Vol.2019-CH-120 No.12
2019/5/11

デジタル学術情報流通環境に基づく文献目録情報作成の試み —『史学雑誌』文献目録作成を事例に—

小林拓実[†1]　小川潤[†2]

概要：文献目録の作成においては，関連する著作を作成者が確認し，必要性を認めたものを記録するのが一般的である．それを学会誌の業務として行う場合，チェックする対象の雑誌・書籍の数が膨大となり，作業者の負担が非常に大きくなってしまうという困難が生じていた．そこで今回，東京大学人文社会系研究科の史学系大学院生が史学会の依頼を受け行っている『史学雑誌』の西洋史文献目録の作成作業を題材として，Python と，CiNii 等の学術情報基盤で公開される Web API を用いた著作データ取得の自動化を試みた．この結果，作業時間の短縮には成功したものの，取得データと史学系の雑誌として要求されるデータのギャップを補正する必要性，またそのシステムを持続的に運用する上での課題も明らかになった．

キーワード：Web API，学術情報基盤，Python

Attempt to edit a library catalog by applying the distributive systems for digital academic information A case study on *The Historical Journal*

Takumi Kobayashi[†1]　Jun Ogawa[†2]

Abstract: It is general that, in the process of editing a library catalog, each editor searches for all the related books and articles, then decides which of those should be recorded on the catalog. However, in making a comprehensive catalog for the academic journal as a part of entrusted work, it costs him a large amount of time because he needs to consult far much journals than he creates his own research bibliography. Accordingly, we have made a Python program in order to collect the data automatically from the science information infrastructures, such as CiNii, by utilizing their Web API services. As a result of this, we achieved to shorten the time dedicated to creating the catalog. However, two problems still remain; the data we could take does not fit perfectly to the one required, and the system still finds it difficult to be continuously managed in a long term.

Keywords: Web API, Scholarly information infrastructures, Python

1. はじめに

歴史学系の学会誌である『史学雑誌』には，毎年 1・5・9 号に西洋史学分野の文献目録が掲載されている．この文献目録は，当該年に出版された史学関係の論文・書籍を網羅的に掲載するもので，総掲載数は膨大なものとなる．大学院生は，全工程を含めれば一か月にも及ぶ作成作業を年４回行わなければならず，作業の部分的なデジタル化によって若干の効率化が見られるとはいえ，その負担は決して軽いものとは言えない．文献目録作成作業が学生の大きな負担となっている要因の 1 つとして，論文・書籍の検索，そして最終的な書誌情報の入力が未だに手動で行われている点があげられる. 時には 100 件以上にも及ぶ対象文献を，実際に図書館を訪れて確認し，1 件ごとにエクセルに打ち込まなければならないのである．

このような作業の時間を短縮し，学生の負担を軽減するための手段として我々は，学術情報基盤が提供する Web API を用いて文献情報を自動で検索し，『史学雑誌』において要求される形式に則って出力するプログラムの構築を試みた．これにより，文献の検索・確認をより容易にするとともに, 煩雑な入力作業を大幅に効率化することができた．

今回紹介するプログラムは，なんら技術的な革新性を有しているわけではなく，弊研究室の抱える問題に対処するための方法を示す個別具体的な事例紹介である．しかしながらこれは，既存の技術をいかに用いれば学生や研究者の研究環境を改善し，向上させることができるのかという実践的な問題意識に基づく試みであり，一般的にまだ情報学的知見の有用性が十分に認識されているとは言えない人文学分野において，分析や解釈といった研究内容に関わる技術に加え，書誌検索や目録作成のような日常の作業を効率化しうる技術活用を広く発信することに貢献するものである．

2. 従来型作業の工程と問題点

文献目録自動作成の紹介に入る前に本章では，手動で行っていた従来の目録作成の作業工程の概要，そしてその課題を述べておきたい．これを述べるのは，次章で述べる自

†1 東京大学大学院
The University of Tokyo
†2 東京大学大学院
The University of Tokyo

動化の意義を明らかにするための前提となる基礎情報であるがゆえであり，我々の抱く問題意識を共有してもらうために不可欠であろうと考えるがゆえである．

(1) 作業量・作業人数

史学雑誌西洋史文献目録の作成作業においては，対象となる雑誌は 699(オンライン版を含む)，調査対象の出版社は 46 あった．ここから時期的な変動もあるものの，関連があると判断されるものとして 400-600 本程度の論文，80-120 冊，多い時では 170 冊程度の書籍を抽出し，所定の形式で記載している．

