

学術研究のためのオープンソース・ソフトウェア(2)BiBTEXとZotero

MIYAZAKI, Kenji / 宮崎, 憲治

(出版者 / Publisher)

法政大学経済学部学会

(雑誌名 / Journal or Publication Title)

経済志林 / The Hosei University Economic Review

(巻 / Volume)

83

(号 / Number)

2

(開始ページ / Start Page)

119

(終了ページ / End Page)

149

(発行年 / Year)

2015-11-30

(URL)

<https://doi.org/10.15002/00012444>

【研究ノート】

学術研究のためのオープンソース・ソフトウェア (2) : BiBTeXとZotero

宮 崎 憲 治

1 はじめに

宮崎 (2015) では学術研究のためのオープンソース・ソフトウェアとして、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (以降 latex), とくに $\text{X}_{\text{L}}^{\text{L}}\text{A}_{\text{E}}\text{X}$ (以降 xelatex) を紹介した。 latex を使うメリットの一つは、図や表や式番号にラベルをつけておきそれを参照するようにすれば、順番が入れ替わったりしても正しく処理してくれることである。さらに、参考文献についても文献リストに挙げたものを適切に参照してくれることもメリットの一つである。

具体的に参考文献に引用されているものを引用するためには、まず次のように文献リストを入力する。

```
\begin{thebibliography}{9}
```

```
\bibitem{aaa} 齊藤誠『父が息子に語るマクロ経済学』勁草書房，東京，2014.
```

```
.....
```

```
\end{thebibliography}
```

ここで `{aaa}` 内の `aaa` は `bib` キーと呼ばれる^{*1}。

^{*1} 他の呼び方として `bibtex` キーや `cite` キーなどがある。

引用のさい「齊藤\cite{aaa}」とする。そうすると作成した文献リストに対応する bib キーがあれば、対応する番号が挿入される。なお著者の名前も必要である。これにより参考文献との対応がある程度改善する。しかし完全でない。標準に搭載している参考文献は番号型式での引用しか対応していない。また参考文献リストに挙げたものをすべて使わなくても、エラーがなく本文に引用していない参考文献が残ってしまう。あと手入力で文献リストを作るため、参考文献によって書式が微妙に異なったりする可能性がある。

経済学など社会科学での引用は「著者（年）」が標準的である。そのため latex で参考文献を作成する際に、リストだけは作っておくが、後に手入力で引用している人が多い。経済学のワーキングペーパーで引用に番号がふられ、本文に著者（年）となっているのはほとんどそうであろう。わたしも最近までそうしてきた。しかし、論文がアクセプトされ、latex のコードを提出後に、いつも校正者に指摘されるのは、本文で利用されている引用の年がずれていたり、引用されていない論文が参考文献リストに存在しているといったことであったため、一念発起して BiB_TE_X（以下 bibtex）に切り替えた。

bibtex とは latex と組み合わせて文献データベースから自動的に参考文献リストを作成するパッケージで、Oren Patashnik と Leslie Lamport によって1985年に作られた^{*2}。その後国際化がすすみ、日本語対応の bibtex について jbibtex が開発された。現在は pbbibtex と呼ばれている。他にも内部をユニコード対応した upbbibtex がある。いずれにせよ (p/up) bibtex は古いパッケージのため、なかなかカスタマイズが難しい。一方 biblatex という比較的カスタマイズが簡単にできるパッケージが勢力を伸ばしはじめている。今後こちらのほうが主流になるかもしれないが、現状では日本語文献と外国語文献を別々に分けてくれるようなスタイルファイルは存在しないはずである^{*3}。

^{*2} Patashnik（1988）は作者自身によるマニュアルである。

私自身 bibtex の存在は昔から知っていたけれど、いまいち使う気になれなかった。理由は bibtex ファイルを作成するための bib ファイル作成が億劫だったからである。しかしながら、インターネットの発展とともに、ネットから書誌情報を収集することが主流になるとともに、文献管理ソフトが充実してきて状況が一変した。文献管理ソフトとして Zotero を使っているが、これで文献を管理し、それを bib ファイル形式に出力して、過不足ない参考文献つき論文が非常に短時間で作成できるようになり、文献整理にかかる時間は激減した。

文献整理時間が激減した割に論文作成の生産性は上がっていないのはいささか恥しいかぎりであるが、以下 bibtex の使い方と Zotero の使い方を示しながら、私がどのように文献整理しているのかを紹介したい。

2 bibtex による参考文献作成

ここでは bibtex でどのように参考文献を作成するかを説明する。はじめに bib ファイルがどのような形式か説明し、bibtex をどう使えばよいかを説明する。そして社会科学でよく使われる「著者 (年)」ためのパッケージ natbib を説明し、これで日本語文献を扱うための jecon.bst を説明する。最後にいくつかの注意をのべる。

2.1 bibtexを使う場合

bibtex を使うためには、本文が書かれた tex ファイルとは別に bib ファイルというデータベースファイルを用意する必要がある。

2.1.1 bib ファイル

bib ファイルは、例えば文献ごとに

*3 <http://shirotakeda.org/blog-ja/?p=2660>

```

@article {mye:1981,
  author   = {Roger B. Myerson},
  title    = {Optimal {Auction} {Design}},
  journal  = {Mathematics of Operations Research},
  year     = {1981},
  volume   = {6},
  number   = {1},
  month    = {feb},
  pages    = {58--73}
}

```

という書式という書式によって書かれている。文献の種類が指定され、中括弧でいくつかの項目があるが、一番最初の行にある `mye:1981` は bib キーと呼ばれる。それ以外はフィールドと呼ばれる。

