User guide for the dumbib package[*]

Shivam Garg (sgdpsi@gmail.com)

February 19, 2025

The dumbib package helps with bibliography management for LaTeX, by creating forward links (i.e. when you cite a paper, the in-text citation links to the bibliography entry in the reference section) and backward links (i.e. the bibliography entries link back to all the places in the document where they were cited). **This is the only functionality that this package provides**; to remind users of how minimalistic dumbib is in its processing, it even has the word “dumb” (or “dum” for dummy, depending on how you prefer it) in its name.

Owing to this minimalistic processing, dumbib gives its users a near complete control over how their bibliography looks like, which is its primary appeal. This is in contrast to popular bibliography management packages such as BibTeX or BibLaTeX, which provide a lot of functionality but require additional effort from the users to tweak the program output to their exact liking. However, for some people (in particular those who like to manage their bibliography manually) the flexibility that dumbib affords by requiring that its users take care of the missing functionality should make the trade-off worthwhile. The nice thing about this arrangement is that dumbib takes care of the core LaTeX issues (providing forward and backward links) and the users do the remaining work, such as formatting the references in a unified style—work which can be readily automated using, say, a Python script. (This argument assumes that for some users, programming in Python would be a more comfortable experience than tweaking other bibliography packages or, worse, programming in LaTeX.)

To use dumbib, ensure that the dumbib.sty file (which is available at the link given in the footnote at the bottom of this page) is in the path of your system and then include the package in the preamble of the TeX document as follows:

```
\usepackage{dumbib}
```

The package provides three main commands for bibliography management:

1. \dumbibReferenceEntry for creating a bibliography database,
2. \cite for citing the references in the text, and
3. \dumbibCreateBibliography for laying out the references in the bibliography section.

[*] The package is available at the following link: https://github.com/svmgrg/bibtex_alternative.

1 Creating a bibliography database

At the beginning of your document you will need to create a database of all the references you plan to use in your document so that `dumbib` is aware of them. This is done using the command

```
\dumbibReferenceEntry[<optional>]{<key>}{<author>}{<year>}{<citation_text>}
```

which has four mandatory arguments:

- `<key>` is the citation key which you will invoke when you want to cite this reference,
- `<author>` and `<year>` refer to the author(s) and the year of publication respectively that appear in-text when you cite this reference, and
- `<citation_text>` is the bibliography entry that will go in the reference section.

There is also an optional argument `<optional>`, which can be used to store text for creating variations while citing references (see Section 4 for details). Note that if nothing is specified in the `<optional>` field, the text '`??`' gets automatically stored instead. We now give some examples.

The following snippet adds two references to the `dumbib` database in an APA-like style:[1]

```
\dumbibReferenceEntry{lowry_etal1951}{Lowry et al.}{1951}{
  Lowry, O. H., Rosebrough, N. J., Farr, A. L., Randall, R. J. (1951).
  Protein measurement with the Folin phenol reagent.
  \textit{Journal of Biological Chemistry, 193}(1), 265-275.
}

\dumbibReferenceEntry{noorden_etal2014}{Van Noorden, Maher, and Nuzzo}{2014}{
  Van Noorden, R., Maher, B., Nuzzo, R. (2014).
  The top 100 papers. \textit{Nature News, 514}(7524), 550.
}
```

The above snippet will need to go **after** the `\begin{document}` command in your TEX file. (If you prefer, you can instead put these reference declarations in a separate file, say, `database.tex`, and include that in your main TEX file using the command `\include{database.tex}`.)

Note that you are free to specify exactly how the citation appears in-text and in the reference section. For instance, you could have instead declared the article by Van Noorden, Maher, and Nuzzo (2014) to `dumbib` as follows:

```
\dumbibReferenceEntry{noorden_etal2014}{Van Noorden et al.}{2014}{
  R. van Noorden, B. Maher, R. Nuzzo. (2014).
  The top 100 papers. \textit{Nature News}.
}
```

[1] Even though in this example, the author list, the paper title, and the publication venue appear in separate lines inside the `\dumbibReferenceEntry` command, this was done only to improve readability; `dumbib` doesn't care if they all were instead formatted in a single line.

However, since you have complete freedom in deciding exactly how the references appears in-text, you will have to manually take care of things which other bibliography management packages could have automated for you. For instance, if your bibliography has multiple articles from the same author(s) all published in the same year, you will **manually** need to append ‘`a`’, ‘`b`’, etc. to their year of publication:

```
\dumbibReferenceEntry{bach2023a}{Bach}{2023a}{
  Bach F. (2023a).
  \textit{Learning theory from first principles.} MIT press.
}

\dumbibReferenceEntry{bach2023b}{Bach}{2023b}{
  Bach F. (2023b).
  On the relationship between multivariate splines and infinitely-wide
  neural networks. \textit{arXiv:2302.03459.}
}
```

2 Error handling

Whenever `dumbib` encounters an error, it prints a succinct warning message on the terminal, and another verbose error message that will **ostentatiously appear in the PDF document**. For instance, if you create two different reference entries with the same key using the command `\dumbibReferenceEntry`, then `dumbib` will raise an error and **ignore the second entry all together**, i.e. the citation key will continue pointing to the reference that was declared first. This behavior is illustrated in the example below. (Although, `dumbib` will gladly accept two otherwise identical references as long as they have a different key; it is really dumb!)