作業の人員は，西洋史分野においてはこの分野の若手大学院生が担っており，そのため進学者の人数によって作業の人数に大幅な変動が見られる．例として，著者の代では 9 名進学したが，翌年は 5 名，作業人数としては実質 4 名のみであった．

これらの作業を，作業月の 1 日に開始号令をかけ，2 週間から 20 日程度かけて行っていた. 具体的な作業工程については次項で詳述するが，作業人数が多い時でも一人当たり 150-200 程度の雑誌あるいは 20 社以上の出版社の新刊情報のチェックが求められることから，当該作業について相当の時間をかけることとなり, 何よりも作業時間の短縮が，また作業人数に関わらず安定的に作業を継続する環境を構築することが課題となっていた．

(2) 作業工程

従来の作業工程においては，論文担当と書籍担当とで工程が大きく異なっていた．以下で，それぞれの作業工程について説明する．

論文担当

- OPAC から雑誌新刊情報を取得するプログラムを用いて，新刊が発行されている雑誌を確認
- 各雑誌が所蔵されている図書館（あるいはリポジトリ）に赴き，新刊中に収録されている論文を検索，確認（作業人数にも依るが，ほとんどの場合，一人が複数の図書館を担当）
- 図書館で現物を確認した後, 必要と思われる論文の書誌情報を記録し，それを記録用エクセルファイルに 1 件ずつ，手動で入力．入力に際しては厳密な形式が存在し, タイトルや人名の表記に関して詳細な規定があるのはもちろん，書誌タイプ（論文，研究ノート，書評など）によって入力形式が異なる．

書籍担当

- 確認すべき書店のリストに基づいて各書店ウェブサイトを確認し，新刊情報を収集する．
- 細かな規定に基づいて手動で入力を行う必要がある点は論文担当と同様

入力作業が終了した後，作業者全員が集って編集会議を行い，実際に採用する文献の選定，入力ミスなどの確認を行った上で，史学会に提出する．その後，史学会から送られる校正刷り原稿を作業者全員で校正して，一連の作業が終了する．

以上が作業工程の概要であるが，この一連の工程のうち，作業者の負担が大きいのは編集会議以前，すなわち文献調査の段階である．それゆえ次節では，特に文献調査の段階に焦点を当てて，その問題点を述べることとしたい．

(3) 問題点

さて，前節の説明からも明らかなように，文献調査作業は①文献の検索・確認と，②書誌情報の入力という 2 つの段階からなる．

まず，①の段階における問題として，現状では新刊が搬入されている雑誌名を把握することはできるものの，その中に含まれる論文の情報を一括で取得できるシステムが存在しない点があげられよう．むろん雑誌名がわかる以上，CiNii や NDL search, J-STAGE などの検索エンジンを用いて論文情報を検索することはできるが，これらの検索エンジンでは複数の雑誌にまたがる書誌情報を一括で検索・取得することができないなど，文献目録の作成作業に最適化されているとは必ずしも言えない．そのためにこれまでは，論文であれば雑誌名ごとに別個で収録論文を検索するか，実際に図書館に赴いて確認していた. また, 書籍であれば，対象を横断的に検索・書誌情報の取得が可能なプラットフォームが存在せず，出版社ごとに新刊を確認する必要があった．論文と比して，各図書館を回る必要がない点では負担が少ないと言えるが, 50 社近い出版社の web サイトを閲覧し新刊情報を収集するとともに，必要に応じて適宜現物を確認する作業は時間を要する．

続いて②の段階についてであるが，これまでの作業における大きな困難はまさにこの工程にあったと言える．すなわち，すでに述べたように，書誌情報を 1 件ずつ記録用エクセルファイルに手動で打ち込む必要があったのである．作業人数にも依るが，場合によっては 100 件を超える文献の一々を手動で入力するためには膨大な時間を要するとともに，入力ミスを犯す可能性も当然ながら高くなる．というのも,『史学雑誌』の目録作成においては，全角・半角の違いに至るまで厳密な形式が定められており，ミスを完全に防ぐことは非常に困難なのである．この部分を自動化することができれば，作業全体の時間を大幅に削減することはもちろん, 入力ミスを減らすことも可能になるであろう．

