichi",
  "family": "Kawamoto",
  "sequence": "first",
  "affiliation": [
    {
      "name": "Faculty of Design Kyushu University Fukuoka Japan"
    }
  ]
}
```

```
"license": [
  {
    "start": { ... }, // 3 items
    "content-version": "vor",
    "delay-in-days": 0,
    "URL": "http://creativecommons.org/licenses/by/4.0/"
  }
]
```

"funder": [

```
{
  "DOI": "10.13039/501100001691",
  "name": "Japan Society for the Promotion of Science",
  "doi-asserted-by": "publisher",
  "award": [
    "JP20K21035"
  ],
  "id": [ ... ] // 1 item
}
```

<学術雑誌論文>

Investigation of the uncertainty of initial and boundary conditions in the hindcasts of flow fields over urban areas using a mesoscale numerical weather prediction model

作成者 作成者名: Kawamoto, Yoichi
所属機関: Faculty of Design, Kyushu University

本文言語 英語
出版者 Wiley
発行日 2022-09-04
収録物名 JAPAN ARCHITECTURAL REVIEW
巻 5
号 4
開始ページ 517
終了ページ 529
出版タイプ Version of Record
アクセス権 open access
権利関係 Creative Commons Attribution 4.0 International
関連DOI 以下と同一: <https://doi.org/10.1002/2475-8876.12282>
関連URI
関連HDL

実際のリポジトリ
登録例

本文ファイル

ファイル	ファイルタイプ	サイズ	閲覧回数	説明
 7160876	pdf	1.22 MB	73	

詳細

PISSN 2475-8876
EISSN 2475-8876
レコードID 7160876
主題 mesoscale numerical weather prediction
sea breeze
uncertainty
wind environment
WRF

<https://hdl.handle.net/2324/7160876>

助成情報

助成機関名: Japan Society for the Promotion of Science
研究課題番号: JP20K21035

"language": "en",

"publisher": "Wiley",

```
"issued": {
  "date-parts": [
    [
      2022,
      9,
      4
    ]
  ]
},
```

"container-title": ["JAPAN ARCHITECTURAL REVIEW"],

"volume": "5",

"issue": "4",

"page": "517-529",

```
"issn-type": [
  {
    "value": "2475-8876",
    "type": "print"
  },
  {
    "value": "2475-8876",
    "type": "electronic"
  }
],
```

ChatGPTでCrossref API用のPythonコードを作成

以下の手順で論文の書誌情報を取得するPythonコードを教えてください。

その際、クエリに「mailto=qir@m.kyushu-u.ac.jp」パラメータを含めるか、または、User-Agentヘッダーに「mailto:qir@jimu.kyushu-u.ac.jp」を含めます。また、1秒間に50回のリクエストレートとしてください。実行後、標準出力に完了メッセージとかかった時間を出力してください。