文献の種類について、論文の場合は `@article` ではじめる。書籍の場合 `@book` である。本の一部の論文を引用する場合 `@incollection` である。学会の予稿集なら `@proceedings` で、出版されていない論文は `@unpublished` である。大学や研究機関で出しているワーキングペーパーなら `@unpublished` の `note` 欄にワーキングペーパー名等を書き込めばよい。もしくは `@techreport` とすればよい。ただこのままだと“Technical Report”と出力されるので、これ変更したければ `type` 欄に入力すればよい。

他にも引用できる論文の形式があり、必要項目や任意項目にわけられる。まとめると表1のようになる。

気をつけることをいくつか述べる。まず論文のタイトル・フィールドで最初の単語以外で大文字にしたいときは、`{...}` を挿入する。またページ1の範囲は `{58-73}` でなく `{58--73}` とマイナスを2つ入れる。なおそれぞれのフィールドをダブルクォーテーションで括ってもよい。

他にも英語文献のとき著者名は

表 1 bibキー

文献種類	bibtex	必要項目	任意項目
学術論文	@article	author, title, journal, year	volume, number, pages, month, note
書籍	@book	author/editor, title, publisher, year	volume, number, series, address, edition, month, note
書籍の一部	@incollection	author, title, booktitle, publisher, year	editor, volume, number, series, type, chapter, pages, address, edition, month, note
未発表論文	@unpublished	author, title, note	month, year
技術報告書	@techreport	author, title, institution, year	type, number, address, month, note
その他	@misc	なし	author, title, howpublished, month, year, note

```
author = {Doe, Jhon}
author = {John Doe}
```

のどちらかで指定するのに対し、日本人名の場合は

```
author = {太郎, 山田}
author = {山田 太郎}
```

であることに注意されたい。

さらに日本語の文献に `yomi` というフィールドを付け加える必要がある。このフィールドにアルファベットかひらがなで筆者の姓のみを加えればよい。複数筆者でも順序を決めるだけなので第一著者だけの姓を入れる。アルファベットの姓の場合、英語文献と混在した参考文献が作られる。ひらがな入力の場合、欧文の著者リストがまず表示され、次に日本語文献があるうえお順に表示される。

2.1.2 bibtex の使いかた

bib ファイル（ここでは `bibtest.bib`）を準備したうえで `tex` ファイルを次のように編集する。

- 1. `\begin{document}` の後、参考文献リストの前に `\bibliographystyle{plain}` を指定する。

2. 引用したい箇所に `\cite{bibキー}` を入力する。
 3. 参考文献リストを置きたい場所に、`\bibliography{bibtest}` を入れる。最初の指定はスタイルファイルの指定である。日本語を扱う場合 `jplain.bst` が一番基本的である。拡張子をつける必要がない。
- そして次の手順で実行する。

1. まず latex エンジンで一回コンパイル。(aux ファイルを作成)
2. bibtex エンジンを実行。(bbl ファイルを作成)
3. latex エンジンを実行。(bbl ファイルを取り込む)
4. 再度 latex エンジンを実行。(相互参照を解決する)

latex エンジンとして pdflatex, platex, uplatex, xelatex, lualatex などを用いる。内部がユニコード化した uplatex, xelatex, lualatex を用いれば、第二水準を越える漢字を含む日本語を追加パッケージなしで扱える。わたしはメインに xelatex を使っている。

bibtex エンジンとして、多くの日本人は pbibtex を用いている。しかしながら内部がユニコード化した upbibtex を用いれば、第二水準を越える漢字を含む日本語が扱えるので、upbibtex を使用することをお勧めする。

なおコンパイルを何度も実行して面倒にみえるが、宮崎 (2015) で紹介した latexmk (<http://users.phys.psu.edu/~collins/software/latexmk-jcc/>) をつかえば適切な回数のコンパイルを一発で実行してくれる。

簡単に例を示すと、次のソースコードを実行すればよい。

```
\documentclass{bxjsarticle}
\usepackage{xltextra}
\usepackage{zxjatype}
\usepackage[ipa]{zxjafont}
\begin{document}
Deaton \cite{deaton_understanding_1992},
Hayashi \cite{hayashi_1990s_2002},and
```

```
齊藤\cite{hayashi_1990s_2002,saito_2014}
```

```
\bibliographystyle{jplain}
```

```
\bibliography{bibtest}
```

```
\end{document}
```

ただこのままだと、「著者（年）」形式の引用ができない。スタイルファイル `jplain.bst` を用いた場合、著者 [番号] である。

2.2 natbib

著者（年）形式にするには `natbib` というパッケージをもちいればよい。

1. `\begin{document}` の前のプリアンブル部に `\usepackage{natbib}` を挿入する。
2. `\begin{document}` の後、参考文献リストの前に `\bibliographystyle{apa}` を指定する。
3. 参考文献リストを置きたい場所に、`\bibliography{bibtest}` を入れる。

引用したい箇所に `\citet` もしくは `citep` を入れればよい。前者だと Deaton (1992) と名前（年）と表示できる。後者だと (Deaton, 1992) と括弧つきで（名前, 年）と出力できる。自分で名前を入力する必要はない。複数の著者も対応可能である。

たとえば次のソースコードを実行すればよい。

```
\documentclass{bxjsarticle}
```

```
\usepackage{xltextra}
```

```
\usepackage{zxjatype}
```

```
\usepackage[ipa]{zxjafont}
```

```
\usepackage{natbib}
```

```
\begin{document}
```

```

\citet{deaton_understanding_1992},
\citep{deaton_understanding_1992},
\citet{hayashi_1990s_2002,deaton_understanding_1992}と
\citep{hayashi_1990s_2002,deaton_understanding_1992}
\bibliographystyle{apa}
\bibliography{bibtest}
\end{document}

```

スタイルファイル **apa.bst** は日本語に対応していないので、日本語文献を引用しても上手く表記されない。

2.3 jecon.bst

奥村・黒木（2013）によると **natbib** を用いた日本語が扱えるスタイルファイルがないと述べているがそれは誤りである。京都産業大学武田史郎教授のスタイルファイル **jecon.bst** を用いれば、日本語でも著者（年）形式にすることができる。このスタイルファイルは