次章において述べる自動化プログラムは，以上のような問題点を踏まえ，これを改善することを目的として構築した．すなわち，複数の雑誌や出版社にまたがる書誌情報を

一括して検索，取得すると同時に，それを目録の形式に整形し出力することで，文献目録作成作業の大幅な効率化を図るものである．

3. Web API を用いた自動化の試み

本章では，Web API を用いた文献目録自動作成の具体的な内容について説明する．まず，文献情報を収集する際に頻繁に用いられるウェブ情報基盤と，それが提供する Web API について簡単に述べた上で，Web API を活用する意義を明らかにする．その上で，実際のプログラムについて述べ，それによって得られた成果と今後の展望についても言及することとしたい．

3.1 Web API を用いる意義

CiNii や NDL search といった情報基盤は，Web API を提供している[1]． API とは Application Programming Interface の略称だが，このことは，コンピュータ同士のやりとりであると同時に，その両端にいるサービス提供者と利用者の間のインターフェイスであるとも言える．利用者はこれを用いることで，自らのシステム開発の際に，他のサービスの提供するデータを有意義に活用することができる[2]．Web API は，このような API の機能を HTTP/HTTPS ベースで実現させるものである．これを用いれば，パラメータと値を指定することで特定の情報を URL の形でリクエストし，取得することができる．このパラメータの指定方法は API 提供者によって異なるが，リクエストの例をあげると，«http(s)://○○○/□□□?(パラメータ=値)& (パラメータ=値)&…&(パラメータ=値)»というような URL にアクセスすることで，パラメータの値に一致する情報が得られる．レスポンスの形式も JSON, XML, XHTML など複数ありうるが，これもパラメータを用いて指定することが可能である．

このような Web API の活用は，我々が目指す文献目録の作成に際していくつかの重要な利点をもたらす．まず検索に際しては，特定の条件を含んだ URL を自動的に作成するプログラムを組めば，条件の異なる複数の検索リクエストを順次作成し，書誌情報を大量に，一括で取得することができるようになる．そして取得した情報の表現に関しても，Web API は JSON や xml といった構造化されたデータ形式でレスポンスを返してくれるため，そのレスポンスを，Python 等を用いたプログラムで容易に操作・整形し，必要な形式に直して出力することが可能となる．

Web API を提供している情報基盤は多く存在するが，今回の試みにおいては CiNii API を主に用いた．その理由としては，無料で利用が可能である点，検索パラメータの種類が比較的豊富で詳細な検索が可能である点，そして，構造化データである JSON 形式でレスポンスを得られる点があげられる．もちろん，CiNii 以外の API を用いることも可能であるが，J-Stage は論文の発行団体の方針次第で最新号が非公開であり，最新の文献目録を作成するという目的にそぐわないケースが考えられることから採用を見送った．NDL search については，論文検索において CiNii articles を採用することと関係して，リクエストのパラメータやレスポンス形式などの仕様が異なり，異なるプログラムを組む必要があるため，今回は扱わないこととした．

ここまで，Web API を活用する意義について述べたので，次節では CiNii API を用いて実際に我々が組んだプログラムの紹介を行うこととする．

3.2 文献目録自動作成プログラム

作成にあたっては執筆者である小林・小川の共同体制をとった．導入に際しては主として市販の技術書を参考にした[3][4]．当初は Web スクレイピングによるデータ取得を試みたが，Tokyo Digital History で行った API 勉強会をふまえ CiNii API の導入を決定するなど協力を仰いだ．書籍部分を小林が，論文部分を小川が記述し，その際必要となった ISSN リスト等はそれぞれの雑誌を担当する院生に ISSN の調査を依頼した．

文献目録自動作成の第一段階として，まずは取得したい論文・書籍を含む雑誌名・出版社名，そして出版年（可能なら年月日）を指定し，それをもとに API のリクエスト URL を生成する必要がある．この作業自体は非常に簡単で，プログラム中に予めリクエスト URL のテンプレートを用意したうえで，変数の部分（パラメータの値）に必要な雑誌名（ISSN）・出版社名と出版年月日を代入していけばよい[5][6]．

さて，URL を用いてリクエストを行うと，CiNii は以下のような JSON ファイルを返す．


```
----------------------------------------------------------
"dc:publisher": [
{
"@value": "史学会 ; 1889-"
}
],
"prism:publicationName": [
{
"@value": "史学雑誌"
}
],
"prism:issn": "0018-2478",
"prism:volume": "128",
"prism:number": "2",
"prism:startingPage": "256",
"prism:endingPage": "238",
"prism:publicationDate": "2019-02",
```