1. 1行に1つDOIが記録されている”doi_list.txt”という名称のテキストファイルを読み込む。
2. Crossref APIから以下の書誌情報を取得する。
 - ・DOI (DOI: "string")
 - ・タイトル (title: ["string"])
 - ・ライセンス (license: [{"URL": "string", "content-version": "string"}]で、"content-version"が"vor"の場合の"URL"の値) ※該当しない場合は空欄
 - ・出版社 (publisher: "string")
 - ・発行年 (issued: {"date-parts": [[y, m, d]]}をYYYY-MM-DDまたはYYYY-MMまたはYYYYのフォーマットに変換)
 - ・言語 (language: "string")
 - ・資料種別 (type: "string")
 - ・pISSN (issn-type: [{"type": "string", "value": ["string"]}]]で、typeが"print"の場合の"value"の値) ※該当しない場合は空欄
 - ・eISSN (issn-type: [{"type": "string", "value": ["string"]}]]で、typeが"electronic"の場合の"value"の値) ※該当しない場合は空欄
 - ・ジャーナル名 (container-title: ["string"])
 - ・巻 (volume: "string")
 - ・号 (issue: "string")
 - ・ページ (page: "string")
 - ・要約 (abstract: "string")
 - ・件名 (subject: ["string"])
 - ・助成機関名 (funder [{"name": "string"}])
 - ・助成機関DOI (funder [{"DOI": "string"}])
 - ・助成番号 (funder [{"award": ["string"]}])
3. 取得した情報から改行コードを除外する。また、前後の空白文字も除外する(トリム)。
4. 取得した情報を”paper_matadata.tsv”という名称のタブ区切りのテキストファイルに出力する。

項目数が多いため
複雑そうに見えるが
やっていることは単純

JSON形式の型を踏まえて
質問するとエラーが出にくい

- ・ 文字列 ("...")
- ・ 数値 (2024など)
- ・ オブジェクト ({ ... })
- ・ 配列 ([...])

ChatGPTでCrossref API用のPythonコードを作成

Crossref APIを使って論文の著者情報を取得するPythonコードを教えてください。
その際、クエリに「mailto=qir@m.kyushu-u.ac.jp」パラメータを含めるか、または、User-Agentヘッダーに「mailto:qir@m.kyushu-u.ac.jp」を含めます。また、1秒間に50回のリクエストレートとしてください。
処理が完了したら標準出力に完了メッセージと処理件数と処理時間を出力してください。

処理手順は以下の通りです。

1. DOIのリストを"doi_list.txt"から取得します。
2. DOIから以下の著者情報を取得します。
 - ・ "ORCID": "string" (http://...となっていたらhttps://...に変換)
 - ・ "family": "string" 及び "given": "string" を", "で結合
 - ・ "affiliation": [{"name": "string"}]の"string"を複数ある場合は", "で結合
 - ・ "sequence": "string"
3. 取得した著者情報は以下の構成で"authors_metadata.tsv"に出力します。
"DOI queue number"¥t "DOI" ¥t "author queue number" ¥t "ORCID" ¥t "given, family" ¥t "affiliation" ¥t "sequence"

著者情報のメタデータは共著者数に応じて
取得するデータ量が増減するため
それ以外のメタデータとは分けて取得

なお、Crossref APIで取得する著者情報のデータ構造は以下の通りです。

