LaTeX Crash Course
Meeting 3: The Bibliography Management

Jan Sprenger

June 25, 2021

The bibliography management is one of the nicest features of LaTeX. We first look at the place where references are stored: a file with the ending `.bib`.

1 Reference Management

The inner life of a .bib file

A `.bib` file is a collection of bibliographic entries such as the following:

```
@Book{Carnap1950,
  title     = {{Logical Foundations of Probability}},
  publisher = {University of Chicago Press},
  year      = {1950},
  author    = {Carnap, Rudolf},
  address   = {Chicago},
  keywords  = {Objective Probability},
}
```

This format for storing references is called `BiBTeX`—not to be confused with the `bibtex` engine for working on .bib files (see Section 2). This is an example of a reference in **book** format with the basic information: author, title, publisher, year (and a bit more). For **journal articles** the fields are slightly different:

```
@Article{Frege1892,
  author  = {Frege, Gottlob},
  journal = {{Zeitschrift f\"ur Philosophie und philosophische Kritik}},
  title   = {{\"Uber Sinn und Bedeutung}},
  year    = {1892},
```

```
  pages   = {25--50},
  volume  = {100},
}
```

Here we have the journal, the number, the volume and the page range as category-specific fields. The third important category for philosophers are **chapters in edited volumes**, for example:

```
@InCollection{Garber1983,
  Title                    = {{Old Evidence and Logical Omniscience}},
  Author                   = {Garber, Daniel},
  Booktitle                = {Testing Scientific Theories},
  Publisher                = {University of Minnesota Press},
  Year                     = {1983},
  Address                  = {Minneapolis},
  Editor                   = {Earman, John},
  Pages                    = {99--132}
}
```

There are many more categories, like `@phdthesis`, `@inproceedings` (for conference proceedings), `@TechReport`, and so on. They are frequently used in the natural sciences and in economics, but you are less likely to encounter them.

The very first field of a reference entry in BibTeX format is the **citation key** (e.g., Carnap1950,Garber1983, Plebani2018Fictionalism). This is a field you can choose freely according to your preferred convention. You need the key for calling the reference in the LaTeX file, e.g. `\cite{Carnap1950}`.

Managing .bib files

There are a lot of free programs that allow you to manage `.bib` files. JabRef, Referencer and Zotero are the most well-known ones. I recommend **JabRef** since it allows you a great deal of control and it has the advantage of being specifically designed for the `.bib` format. Therefore it is stronger in features such as sorting, grouping, debugging your `.bib` file, defining new entry types, and so on.

JabRef also has a very nice feature: it looks up all the bibliographic information on the internet and generates a BiBTeX entry when you paste the DOI, the ISBN, the arXiv ID (this is an important preprint archive) in its "ID-based entry generator". It does not always work, but impressively often.

Zotero also seems to work fine, all in all, but I would only use it if you regularly write papers in MS Word and similar WYSIWYG editors where

.bib files are of little use. I have no experience with **Referencer** but it is also designed for `.bib` files.

Building and expanding your reference database

On the website `philpapers.org` you can find almost any paper ever written in philosophy or a related discipline. Even better: they have a citation function that allows you to export the reference in BibTeX format. You can just make a new entry in your reference manager and paste the code there.

Some people also share their `.bib` files online. For example, Ted Sider has a large database of almost 2000 entries, mainly on analytic metaphysics. There may be other philosophers, too. I will also send you my own master file.

To avoid copying the relevant references for each new paper/project, I recommend to have just one big file and to refer to it from your `.tex` file (=your paper) by calling a path-specific file, e.g.

```
\addbibresource{../../References.bib}
```

goes up two levels in the hierarchy and calls `References.bib` in that folder.

2 The Bibliography Engine of LaTeX

This section explains the mechanism behind LaTeX's referencing system. It is a bit of "graue Theorie", but very useful for understanding the logic behind LaTeX's referencing system, and it will help you to debug your code if necessary. The actual citation commands will be explained in the next section.

In order to process the information stored in a `.bib` file (which may have lots of entries...), LaTeX needs an engine that extracts the information that is relevant for this specific paper. This information is then stored in a `.bbl` file and linked to the paper such that you actually have, instead of `\cite{Frege1892}`, "Frege 1892".