"dc:date": "2019-02",

このファイルの中には個々の論文あるいは書籍のメタデータが収められており，プログラムは JSON パーサーを用いてその中からタイトル，著者，出版年月日，ページ数などの必要な情報を抽出し，正規表現を活用して表記形式を出力に適した形へと修正する．ここでいう「出力に適した形」とはむろん，史学雑誌文献目録の表記形式に則った形を意味する．

このように整形された書誌情報は, 最終的には TSV 形式に基づいてファイルに出力され，従来の方式における記録用エクセルファイルの書式に則った以下のような形で表示される．

"" "" 植村 利男 "" 学会消息 日本経済政策学会第 74 回全国大会(亜細亜大学大会) 亜細亜大学経濟學紀要 1 81-88 18-03

"" "" 布田 功治 "" 1990 年代半ばまでのシンガポール国際金融センターの発展過程 : 国際資本移動と国際金融センター化戦略 亜細亜大学経濟學紀要 1 47-69 18-03

"" "" 水野 明日香 "" 英領ビルマにおける 1941 年土地買い上げ法の制定 : 独立後の農地国有化法の起源 亜細亜大学経濟學紀要 1 15-46 18-03

"" "" 猪原 龍介 "" GIS を用いた NEG 分析 : 福島県域を事例として 亜細亜大学経濟學紀要 1 1-13 18-03

"" "" 宮下 郁男 "" 財政依存の北海道経済 旭川大学経済学部紀要 77 43-52 18-03

"" "" 須川 宏之 "" 旭川大学経済学部の英語教育に関する若干の考察 旭川大学経済学部紀要 77 31-42 18-03

"" "" 木谷 耕平 "" ロボット導入の地域経済への影響 : 先行研究レビュー 旭川大学経済学部紀要 77 21-29 18-03

このように，これまではすべて手動で行っていた検索・出力をプログラムに行わせ，上のような最終的な表示を行うことが可能になった．

以上が，文献目録自動作成プログラムの行う大まかな処理の流れである．技術的な手順を具体的に述べることは本稿の中心的な問題ではないので，細かなコードの詳解は控える．コードを参照されたい方は，GitHub にアップロードされたものを参照して頂きたい[7].

3.3 プログラム導入の結果

(1) 成果と課題

文献作成作業に我々のプログラムを導入した結果，作業時間が大幅に短縮された．これまで図書館に足を運んで個別に確認していた雑誌の多くが自動検索可能になるとともに, 入力に関してはほぼ完全に自動化することに成功した．これにより，自身の研究以外に担う負担の一部を軽減し，研究環境の向上の一助になったものと考えている．

具体的には昨年度の場合，対象雑誌・出版社は従来通りで，採用論文・書籍数も先述の例年の場合と大きく変化はなかった(論文 300-500，書籍 70-120)．一方で，作業人数は 4 名(初回のみ 5 名)と半減しており, 一人当たりの作業数は単純に 2 倍になったと考えられるが，作業期間は一昨年度(9 名で作業)と同じかやや短い程度(2-3 週間)になっている．

しかし，依然として課題もある．一つは，前節で示した完成形ファイルの中に""で示された空欄が見られるように, 自動では取得不可能な情報，すなわち手動入力に依らざるをえない情報が存在する点である．これは論文の扱う時代の区分，著者の読み仮名がこのような情報に該当するが，これに関しては, 現状では CiNii API のみを用いているが，将来的に NDL search や J-STAGE の API を併用することで改善できる可能性があると考えている．各情報基盤が有する情報は少しずつ異なるため，我々の必要に応じて，複数の情報基盤を横断的に検索し，情報を取得できるプログラムを組むことで，目的に最適化された出力を得ることができる．

他の課題として，人員の変更に伴う担当分野の変遷に対してプログラムが柔軟に対応できていないことが挙げられる．現状では，取得データの総量が大きいこともあり，総責任者が担当者の作業範囲を手動で設定し，それぞれの範囲ごとにプログラムを実行することとしている. そのため，引き継ぎの際の担当範囲の再設定や，各作業の開始時に担当人数分プログラムを実行する必要があるなど，余計な手間が発生している．これに対しては，Web アプリなど GUI で実行できる環境を構築し，各々が自らの担当範囲を選択・入力しデータ取得を行うことでより効率的な作業を行うことができると考えられる．