```
{
  "message": {
    "author": [
      {
        "ORCID": "string",
        "suffix": "string",
        "given": "string",
        "family": "string",
        "affiliation": [{"name": "string"}],
        "name": "string",
        "prefix": "string",
        "sequence": "string"
      }
    ]
  }
}
```

JSONのデータ構造を
そのままサンプルとして
提示すると楽

GakuNin RDMでPythonコードを実行

 GakuNin RDM

サポート九州大学

選択リセット

The place to share your research

GakuNin RDMは、国立情報学研究所が提供する研究データ管理システムです。研究チームが持つ多様なデータを一元管理し、効率的で公正なデータ活用をサポートします。



GakuNin RDM

Copyright © 2016-2024 国立情報学研究所 | 利用規約 | プライバシーポリシー

<https://rdm.nii.ac.jp/>

NIIが提供する研究データ管理基盤GakuNin RDMのデータ解析機能でPythonコードを実行可能

GakuNin RDMが使えなければGoogle ColabでもOK

<https://colab.google/>

 GakuNin RDM

マイプロジェクト 検索 サポート Ashikita Takuya

論文メタデータ取得 ファイル Wiki 解析 メンバー アドオン 設定 証跡管理

 論文メタデータ取得

メンバー: Ashikita Takuya, OKAMURA NANAKO, HIRANO KAORU

所属機関: 九州大学

作成日時 2023-07-21 10:44 AM | 最終更新日時 2023-10-02 03:26 PM

カテゴリー: プロジェクト

説明: プロジェクトに簡単な説明を追加してください

ライセンス: ライセンスなし

Wiki

重要な情報、リンク、または画像をここに追加して、プロジェクトを説明してください。

ファイル

ストレージプロバイダーをクリックするか、ドラッグ&ドロップしてファイルをアップロードします

名前 最終更新日時

論文メタデータ取得

NII Storage

.binder

.ipynb_checkpoints

author.ipynb 2023-10-02 10:28 AM

column_convert.ipynb 2023-10-02 03:23 PM

column_name_conversion.xlsx 2023-10-02 11:52 AM

crossref.ipynb 2023-10-02 10:28 AM

paper_metadata.tsv 2023-10-02 11:54 AM

paper_metadata_template.tsv 2023-10-02 03:21 PM

unpaywall.ipynb 2023-10-02 10:27 AM

Unpaywall_API.docx 2023-09-15 03:59 PM

引用

コンポーネント

コンポーネントを追加 プロジェクトをリンク

コンポーネントを追加して、プロジェクトを整理します。

タグ

タグを追加してプロジェクトを発見しやすくする

最近の活動

更新 フィルタをクリア

開始日時 終了日時

ユーザでフィルタ (現在のメンバーから選択)

☐ 過去のプロジェクトメンバーも含めて検索する

Ashikita Takuyaが論文メタデータ取得のNII Storageのファイル.binder/environment.ymlを更新しました 2023-10-02 03:26 PM

Ashikita Takuyaが論文メタデータ取得のNII Storageのファイル.ipynb_checkpoints/column_convert-checkpoint.ipynbを更新しました 2023-10-02 03:24 PM

Ashikita Takuyaは論文メタデータ取得のNII Storageからfile.-column_convert.ipynbを削除しました

GakuNin RDMでPythonコードを実行

 GakuNin RDM

マイプロジェクト検索サポートAshikita Takuya

論文メタデータ取得ファイルWiki**解析**メンバーアドオン設定

私の解析環境

このプロジェクトから作った解析環境が 1 個あります。すべての解析環境を表示

解析環境名	URL	最終更新	起動	削除	
thcpr-osfstorage-7zva9vtj	/user/	@kyushu-u.ac.jp/thcpr-osfstorage-7zva9vtj/	2024-10-28 06:11 PM		

新しい解析環境

基本イメージ

Python 3.9 + R 4.1.3 (JupyterLab 3.x, deprecated) ✓
Jupyter Notebook, JupyterLab, RStudio, Shinyが使えます。

作成した解析環境

JupyterLabの起動ボタン

変更

追加パッケージ

apt-get **+ 追加**

conda pandas: ✕ openpyxl: ✕ **+ 追加**

pip **+ 追加**

R (CRAN) **+ 追加**

自動実行スクリプト



環境作成

このプロジェクトのデフォルトストレージの内容がコピーされます。

Pythonコードを実行可能な仮想環境を作成

新しい解析環境を作成: <https://binder.cs.rcos.nii.ac.jp>

GakuNin RDMでPythonコードを実行

The screenshot displays the GakuNin RDM interface. On the left, a file browser shows a list of files, with 'crossref.ipynb' highlighted. A blue callout box points to this file with the text 'Jupyter Notebook 形式のファイル'. The main area shows the 'crossref.ipynb' notebook with a code cell [31] containing Python code for fetching paper metadata from Crossref. A blue callout box points to the 'Run' button (a play icon) with the text '実行ボタン'. Another blue callout box on the right contains the text 'ChatGPTで作成した Pythonコード'. The code cell output is not visible, but the code is as follows:

```
[31]: import requests
import json
import time

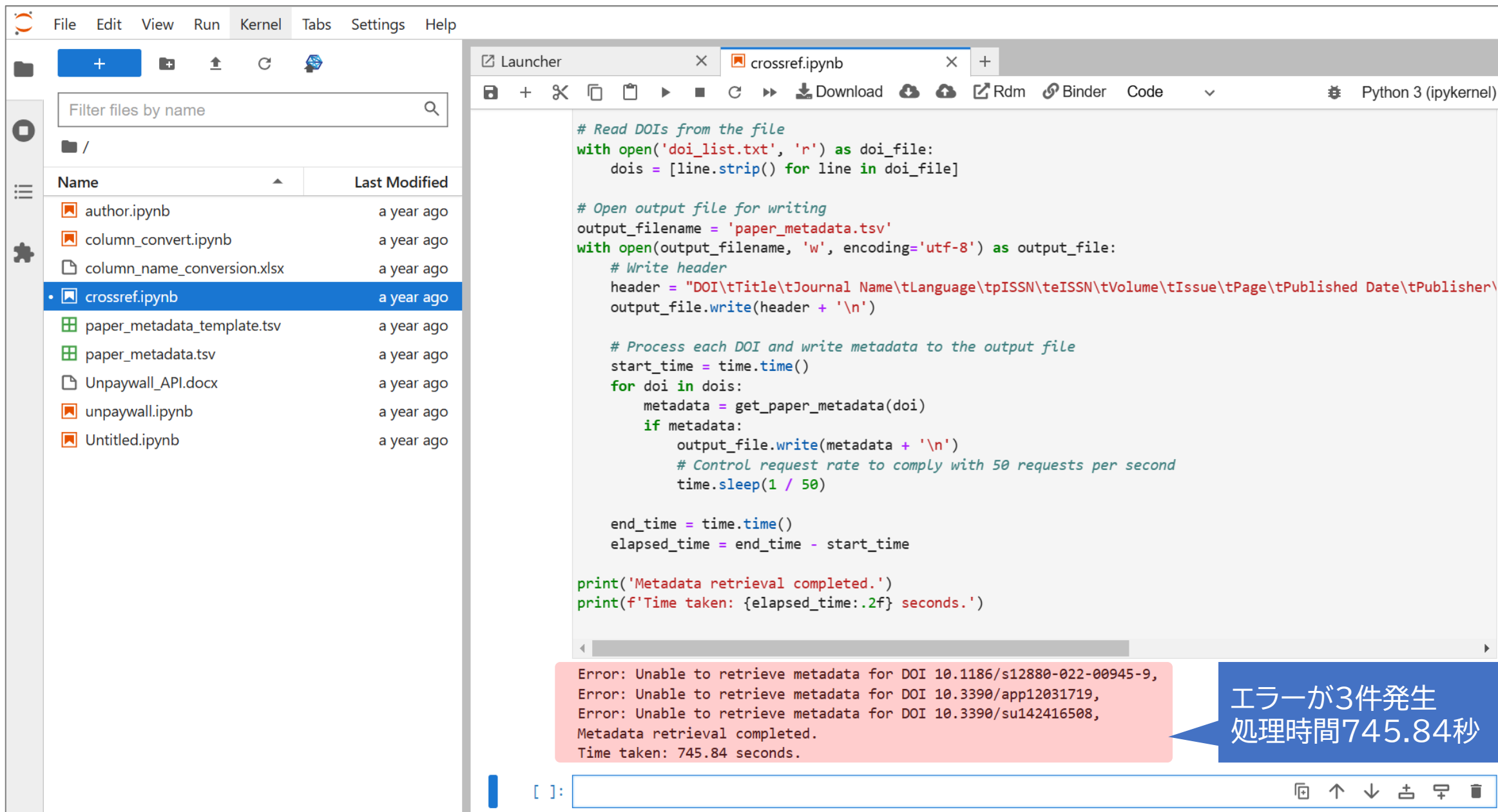
# Function to get paper metadata for a DOI
def get_paper_metadata(doi):
    api_url = f'https://api.crossref.org/works/{doi}'
    headers = {
        'User-Agent': 'mailto:qir@m.kyushu-u.ac.jp'
    }
    response = requests.get(api_url, headers=headers)

    if response.status_code == 200:
        data = response.json()
        metadata = data['message']

    # Extract required metadata
    doi = metadata.get('DOI', '')
    title = metadata.get('title', [''])[0]
    journal_name = metadata.get('container-title', [''])[0]
    language = metadata.get('language', '')
    p_issn = next((item['value'] for item in metadata.get('issn-type', []) if item.get('type') == 'print'),
e_issn = next((item['value'] for item in metadata.get('issn-type', []) if item.get('type') == 'electronic'),
volume = metadata.get('volume', '')
issue = metadata.get('issue', '')
page = metadata.get('page', '')
issued_date_parts = metadata.get('issued', {}).get('date-parts', [[]])[0]

# Format published date based on available parts
if issued_date_parts:
    if len(issued_date_parts) == 3:
        published_date_formatted = f'{issued_date_parts[0]:04d}-{issued_date_parts[1]:02d}-{issued_date_parts[2]:02d}'
    elif len(issued_date_parts) == 2:
        published_date_formatted = f'{issued_date_parts[0]:04d}-{issued_date_parts[1]:02d}'
    elif len(issued_date_parts) == 1:
        published_date_formatted = f'{issued_date_parts[0]:04d}'
```


GakuNin RDMでPythonコードを実行



The screenshot displays the GakuNin RDM interface. On the left, a file explorer shows a list of files, with `crossref.ipynb` selected. The main area shows the Jupyter Notebook code for `crossref.ipynb`, which is running in a Python 3 (ipykernel) environment. The code reads a file `doi_list.txt`, writes a header to `paper_metadata.tsv`, and processes each DOI to retrieve metadata. The output shows an error for three DOIs and a total time of 745.84 seconds.