<http://shirotaekeda.org/ja/tex-ja/jecon-ja.html>

からダウンロードできる。付属のマニュアルがよく書けているのでそれを読めば使い方がわかるだろう。

手順は次のようにする。

1. `\begin{document}` の前のプリアンブル部に `\usepackage{natbib}` を挿入する。
2. `\begin{document}` の後、参考文献リストの前に `\bibliographystyle{jecon}` を指定する。
3. 参考文献リストを置きたい場所に、`\bibliography{bibtest}` などを入れる。

たとえば次のソースコードを実行すればよい。

```
\documentclass{bxjsarticle}
\usepackage{xltextra}
\usepackage{zxjatype}
\usepackage[ipa]{zxjafont}
\usepackage{hyperref}
\usepackage{natbib}
\begin{document}
This is a test. \citet{deaton_understanding_1992}and\itemp
{ogawa_1986},
\citet{hayashi_1990s_2002,saito_2014}と
\citep{hayashi_1990s_2002,saito_2014}
\bibliographystyle{jecon}
\bibliography{bibtest}
\end{document}
```

結果は図 1 の通りである。

This is a test. Deaton (1992) and (小川, 1986), Hayashi and Prescott (2002); 齊藤 (2014) と (Hayashi and Prescott, 2002; 齊藤, 2014)

参考文献

Deaton, Angus (1992) *Understanding Consumption*: Oxford University Press.

Hayashi, Fumio and Edward C Prescott (2002) “The 1990s in Japan: A Lost Decade,” *Review of Economic Dynamics*, Vol. 5, No. 1, pp. 206–235, January, DOI: <http://dx.doi.org/10.1006/redy.2001.0149>.

小川一夫 (1986) 「日本における恒常所得仮説の検証: 消費と労働供給の同時決定モデル」, 『国民経済雑誌』, 第 154 巻, 第 3 号, 63–85 頁.

齊藤誠 (2014) 『父が息子に語るマクロ経済学』, 勁草書房, 東京.

図 1 jecon.bstおよびupbibtex

私は、このスタイルファイルをプロジェクトのフォルダごとに入れていく。他の研究者にこのファイルをもっていないと上手くコンパイルしてく

れないからである。もし自分だけ使うのなら、デフォルトだと

`~/texmf/bibtex/bst`

に `jecon.bst` を配置しておけばよい。

著者（年）形式に対応できるだけでなく、日本語の書籍で翻訳者を含めることができる。そのときフィールドは `jauthor` である。また翻訳書の原作者名は `{T.ピケティ}` というように、名前の半角アルファベットとカタカナまじりにする。ここで詳述しないが英語の本に日本語の書誌情報も加えることもできる。

ところで最近の翻訳書で原作者の名前がアルファベットのままの本もある^{*4}。このままだと洋書と同じところに順番に表記される。もし和書と洋書の参考文献リストを区別したいのなら、カタカナ名に変更するか、`yomi` フィールドにカタカナを入力する。

さらに `jecon.bst` は他のフィールドとして `doi` ナンバーや `url` フィールドやアクセス日時 `access` フィールドもあり、最近のインターネットからの引用にも対応している。もし `bib` ファイル内で `url` アドレスがあるのなら、`\usepackage{url}` もしくはそれを含めている `\usepackage{hyperref}` をプリアンプル部につける必要がある。

2.4 その他bibtexについて

`bib` ファイルの編集は、テキストエディタを用いればよい。例えば次のようなものがある。

- BibDesk (<http://sourceforge.net/projects/bibdesk/>)
- Texmaker (<http://www.xmlmath.net/texmaker/>)
- Emacs (<https://www.gnu.org/software/emacs/>)
- JabRef (<http://jabref.sourceforge.net/>)

^{*4} たとえばオライリー・ジャパン社 (<http://www.oreilly.co.jp/index.shtml>) の本がそうである。

BibDesk は Mac 専用で, MacTex をデフォルトで導入すると自動的にインストールされる。基本的な bib ファイルを編集出来るだけでなく, citeulike などのウェブサービスの文献リストから一括で書誌情報をとり込む機能がある。わたしは以前, citeulike を利用していたので, この取り込み機能が非常に役立った。

Texmaker については前回に宮崎 (2015) で紹介した。tex のソースの編集と同じエディタであつかえるが, だからといって, bib ファイルを読み取り, 引用の際に補完してくれるわけではない。そのような機能があるエディタの一つは Emacs であり, このエディタをわたしは偏愛しているけれど, 古いエディタで独特のキーバインディングのため万人に勧めにくい。以下, 簡単に JabRef について紹介する^{*5}。

JabRef は最も有名な bib ファイル文献管理ソフトウェアである。実行には Java が必要である。入手は公式ホームページからダウンロードすればよい。設定であるが, 最初にダウンロードしたときは英語のままなので次の手順で変更する。「Option→Preference→General」より [Language] を「Japanese」にし, [DefaultEncoding] を「UTF8」にして再起動する。おそらくこれで大丈夫であるが, もし上手く表示されないようだ次のことを試せばよい。

- 「オプション→設定→外観」の [表フォント設定] ボタンで日本語を表示できるフォントを選ぶ。
- 「オプション→設定→項目プレビュー」`<font face="arial">`の `arial` の部分を日本語を表示できるフォントに変更する。

主な使い方について, 公式ホームページにあるドキュメントの他に

http://keijisaito.info/arc/biblio/jabref_jbib.htm

が参考になるかもしれない。bibtex 形式での書誌情報が出力しているウェブサイトからのインポート方法が記述されている。ただ私は次節に説明す

^{*5} たとえば bibtex-mode と refTeX-mode である。

るように Zotero を使っているのであまり使わない。

Zotero だと英語文献は何の問題なく、bib ファイルに出力してくれる。日本語文献だと若干の修正が必要な場合 JabRef を利用している。日本語文献を扱うために yomi というフィールドを追加して入力すればよい。