There are two engines: `bibtex` and `biber`. BibTeX is set as the default by many LaTeX editors, including TeXShop and TeXStudio. These engines then work together with *packages* that define citation macros such as `\citet{Frege1892}`, `\textcite{Carnap1950}`, and so on.

You can call either the `natbib` or the `biblatex` package. If you call `natbib`, you have to use the `bibtex` engine by force. `biblatex` works either with `biber` or with `bibtex`, though calling `biber` is recommended. Figure 1 explains the different levels of bibliography management.

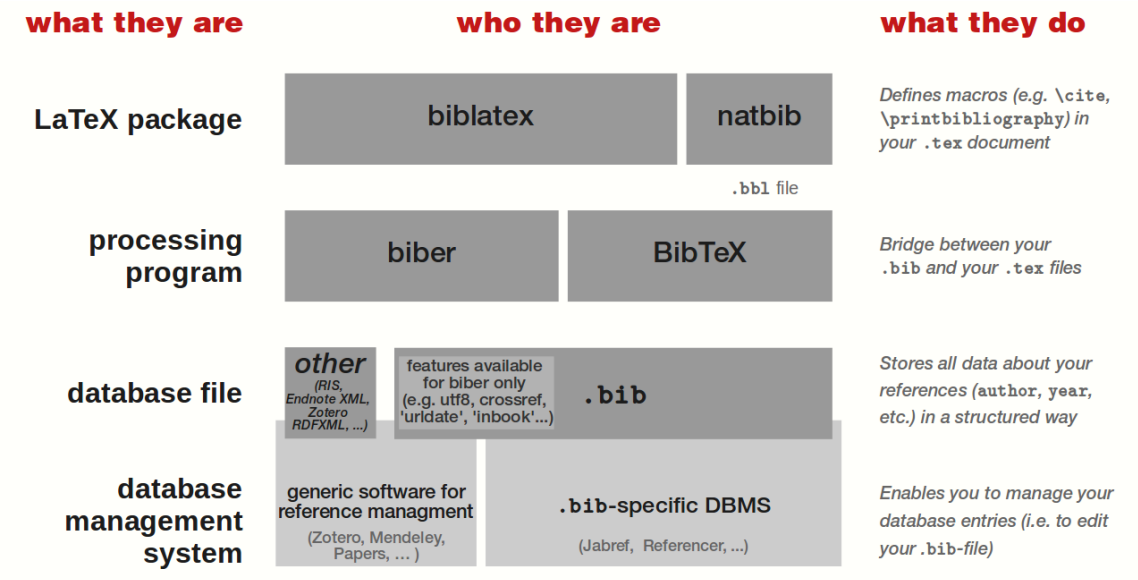

Figure 1: A graphical illustration of the different roles of bibliography engines and bibliography packages.

I recommend to use the `biblatex` package with the `biber` engine. There are several reasons:

- Neither `bibtex` nor `natbib` have not received any major updates in the last years, while the `biblatex`/`biber` combo is continuously developed.
- `bibtex` has been developed in the 1980s and it therefore struggles with any non-ASCII input in the .bib files (e.g., umlaute, diacritics, etc.) So you have to rewrite the character "ü" as "`\"u`, which is sort of annoying. By contrast, `biber` has full support for unicode-compatible character encodings, such as `UTF-8`.
- Configuring `natbib` to make it suit your needs is much harder than configuring `biblatex`. You basically have to learn a new language and to modify `.bst` files, which is nothing you really want to do. I speak from experience. By contrast, `biblatex` allows you to set the desired options when calling the package in the preamble of your `.tex` file.
- `biblatex` has support for multiple bibliographies (e.g., per book chapter).

3 `biblatex`: Styles and Citation Commands

You call the `biblatex` package in the preamble of your file. The most important option is to specify the *backend* (i.e., the engine: `biber` or `bibtex`) and the *style* (numeric, author-year, etc.). Thus, you will add a line such as

```
\usepackage[backend=biber,style=authoryear]{biblatex}
```

to your preamble.