```
# Read DOIs from the file
with open('doi_list.txt', 'r') as doi_file:
    dois = [line.strip() for line in doi_file]

# Open output file for writing
output_filename = 'paper_metadata.tsv'
with open(output_filename, 'w', encoding='utf-8') as output_file:
    # Write header
    header = "DOI\tTitle\tJournal Name\tLanguage\tISSN\tVolume\tIssue\tPage\tPublished Date\tPublisher\n"
    output_file.write(header + '\n')

# Process each DOI and write metadata to the output file
start_time = time.time()
for doi in dois:
    metadata = get_paper_metadata(doi)
    if metadata:
        output_file.write(metadata + '\n')
        # Control request rate to comply with 50 requests per second
        time.sleep(1 / 50)

end_time = time.time()
elapsed_time = end_time - start_time

print('Metadata retrieval completed.')
print(f'Time taken: {elapsed_time:.2f} seconds.')
```

Error: Unable to retrieve metadata for DOI 10.1186/s12880-022-00945-9,
Error: Unable to retrieve metadata for DOI 10.3390/app12031719,
Error: Unable to retrieve metadata for DOI 10.3390/su142416508,
Metadata retrieval completed.
Time taken: 745.84 seconds.

エラーが3件発生
処理時間745.84秒

GakuNin RDMでPythonコードを実行

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name Last Modified

- author.ipynb a year ago
- column_convert.ipynb a year ago
- column_name_conversion.xlsx a year ago
- crossref.ipynb a year ago
- paper_metadata_template.tsv a year ago
- paper_metadata.tsv a year ago**
- Unpaywall_API.docx a year ago
- unpaywall.ipynb a year ago
- Untitled.ipynb a year ago

Pythonコードの実行によりTSV形式で出力されたファイル

Launcher x crossref.ipynb x paper_metadata.tsv x +

Delimiter: tab

	DOI	Title	License	Publisher	Published Date	Language
1	10.1001/jamanetw...	Concordance Betwe...		American Medical A...	2022-12-05	