JabRef は、Zotero と同じように、bib キーを自動的に設定する機能がある。Zotero では日本語文献についてはキーをうまく作成しないのだが、JabRef では作成可能である。また作成ルールは「オプション→設定→BibTeX 鍵の生成」で変更も可能である。

さらに、JabRef は、ファイル名をbibキーに合わせて変更したり、ファイルの保存先をクラウドストレージサービスのフォルダ（例えば Dropbox）に変更できる。Zotero の共有機能を使わない場合でも簡単にファイル共有ができ、タブレット上で Dropbox といったアプリから直接利用が可能なるようである。詳しくは以下のサイトを参照されたい。

<http://ur.edu-connect.net/archives/2267>

2.4.1 参考文献リスト

授業でリーディングリストを配布する機会がある。その場合 bib ファイルを用意し、`\nocite{*}` をつけた次のファイルを作成すればよい。

```
\documentclass{bxjsarticle}
\usepackage{xltextra}
\usepackage{zxjatype}
\usepackage[ipa]{zxjafont}
\usepackage{natbib}
\usepackage{hyperref}
\begin{document}
\bibliographystyle{jecon}
```

```
\nocite{*}
\bibliography{bibtest}
\end{document}
```

これをテンプレートファイルとして作っておけば `\bibliography{bibtest}` を変更するだけで文献リストのソースコードができる。

2.4.2 beamer

プレゼンの際、最後に参考文献を挙げないことが許されているようであるが、本来なら掲載するほうが、聴衆に親切であり、望ましい。bibtex をもちいれば、年号や名前の不一致もなく作成することができる。

beamer プレゼンに参考文献をつける際、次のことに気をつけよう。文献リストは多くの文献があるため 1 ページに収まることが少ない。それを解消するため、字を小さくするか、`allowframebreaks` のオプションをつけて分割表示すればよい。次のようにフレームを作成する。

```
\begin[t,allowframebreaks]{Reference}
\beamerdefaultoverlayspecification{
\footnotesize
\bibliographystyle{jecon}
\bibliography{bibtest}
\end{frame}
```

なおフレームオプション `allowframebreaks` はスライドアニメーションと相性が悪いので、プリアンブル部に `\beamerdefaultoverlayspecification{<+>}` と指定した場合、`\beamerdefaultoverlayspecification{}` をフレーム環境内に設定しておかないと上手く表示されない^{*6}。あとフレームオプションに `label` があると上手くいかない^{*7}。

2.4.3 bibtexをつかえない研究者との共同研究

latex を使っても, bibtex を使わないひとは多い。私のように bib ファイルの作成が面倒という他に, 最近まで日本文献の引用がきちんと出来なかったことが理由の1つだろう。また同僚によると共著を書くときに相手が使わないからだと言っている。

bibtex を知らない研究者とは共著をしない, もしくは bibtex を使えと指導することができるのなら問題解決である。しかしながら, 若い研究者が bibtex を使えない年上の研究者と共同研究するとき, 使用を強制するのは立場上難しいだろう。

その場合, 年下のほうが文献管理し, 年上にファイルを渡すさいに, 次のようにするとよい。

1. tex ファイル内の `\bibliographystyle`, `\bibliography` 命令を両方とも消す。
2. bbl ファイル内の文献箇所 (`\begin{thebibliography}{9}...`
`\end{thebibliography}`) をコピーし, `\bibliography` 命令のあった部分に貼り付ける。

スクリプトファイルを作れば一括でできるだろう。また同様の手順を論文がアクセプトされて編集者にソースコードを送る際におこなうとよい。

latex どころでなく Microsoft Word しか使えない研究者との共著する場合もあるだろう。また日本語の編集本で doc/docx ファイルで論文を提出しなければいけないこともあるかもしれない。そのために `tex2word` (<http://www.chikrii.com/products/tex2word/>) を私は使っている。これは Windows版Word のアドオンで商用ソフトウェアである。日本語に変換するには, 日本語コードを utf8 からシフト JIS に `nkf` などを使って変

*6 <http://tex.stackexchange.com/questions/41909/beamer-weird-bug-in-bibliography>

*7 <http://tex.stackexchange.com/questions/237340/label-in-frame-options-disable-allowframebreaks-in-beamer>

換しなければならない。また bibtex には対応していないので、上述のように bbl ファイルの中身を挿入する必要もある。

3 Zotero による論文収集

3.1 論文収集

前節では bibtex を利用して、論文内での引用方法などを紹介した。ここでは bib ファイルの存在を前提に話をすすめてきた。どのように参考文献を集めて管理するかも研究活動のうえで重要なことである。文献整理に時間を費して本来の研究時間を削ってしまっは本末転倒である。直接 bib ファイルを JabRef で管理している知人がいるが、文献管理専用のソフトウェアを用いたほうが遥かに能率的である。文献管理のためのソフトウェアやサイトは次のものがある。

- EndNote (<http://endnote.com>)
- RefWorks (<https://www.refworks.com/jp/>)
- citeulike (<http://www.citeulike.org>)
- Mendeley (<https://www.mendeley.com>)
- Zotero (<http://www.zotero.org/>)

これらのソフトの機能について、いくつかのインターネットサイトで比較されている。

<http://umb.libguides.com/managecitations>
[http://www.library.wisc.edu/services/citation-managers/
comparison-chart/](http://www.library.wisc.edu/services/citation-managers/comparison-chart/)

私は以前は citeulike に登録して文献を収集していたが、現在は Zotero を利用している。EndNote や RefWorks は商用ソフトなので論外だとし