For reasons explained above, I recommend to use the `biber` engine. The style determines how in-text citations appear and how the bibliography is sorted. This decision depends on your audience, but most professors will probably prefer the `authoryear` style (=Frege 1892). Some of our historians may prefer `authortitle` (=Frege: "Über Sinn und Bedeutung"). In natural science the simple `numeric` style (=[1],[2],[3], in order of citation) prevails. See the highly useful `biblatex` cheat sheet for more information and more information on styles.[1]

Still in the preamble, you should tell LaTeX the .bib file which `biber` should use. This happens with the following command:

```
\addbibresource{SampleReferences.bib}
```

Finally, at the end of the document (in reality, wherever you want...) you can tell LaTeX to print the bibliography:

```
\printbibliography
```

LaTeX will print the cited references—and only those that have been cited. If you want a longer reference list, you can add other items with the `\nocite` command. For exmple, `\nocite{Savage1954}` will add Savage's "Foundations of Statistics" to the literature list, even if it has not been cited.

I haven't yet told you **how to cite a reference**. Very simple: you use one of the numerous citation commands and you add the citation key of the reference. I explain the most common commands, always relative to author-year style. A full overview is in the `biblatex` cheat sheet.

- `\cite{Carnap1950}` prints "Carnap 1950". Use this command to refer to the *work*. Author and year form a semantic unit; they are a noun phrase analogous to "decision theory".

[1] If you look for a specific style, there is a high probability that somebody has already implemented it and that you can call it as an option.

- `\textcite{Carnap1950}` prints "Carnap (1950)". Use this command to refer to the author, e.g., "as Carnap (1950) showed...".
- `\parencite{Carnap1950}` prints "(Carnap 1950)". Use this command as a cross-reference
- `\footcite{Carnap1950}` prints the reference in a footnote. Don't do this in the term papers for my courses. It can make sense, however, if you use a more traditional humanities style like `authortitle` or `verbose` where references are often put in the footnotes.

4 Some Q&A

Q: The entry in my bibliography contains also DOI, URL, and other irrelevant information. How can I switch this off?
A: Call the `biblatex` package with the options `doi=false, url=false, eprint=false`, etc.

Q: (same question, essentially) This does not work for the issue number of a journal article. I just want the volume number!
A: Add `\AtEveryBibitem{\ifentrytype{article}{\clearfield{number}}{}}` to your preamble. Analogous instructions work for other fields you want to suppress. The structure of the macro is an if-else statement: if the condition is satisfied (i.e., the entry is of type `article`), then execute the commands in the first curly bracket. Otherwise ("else") do what is specified in the second curly bracket (here: nothing).

Q: In the author-year style BibLaTeX introduces the page range of a journal article by default with "pp.". How can I switch this off? I just want the page range.
A: Add `\DeclareFieldFormat[article]{pages}{#1}` to your preamble. Omitting the `[article]` option eliminates "pp." for all publication types.

Q: I want to work with `biblatex`, but I have worked in the past with `natbib` and the `natbib` citation commands. Do I have to replace them all when I copy and paste material from that old paper into a new one?
A: No worries: just call `biblatex` with the option `natbib=true`.

Q: I am citing three chapters from the same book. Can I tell LaTeX that it need not print the book details any time I cite a chapter?

A: Yes: the solution is to modify the entry of the chapter in the `.bib` file and to add the field `crossref= {BigBook2000}` (where BigBook2000 is the citation key of the collection from which you are citing chapters). Check the internet for further details.

Q: An error occurs in compiling. What now?
A: Go to the log file (e.g., for `example.tex` this is `example.log`) and copy the error message at the end of the file into your search engine.

Q: I compile my `.tex` file and at the place of my in-text citation I only see question marks in boldface.
A: Probable causes: something is wrong with your citation keys, you have called the wrong `.bib` file, or you have compiled only once. On the terminal, you need to call `pdflatex`, then `biber`, and then again `pdflatex`. In most editors, it suffices to call `pdflatex` (or its cousin, `latexmk`) twice and the references will show up correctly.

Q: Some ugly bibliography-related error occurs and the internet does not tell me what to do.
A: Good chance that your `.bib` file is corrupted: empty citation keys, no year specified (this is bad if you use author-year style), etc. Go to JabRef and fix it there.

5 Essential Internet Resources

- The famous BiBLaTeX cheat sheet.
- Wiki book on LaTeX, section on bibliography management.
- Overleaf Tutorials: Getting Started with BibLaTeX and Bibliography Management with BibLaTeX.
- More explanation on biblatex vs. bibtex and natbib vs. biber: here, here and here.