2	10.1002/2475-887...	Investigation of the...	http://creativecom...	Wiley	2022-09-04	
3	10.1002/2475-887...	Stochastic nature of...	http://creativecom...	Wiley		
4	10.1002/2475-887...	Influence of inlet tu...	http://creativecom...	Wiley		
5	10.1002/2475-887...	Computational fluid...	http://creativecom...	Wiley		
6	10.1002/adma.202...	Heterointerface Cre...	http://creativecom...	Wiley		
7	10.1002/adma.202...	A Materials Accelera...	http://creativecom...	Wiley	2022-12-14	
8	10.1002/adpr.2022...	Embedding Optical ...	http://creativecom...	Wiley	2022-05-18	
9	10.1002/aesr.2022...	FeF₃ ...	http://creativecom...	Wiley	2022-10-07	
10	10.1002/anie.2022...	Engineering Transp...	http://creativecom...	Wiley	2022-11-07	
11	10.1002/ansa.2022...	Precision evaluation...	http://creativecom...	Wiley	2022-10	
12	10.1002/cam4.4537	IL-34 in hepatoblast...	http://creativecom...	Wiley	2022-02-08	
13	10.1002/cctc.2022...	The Effect of Ru Pre...	http://creativecom...	Wiley	2022-06-07	
14	10.1002/deo2.91	Comparisons of out...	http://creativecom...	Wiley	2022-01-26	
15	10.1002/deo2.93	An adult case of co...	http://creativecom...	Wiley	2022-02-02	
16	10.1002/ecs2.4182	Early snowmelt by ...	http://creativecom...	Wiley	2022-07	
17	10.1002/ehf2.13934	Home-based cardia...	http://creativecom...	Wiley	2022-05-09	
18	10.1002/elsa.2022...	Numerical analysis ...	http://creativecom...	Wiley	2022-04-27	
19	10.1002/epi4.12616	Impact of <scp>CO...	http://creativecom...	Wiley	2022-06-12	
20	10.1002/hbm.25899	<scp>Higher-resolu...	http://creativecom...	Wiley	2022-05-07	
21	10.1002/hep4.1838	Impact of Nuclear F...		Ovid Technologies (...)	2021-10-23	
22	10.1002/hep4.1911	Direct Conversion o...		Ovid Technologies (...)	2022-02-27	
23	10.1002/hep4.2033	Up-regulated LRRN...		Ovid Technologies (...)	2022-07-27	