て、フリーで利用できる Mendeley の利用も候補にあがったが、オープンソースでないためソースコードを自分で修正できず、2010年ごろから Zotero を利用することにした。Zotero はブラウザに統合されており、論文収集だけでなく、インターネットサイトも取り込める。Microsoft OneNote や Evernote のようなクラウドメモのようにつかえることも利用の決め手となった。以下どのように私が Zotero を利用しているかに述べていく。

3.2 Zotero とは

Zotero はジョージ・メイソン大学が開発している、オープンソースの引用管理ソフトウェアである。最初は、Firefox の拡張機能として動作していたが、Zotero3.0 からはブラウザと独立して機能するスタンドアローン版も登場している。Zotero の名称は、アルバニア語の「極める、使いこなす」という意味の動詞から由来している。図 2 が Zotero のスナップショットである。

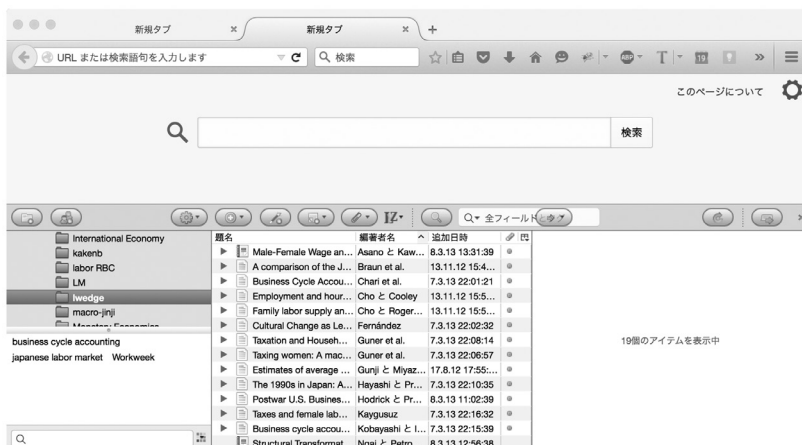


図 2 Zotero

ブラウザで気になったサイトや書誌情報を効率的に利用することができる。Firefox のアドオンであるが、アドオンのサイト (<https://addons.mozilla.org/ja/firefox/>) からインストールできない。インストールは公式サイトからダウンロードする必要がある。ダウンロードしたファイルをダブルクリックを実行すればよい。また上記のサイトにはスタンドアローン版もあり、Firefox だけでなく、Chrome や Safari から利用可能である。

私のメインのブラウザは、インターネット上でブラウズしながら Zotero で気軽に情報登録できるので Firefox である。しかしながらバージョンアップによっては動作が不安定なときがあり、その際には Chrome も利用している。ただこれら2つのブラウザはメモリだけでなく、エネルギー消費もいちじるしいので Mac を使っているときは Safari も利用している。どれを用いても取り込んだデータは Zotero で一元管理される。

3.3 使い方

Zotero の使い方は公式ホームページ

<https://www.zotero.org/support/ja/start>

にある。ただ一部が翻訳されていない。またショートカットキーは

https://www.zotero.org/support/preferences/shortcut_keys

にある。他にも <http://zotero-manual.github.io/zotero-manual/> を参照されたい。

使い方の基本は、興味深いインターネットサイトがあれば、Zotero でワンクリックで取り込むことである。インターネットサイトについて、ブックマークと違ってスナップショットを自動的に取得するので、将来サイトが閉鎖されていても記録されている。記録されたものに対して、メモやタグをつけることが可能である。このように入手したものをアイテムという。

アイテムは手動でも入力可能である。

これらのアイテムはコレクションごとに整理される。コレクションはフォルダやディレクトリのようなものである。通常のディレクトリとファイルの関係と違って、アイテムを複数のコレクションに所属させることができる。またアイテムの重複がないかを確認できる。私は研究プロジェクトごとにコレクションを作っている。

インターネットサイト以外の代表的なアイテムは書誌情報である。書籍については、ISBN がわかっているなら、Zotero から直接 ISBN を入力すれば良い。ただ、日本語の書籍だとローマ字の書誌情報になることもある。ISBN がわかっていなくても Google Books や Amazon などに存在する本なら、そのサイトで書誌情報を得ることができる。

学術論文について Google Scholar (<https://scholar.google.co.jp>) に対応しており、公表論文も未公表論文も調べられる。また英語論文だけでなく日本語論文も入手可能である。公表論文についてdoiがあるならそれを入力すればよい。JSTOR や ElsevierのScience Direct など、論文のPDFがダウンロードできるサイトから、自動的に書誌情報を得ることができる。得た書誌情報に対して、ダウンロードした pdf ファイルを添付資料として貼り付けることができる。さらに pdf のメタ情報を利用して、ファイル名を自動的に変更してくれる。

日本語の学術論文について CiNii (<http://ci.nii.ac.jp/>) から書誌情報を入手することができる。ただ情報情報がずれていることがあるので修正が必要である。例えば、発行年月が「2009年2月」のように日本語表記のままなので、それを「2009-2」のように半角数字のみ変える必要がある。

さらに、ワーキングペーパーも簡単にとることができる。経済学者は *IDEAS: Economics and Finance Research* (<http://ideas.repec.org/>) を利用することが多いが、それにも Zotero は対応しており、ワンクリックで書誌情報を取り込むことができる。特筆することは、ダウンロードに制限

がかかっていなければ自動的に pdf ファイルもダウンロードして添付ファイルとして登録してくれる。

なお、書誌情報の収集の際、インターネットの情報に間違いがあることに注意しなければいけない。先ほどの IDEAS で集めた情報でしばしば綴りやページ番号などが間違っていることがしばしばある。日本人研究者の場合、姓と名が逆になっていることもある。なるべく doi 番号から書誌情報を収集し、それ以外から収集する場合、間違いがないかよく確認したほうがよい。