(2) 今後の展望と取り組み

上述のように，現状では担当分ごとに別個でプログラム

を動かす必要があり，さらにローカル環境で実行するため，作業者は Python が作動する環境を整える必要がある．しかし我々の最終的な目的は，作業に関わる誰もが容易に利用することのできる文献目録自動作成プログラムを構築することであり，このためにはサーバー上で実行可能な Web アプリ化を進める必要がある．これにより，作業者は必ずしもプログラミング・リテラシーを有さなくとも自動作業を実行することが可能になり，文献目録自動作成プログラムは十分な持続性を持つシステムとなるはずだ．

Web アプリ化については現在，Django を用いてすでに試作を進めており，以下のような入力システムは完成している．

この画面から出版年と雑誌名を選択すると以下のように ISSN が表示され，これをもとに検索を行うことを予定している．

出力システムに関してはまだ構築していないが，このシステムが完成した暁には，上の入力画面から必要な情報を選択し入力することで，文献目録の形に自動生成された TSV ファイルを取得できるようになる．

一方で，Web アプリを作成したとしても，これを維持していく上で，プログラミングを理解できる人材を育成していく必要がある．そこで，著者たちの参加する，大学院生を中心としたワークショップ「Tokyo Digital History」を通じて，西洋史学研究室の学生を含む大学院生に，本プログラムを題材としてプログラミング勉強会を開催した．人文学の研究でもデジタル技術を扱うようになりつつある中，個々人の研究へ応用可能な技術を紹介しつつ，文献目録作成作業を運用可能な環境を構築するための活動を今後も行っていく．

4. おわりに

ここまで，我々がすでに構築し，実際の作業にも利用したプログラムについて紹介し，今後の取り組みとしての web アプリ化や人材育成の計画について述べた．

技術的には未だ多くの問題を抱えているシステムではあるが，研究以外の業務の効率化を通じた研究環境の改善に寄与するものであることは間違いない．人文情報学の分野において，分析や解釈といった研究手法の一環としてデジタル技術を活用することはもちろん有意義であるが，同時に，学術情報基盤の充実に伴って，この技術を研究を補助する作業にも応用することで，より研究にフォーカスできる環境を構築することもまた可能である．

一般にデジタル技術活用に関する理解が十分に浸透しているとは言えない人文学分野においては，直接に研究への応用可能性を示すだけでなく，今回の文献目録作成のように，研究を補助するような位置付けでそれを導入するメリットや可能性をより身近に感じてもらうことによっても，デジタル技術，ひいては人文情報学の意義を広め，より活発な活動を促すことに繋がると期待している．

謝辞 東京大学大学院人文社会系研究科人文情報学拠点拠点における人文情報学概論の授業において下田正弘先生をはじめとする諸先生方にご指導をいただいたことを感謝とともに記しておく．

また，Tokyo Digital History のメンバーには勉強会等を通じて本プロジェクトへのコメントやアドバイスをいただいた．試作期間には西洋史学研究室の同期一同に作業の一部を依頼するなど，様々な形で協力いただいたほか，蔡男氏には運用時の状況やフィードバックを賜った．記してお礼を申し上げる．

参考文献

[1] 大向一輝. CiNii のウェブ API 戦略. 情報の科学と技術, vol, 64, no.5, p.170-174. doi: 10.18919/jkg.64.5_170

[2] 水野貴明. Web API: The Good Parts. オライリー・ジャパン, 2014

[3] Sweigart, A. 相川愛三訳. 退屈なことは Python にやらせよう. オライリー・ジャパン, 2017.

[4] Lubanovic, B. 斎藤康毅監訳 長尾 高弘訳. 入門 Python3. オライリー・ジャパン, 2015.

[5] “CiNii Articles - メタデータ・API - CiNii Articles 論文検索の OpenSearch”.
https://support.nii.ac.jp/ja/cia/api/a_opensearch (参照 2018 年 3 月)

[6] “CiNii Books - メタデータ・API - CiNii Books 図書・雑誌検索の OpenSearch”.
https://support.nii.ac.jp/ja/cib/api/b_opensearch (参照 2018 年 3 月)

[7] https://github.com/ToDH-young/shigakuzasshi