24	10.1002/igh3.12777	Comparison of the ...	http://creativecom...	Wiley	2022-06-08	

Crossref APIで取得した論文のメタデータ

手順4: Excelで一括登録用フォーマットを作成

著者情報以外のメタデータ

DOI A	CC BY	Gold OA	メタデータ	
DOI B	CC BY	Gold OA	メタデータ	
DOI C	CC BY-NC	Gold OA	メタデータ	

著者情報メタデータ

DOI A	ORCID1	教員1	所属1
DOI A	ORCID4	教員4	所属4
DOI B	ORCID2	教員2	所属2
DOI B	ORCID5	教員5	所属5
DOI C	ORCID3	教員3	所属3

DOIをキーにして著者情報とそれ以外のメタデータを統合したい

論文と著者が1対1の関係ならExcelのXLOOKUP関数で処理できるが1対多なので使えない

リポジトリ一括登録用フォーマット

DOI A	CC BY	Gold OA	メタデータ	ORCID1	教員1	所属1	ORCID4	教員4	所属4
DOI B	CC BY	Gold OA	メタデータ	ORCID2	教員2	所属2	ORCID5	教員5	所属5
DOI C	CC BY-NC	Gold OA	メタデータ	ORCID3	教員3	所属3	null	null	null

ExcelのFILTER関数とTRANSPOSE関数を使う方法

	A	B	C	D
1	DOI A	ORCID1	教員1	所属1
2	DOI A	ORCID4	教員4	所属4
3	DOI B	ORCID2	教員2	所属2
4	DOI B	ORCID5	教員5	所属5
5	DOI C	ORCID3	教員3	所属3

テキストエディタ
を使ってTabを
■に変換

	A	B
1	DOI A	ORCID1■教員1■所属1
2	DOI A	ORCID4■教員4■所属4
3	DOI B	ORCID2■教員2■所属2
4	DOI B	ORCID5■教員5■所属5
5	DOI C	ORCID3■教員3■所属3

	A	B	C
1	DOI A	ORCID1■教員1■所属1	
2	DOI A	ORCID4■教員4■所属4	
3	DOI B	ORCID2■教員2■所属2	
4	DOI B	ORCID5■教員5■所属5	
5	DOI C	ORCID3■教員3■所属3	
6			
7	DOI A	=TRANSPOSE(FILTER(\$B\$1:\$B\$5,\$A\$1:\$A\$5=A7,""))	
8	DOI B		
9	DOI C		

FILTER関数で同一DOIの
著者情報を全件抽出

TRANSPOSE関数で
著者情報を右側に列挙

7	DOI A	ORCID1■教員1■所属1	ORCID4■教員4■所属4
8	DOI B	ORCID2■教員2■所属2	ORCID5■教員5■所属5
9	DOI C	ORCID3■教員3■所属3	

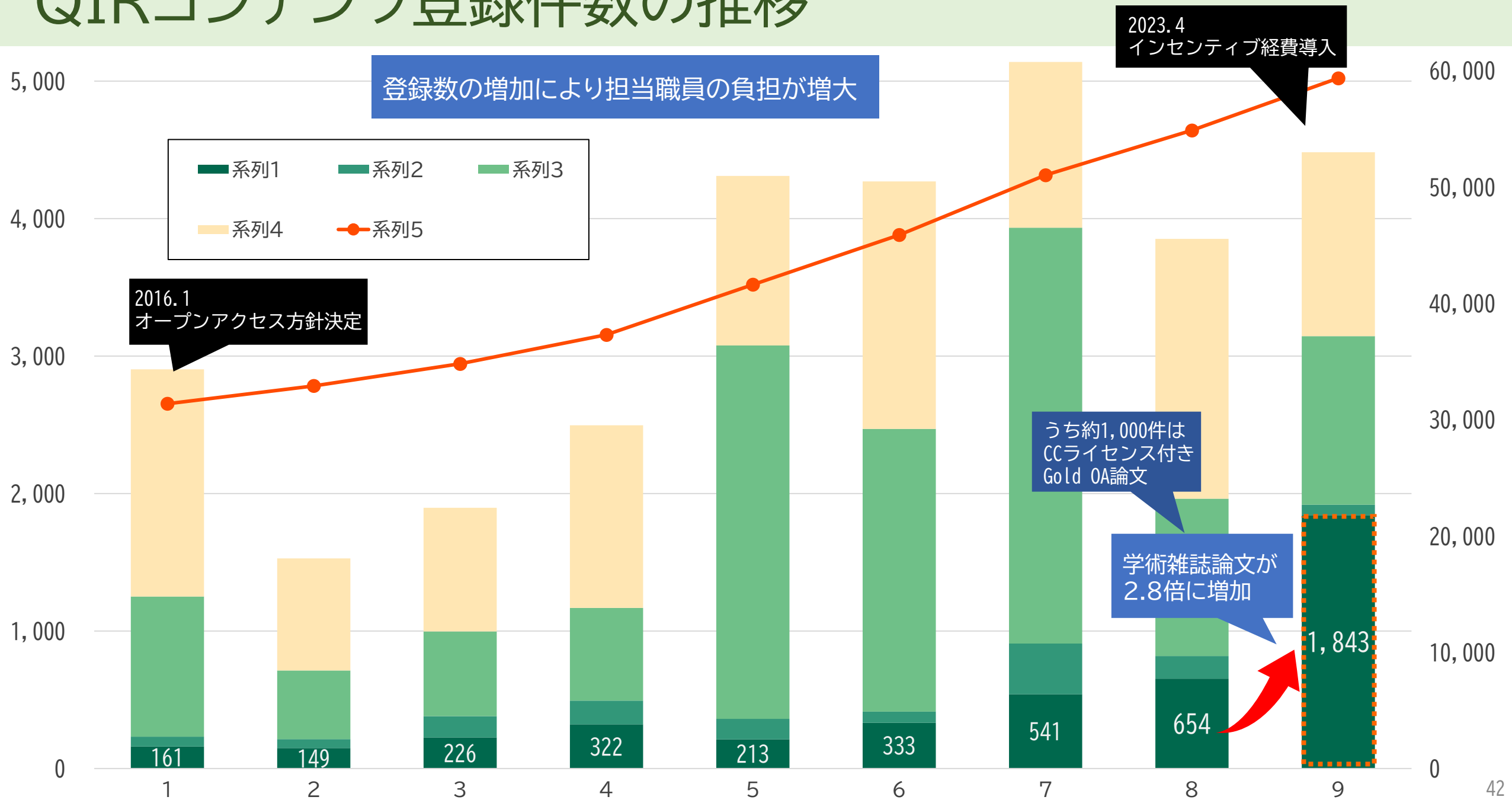
テキストエディタで■をTabに変換

7	DOI A	ORCID1	教員1	所属1	ORCID4	教員4	所属4
8	DOI B	ORCID2	教員2	所属2	ORCID5	教員5	所属5
9	DOI C	ORCID3	教員3	所属3			

リポジトリ登録インセンティブ対応

- 登録者情報は、著者情報と同様にFILTER関数とTRANSPOSE関数を使ってメタデータに紐づけ可能
 - 一括登録用フォーマットを作成して図書館システムにインポート
 - CCライセンス付きGold OA論文のメタデータを1,000件以上
教員の登録負担なしで一括登録
- ※インセンティブ対象教員のリポジトリ登録者が10%から66%に増加
- 本文PDFは学生アルバイトの作業で1件ずつ登録
 - もともとGold OAなのでリポジトリ登録なしでも出版社サイトで本文を読める状態だが、IRDBへのハーベストによってCiNii Researchで検索可能になった（国のOA方針通りの運用???)

QIRコンテンツ登録件数の推移



九州大学のリポジトリ登録支援システム改修案

改修で実現したいこと