こうして各種取り込んだ文献を整理して、フォルダごとに参考文献リストとして出力することができる。latex で利用できるように bib ファイル形式にも出力することができる。bib ファイル形式で出力する際に、それぞれの書誌情報を識別するエン트리キーを、著者や出版年数などから自動的に作成もしてくれる。ただし、漢字やかなまじりの日本語文献については、うまくエン트리キーを作成してくれず、あとで JabRef などの bib ファイル編集支援ソフトを利用する必要がある。

他にも html 形式や docx 形式でも出力可能である。bibtex を使わずに、授業で学生に配布する文献リストを非常に簡単に作成することができる。英語文献の場合、何ら問題がないが、日本語が含まれる場合、複数著者が and で繋がれていたりして、不格好なので修正する必要がある。

3.3.1 ライブラリの共有

さらに Zotero にアカウント登録すれば、書誌情報を同期することができ、複数のパソコンで自分のライブラリーを利用可能である。さらに他のアカウントをもつユーザーとグループ・ライブラリを共有することも可能で、共同作業の際に非常に役立つ。

これらは書誌情報のアイテムだけが同期しており、pdf の添付ファイルが同期されない。同期するためには、Zotero ファイル・ストレージの利用するか、自前で WebDAV を用意しなければならない。Zotero ファイル・

ストレージは一人100メガまで無料で用意されているが、それ以上は有料である。WebDAV を利用するには、自分でサーバーを立ち上げることもできるが、各種の WebDAV に対応しているストレージサービスを利用すればよい。以下にいくつかのサイトがリストアップされている。

https://www.zotero.org/support/kb/webdav_services

私は、Box.net を利用している。クラウドで最も有名な Dropbox は WebDAV に対応していない。執筆時点で、Box.net は、無料で10G提供してくれるので、論文ひとつが2Mであっても5000本の登録可能である。ただ WebDAV だとグループ・ライブラリの中のファイルの同期がサポートされないようである。

なお Zotero 4.0 から、相対パスでのリンクにも対応したようである。それぞれのパソコンに保存先のリンクを Dropbox に指定しておけば、ライブラリだけでなく、添付ファイルもアカウントをつくることなく共有できるようである。ただ私自身は試していない。

3.3.2 アドオン

スタンドアローン版でなく、Firefox のアドオンをつかっているなら、Zotero の機能を拡張してくれるアドオンをもちいることができる。例えば、Zotfile, Zutilo Utility for Zotero, Google Scholar Citation for Zotero, LyZ などがある。最初の Zotfile はスタンドアローンでもインストール可能である。

Zotfile (<http://zotfile.com/>)は、添付した pdf ファイルの名前変更をしてくれたり、pdf ファイルを tablet と同期する機能がある。Zotero の昔のバージョンだとファイル名の機能がなかったので重用したが、最近のバージョンだと名前を変更してくれるので、このアドオンをあまり用いていない。ただ Zotero の名前変更だと変に日本語に対応して、複数著者の場合、「and」でなく「と」となるので、それがみっともないと考えるならこ

れを用いる価値がある。

また tablet の同期についても Dropbox などのクラウドフォルダに出力させて、それぞれの tablet で閲覧する Android でも iPad でも後述するように Zotero と直接連動できるファイルが登場している。ただ、注釈をいれたものを取りこんでくれるので、pdf に直接書き込むことが多いなら重宝するかもしれない。

他のアドオンとして Zutilo Utility for Zotero はキーボードショートカットをカスタマイズしてくれる。ただ私はあまり使っていない。Google Scholar Citation は Google Scholar が計算した引用数も入力してくれる。ただそれが note フィールドに入れられるので、私の使用環境では邪魔なだけであった。

LyZ は LyX で Zotero のコレクションを直接導入できるようになる。LyX は latex のフロントエンドプロセッサであり、これについては別の機会に紹介したい。

3.4 Zotero 変換の注意

Zotero のコレクションを bib ファイルに出力する際の注意を述べておく。引用元の情報がいいかげんなことがある。doi から入力するとある程度信頼できると考えられるが、IDEAS や CiNii はかなり多くのまちがいがあある。一旦印刷してみないと気付かないまちがいも多い。自動入力に頼らず、確認も必要である。

他にも自動入力のままだと、うまく出力されないところがある。また CiNii の情報で Zotero にとりこんだ場合、年月が対応していないので修正が必要である。版の箇所も数字だけで十分であるが、自動入力で「第 2 版」となっていると最終出力の表記がおかしくなる。あとアドオンで Google Scholar Citation をいれていると、ノートの箇所に「引用数」が自動的に挿入される。

3.4.1 日本語の姓名が逆になる。

日本語文献について、名前を姓と名を別々にいれると、表示が逆になるので注意が必要である。このことについて、私は Zotero の姓の場所に姓名を入力しておくようにしている。ただ最近の `jecon.bst` は、これに対応した `bst.sei.mei.order` フラグが導入されているのでそれに対応できる。

3.4.2 翻訳者の名前や「よみ」がない。

翻訳書の書誌情報で2つ問題がある。`jecon.bst` で翻訳者のフィールドとして `jauthor` が求められているが、Zotero のエクスポートでは `translator` になる。また、和書をひらがな順もしくは、洋書と同様にアルファベット順にするための `yomi` フィールドは出力されない。

一番簡単な解決方法は、`bibtex` に出力される度に `translator` フィールドを `jauthor` に変更し、`yomi` フィールドを付け加え、よみがなを追加すればよい。

ただ Zotero のほうで修正があるたびに `Texmaker` などのエディタを使い一括変換してもよいが、一々修正することになり面倒である。一括で済むように工夫するには Unix に馴染みがある人は `sed` を利用すればよい。Zeotro にある翻訳者名はそのまま、「よみ」として「寄稿者」を割り当てておく。`bib` ファイルとして出力したあと次のコマンドを入力する^{*8}。

```
sed -e "s/translator/jauthor/" -e "s/collaborator/yomi/" -e
変更前.bib >> 変更後.bib
```

他のやり方として Zotero の設定ファイル `BibTeX.js` の修正をすれば一括でできる。これについて補論を参照されたい。

^{*8} 実際のコマンドは一行で入力される。

3.4.3 日本語の bib キーが生成されない。

Zotero を使うとき英語文献のみの bib ファイルはそのままの出力でほぼ対応できてる。日本語文献では bib キーが上手く出力されない。

この修正に JabRef を利用するのがよい。JabRef は、Zotero と同じように、bib キーを自動的に設定する機能がある。Zotero では日本語文献についてはキーをうまく作成しないのだが、JabRef ではデフォルトで作成可能である。ただ翻訳書の原作者名は名前の半角アルファベットとカタカナまじりのため、デフォルトでは名前が年数だけの bib キーになってしまう。そのため手動で入力する必要がある。

他のやりかたとして設定ファイルの BibTeX.js を修正すればよい。これについて補論を参照されたい。

3.5 tablet について

pdf で論文を読むとき iOS や Android のタブレットを使えば、パソコンで読むより読みやすい。私は古い人間なので、論文についてほとんどいちいち印刷して書き込んでいるが、おそらく若い研究者はタブレットでそのまま読むことが多いだろう。現在、Apple の iPad Air 2 と Google の Nexus 7 と Sony のデジタルペーパー (<http://www.sony.jp/digital-paper>) を保有しているので、その感想を簡単に述べたい。

iPad Air 2 の OS は iOS であり、Android に比べタッチパッドの使い心地が格段によい。さらに iOS に辞書が内蔵されており、pdf ファイルで未知の単語を長押しすれば辞書が開いて意味を教えてくれる。有料アプリ Goodreader などを利用して、いくつかの書き込み等も可能である。

さらに、iOS には zotpad という無料アプリがある。これによって Zotero の自分のライブラリを閲覧したり、編集したりすることができる。また iOS には Papership というアプリがあり、これは Zotero と mendeley の両方に対応しているようであるが私は試していない。執筆時点で iOS は公式には

Firefox に対応していないが、Foxbrowser というアプリがあり、アドオンは利用できないが、ブラウザの共有機能が有効なので、ブックマークなどが利用できる。

Android は、Gmail や Google Calender などパソコンでも Google 提供のサービスを使っていると便利である。また Android には Firefox のアプリがありブックマークが同期できる。ただパソコン用のアドオンは利用できない。なお Android で Zotero を利用するには Zandy を用いればよい。Nexus 7 は iPad に比べ軽量なので、ノートパソコンとともに持ち運ぶときはもっぱらこちらを使っている。

Sony のデジタルペーパーは e-ink を使っている。e-ink により、画面は白黒であるが目にはやさしくバッテリーの消耗も少ない。pdf を読むことができ、デジタイザペンをつかって書き込みが可能である。さらに WiFi に対応し、WebDAV での接続が可能である。Zotero との連携を模索したのだが、デジタルペーパーから Zotero の WebDAV サーバから直接論文をやりとりするのは困難である。また論文を読むときに iOS のように辞書検索はしてくれない。そのため私は主に自分の原稿を出力し、それを校正するのにつかっている。何度もプリンタに出力する必要がなくなり、紙の節約になっている。

論文を読むためにどの tablet がいいかというと圧倒的に iOS である。ただ私はこの夏、新型の Retina ディスプレイの macbook を購入した。この機種は 1kg と軽量で iPad でやれることはほとんどすべてやれる。そのため外出先で iPad の使用頻度は低くなってしまった。むしろ軽くてディザリングができる Nexus 7 をよく持ち出している。

4 まとめ

以上私の文献管理の仕方を紹介した。つまり Zotero で参考文献を収集し管理し、jecon.sty に対応した bib ファイル形式で出力し、xelatex およ

び upbibtex で論文や参考文献リストを作成することで、非常に簡単にそして正確に参考文献が作成が可能となることを示した。

Zotero の文献管理についてウェブ上からワンクリックで登録できる。さらに Zotero のファイルは添付ファイルを Box.net を通じて管理することで、複数のパソコンで利用可能になる。Zotero 日本語文献について次の 2 点に注意する。

1. 姓フィールドのところに姓名を入力する、
2. 寄稿者フィールドのところによみがなを入力するようしておくことで jecon.sty に対応した bib ファイルを作成しやすくなる。

さらに、補論のように Firefox のソースコードを変更すれば、Zotero のエクスポートのみで、jecon.bst に対応した bib ファイルを作ることができる。

Zotero と jecon.bst を特に変更することなくオリジナルのまま用いるなら、次のようにすればいいだろう。

1. 英語文献なら zetoro のエクスポートで一発で変換できる。
2. 日本語文献が含まれていても文献リストの出力なら、bib キーは必要ないので、JabRef に頼らず、zetoro のエクスポートに sed で一括変換すればよい。
3. 日本語文献が含まれていても翻訳書が存在しないなら、JabRef で自動的に bib キーを生成することができる。
4. 翻訳書が存在する日本語文献の場合、JabRef で一部手動で bib キーを生成する必要がある。

ただもとの書誌情報が更新されるたびに変更の必要がある。

なお、ここでは説明しなかったが Zotero は Microsoft Word などにも対応していて Word のアドオンとしてワンクリックで引用文献が作れる。英語論文での利用ならこれで十分である。しかしながら日本語論文の引用はどうか私はわからない。今後 Word で日本語論文を簡単に引用できる方法を学部教育のために考えていきたい。

また、この原稿をほぼ書き終えたときに気付いたが、Better BibT_EX (<https://zotplus.github.io/better-bibtex/>) というアドオンがある。十分検証していないが、先に議論したような日本語文献を含めたbibファイルの出力がスムーズにいくことが期待される。

参考文献

- Patashnik, Oren (1988) “BiBTeXing,” URL: <ftp://ftp.u-aizu.ac.jp/pub/tex/CTAN/biblio/bibtex/base/btxdoc.pdf>.
- 奥村晴彦・黒木裕介 (2013) 『[改訂第 6 版] L^AT_EX 2 ϵ 美文書作成入門』, 技術評論社, 東京。
- 宮崎憲治 (2015) 「学術研究のためのオープンソース・ソフトウェア (1) : XeLaTeX」, 『経済志林』, 第82巻, 第4号, 285–321頁, 3月。

補論

以下 Zotero の出力で, 一発で `jecon.bst` に対応した `bib` ファイルを作成するためにどのように設定ファイルを変更すればよいかを説明する。そのために Zotero の設定ファイル `BibTeX.js` を改造する必要がある。この変更により, Zotero が動かなく可能性があるのですが, あくまで自己責任で改造していただきたい。このファイルは javascript で書かれており, 自分で修正可能なのがオープンソースコードの素晴らしいところである。

設定ファイル `BibTeX.js` の場所であるが, Zotero の環境設定で詳細のタグに移動し, 「データ・ディレクトリを表示する」をクリックする。そこから `zotero`, `translator` のフォルダを開けば, 該当ファイルは見付かる。Mac や Ubuntu だと, コマンドラインから

```
find ~/ -name "BibTeX.js"
```

で, 場所をさがせる。このオリジナルを別名で保存しておく。

`bib` ファイルの出力の際に, 翻訳者を `jauthor` フィールドに, 寄稿者を `yomi` フィールドに変更するための `BibTeX.js` の修正は以下である。

```
if (translator){
    writeField ("translator","{" + translator.substr (5)+
        "}",true);
```

```

    }
    if (collaborator){
        writeField ("collaborator","{" + collaborator.substr
            (5)+ "}",true);
    }

```

が執筆時点の最新バージョン（4.0.27）では1264 行にあるので、

```

    if (translator){
        writeField ("jauthor","{" + translator.substr (5)+
            "}",true);
    }
    if (collaborator){
        writeField ("yomi","{" + collaborator.substr (5)+
            "}",true);
    }

```

に変更すればよい。

日本語文献でも **bib** キーを作るための変更は、

<http://d.hatena.ne.jp/pinkfive/20101112>

がある。ただ、現在の Zotero とバージョンが違うので、そのまま使えない。先のサイトを参考に BibTeX.js の以下の部分

```

var citeKeyCleanRe = /^[^a-z0-9\!\$\&\*\+\-\.\ \/\:;\<\>\?\[\\]\^\_\\`|]+/g;

var citeKeyConversions = {
    "a":function (flags,item){
        if (item.creators && item.creators[0] && item.
            creators[0].lastName){
            return item.creators[0].lastName.toLowerCase

```

```

        ().replace (/ /g, "_").replace (/,/g, "");
    }
    return "";
},
    "t":function (flags,item){
        if (item["title"]){
            return item["title"].toLowerCase ().replace
                (citeKeyTitleBannedRe, "_").split (/\\s+/g)[0];
        }
        return "";
    },
    "y":function (flags,item){
        if (item.date){
            var date = zotero.Utilities.strToDate
                (item.date);
            if (date.year && numberRe.test (date.year)){
                return date.year;
            }
        }
        return "????";
    }
}

```

が執筆時点の最新バージョン (4.0.27) では10694行以降にあり、それを以下のように置き換える:

```

//var citeKeyCleanRe = /^[^a-z0-9!@$&*+\\-\\.\\/\\:;\\<\\>\\
    ?\\[\\]\\^\\_\\`\\|]+/g;
var citeKeyCleanRe = /["\\#\\%\\'\\ (\\)\\,\\=\\{\\}\\~\\|]+/g;

```

```

var citeKeyJapaneseRe = /[\\u3400-\\u4DBF\\u4E00-\\u9FFF\\
                        uD840-\\uD87F\\uDC00-\\uDFFF\\uF900-\\uFAFF]/;
var citeKeyConversions = {
  "a":function (flags,item){
    if (item.creators && item.creators[0] && item.
        creators[0].lastName){
      // return item.creators[0].lastName.
      toLowerCase ().replace (/ /g,"_").
      replace (/,/g,"");
      var tmp = item.creators[0].lastName.
        toLowerCase().replace (/ /g,"_").replace
        (/,/g,"");
      if (tmp.match (citeKeyJapaneseRe)){
        return item.creators[0].lastName.
          toLowerCase().replace (/ /g,"_").replace
          (/,/g,"");
      }
      else return tmp;
    }
    return "";
  },
  "t":function (flags,item){
    if (item["title"]){
      //      return item["title"].toLowerCase ().replace
      (citeKeyTitleBannedRe,"").split (/\\s+/g)[0];
      var tmp = item["title"].replace (/\\s+/g,"_").
        toLowerCase ();
      var tmp = tmp.replace (citeKeyTitleBannedRe,"").
        split(/\\s+/g)[0];
    }
  }
};

```

```

        if (tmp.match (citeKeyJapaneseRe))return tmp.
        substr (0,10);
        else return tmp;
    }
    return "";
},
"y":function (flags,item){
    if (item.date){
        var date = zotero.Utilities.strToDate (item.
        date);
        if (date.year && numberRe.test (date.year)){
            return date.year;
        }
    }
    return "????";
}
}

```

そうすれば、bib キーとして日本語の姓名が出力される。

現状はこれでうまくいっている。バージョンが変わると使えないかもしれない。そのたび再度修正する必要があるので差分ファイルを作っておき、パッチを当てるようにすればよい。

パッチも今後使えなくなるかもしれないが Firefox や Zotero はオープンソースなので、ソースコードが公開されている。

<https://github.com/zotero/zotero>

ソースコードは **git** で管理されているので、それを **clone** して、バージョンアップごとに **merge** して自分でコンパイルしていけばよいだろう。バージョン管理システム **git** についてはどこかの機会で解説したい。