The following snippet declares two references using the same key “`talagrand2022`”, resulting in the warning and the error message displayed below.

```
\dumbibReferenceEntry{talagrand2022}{Talagrand}{2022}{
  Talagrand, M. (2022).
  Upper and lower bounds for stochastic processes: Decomposition theorems.
  \textit{Springer Nature.}
}

\dumbibReferenceEntry{talagrand2022}{Empty}{0000}{
  Anonymous. (1111). Empty example reference. \textit{Empty journal.}
}
```

This is the succinct warning message that will be printed in the terminal:

```
Package dumbib Warning: Error on line 160! Key ``talagrand2022'' already
(dumbib)                defined.
```

This is the verbose error message that will appear in your output PDF:

Error on line 160! The key “`talagrand2022`” has been already defined. Consider using a different key name.

Original entry: Talagrand, M. (2022). Upper and lower bounds for stochastic processes: Decomposition theorems. *Springer Nature.*

New (ignored) entry: Anonymous. (1111). Empty example reference. *Empty journal.*

The above error message is, by design, excessively intrusive so as to attract the user’s attention to the errors. Once the user solves the corresponding error, the message will disappear.

Raising errors only on the terminal

In case you do not want to see such verbose messages in the output PDF document, you can load the `dumbib` package as follows:

```
\usepackage[error_mode]{dumbib}
```

In this mode, `dumbib` will not show any errors in the output PDF, and instead raise a succinct message as an error on the terminal, i.e. it will pause the LaTeX processing and require the user to press the `<return>` key in order to continue.

3 Citing references in-text

The command to cite a reference in-text is

```
\cite[*][<optional 'a', 'y', 'o', or 'm'>][<manual_text>]{<key>}
```

This command takes the citation `<key>` which is used to identify the reference, along with an optional star, or an optional ‘`a`’, ‘`y`’, ‘`o`’, or ‘`m`’ argument. The six valid variations and their corresponding in-text outputs are as follows:

Variation	In-text output
`\cite{<key>}`	`<author> (<year>)`
`\cite*{<key>}`	`<author>, <year>`
`\cite[a]{<key>}`	`<author>`
`\cite[y]{<key>}`	`<year>`
`\cite[o]{<key>}`	`<optional>`
`\cite[m][<manual_text>]{<key>}`	`<manual_text>`

where the `<author>`, the `<year>`, and the `<optional>` fields correspond to those that were specified when creating the `dumbib` bibliography database.[2] The last row in the above table shows the usage of the manual cite command `\cite[m]`, where `<manual_text>` field specifies exactly how the citation appears in text. (If you use a combination other than the ones listed above, or cite a reference which has not been declared a priori, `dumbib` will raise an error.) We now give an example that illustrates the use of these commands.

The following snippet

```
According to \cite{noorden_etal2014}, the most cited paper in recorded
history is a biology paper by \cite[a]{lowry_etal1951}\ that has been cited
hundreds of thousands of times. Unfortunately, we do not conduct experiments
involving proteins, and thus are more likely to cite some probability or
machine learning texts (such as \cite*{talagrand2022}; \cite*{bach2023a};
\cite[y]{bach2023b}). In particular,
\cite[m][the 2022 book by Talagrand]{talagrand2022} is very readable.
```

produces the following text:

> According to Van Noorden, Maher, and Nuzzo (2014), the most cited paper in recorded history is a biology paper by Lowry et al. that has been cited hundreds of thousands of times. Unfortunately, we don't conduct experiments involving proteins, and thus are more likely to cite some probability or machine learning texts (such as Talagrand, 2022; Bach, 2023a; 2023b). In particular, the 2022 book by Talagrand is very readable.

As the above example shows, **the user needs to manually take care of everything down to the very basic details**. For instance, in the above example, we provided the opening and closing parentheses, the semi-colons, and even ensured that only the year of publication appeared in the last citation (Bach, 2023b). We also had to put a backslash after the command `\cite[a]{lowry_etal1951}` to ensure optimal spacing following a period,[3] just like we would have done had we typed the entire thing manually, i.e. "`Lowry et al.\`".

Note: To create the links, `dumbib` uses the `hyperref` package under the hood. By default, such links would appear as colored bounding boxes. Instead, if you want your links to appear as dark red-colored texts without any bounding boxes, as shown in the example above, you can include the following command in the preamble of your TEX file:

```
\hypersetup{colorlinks, linkcolor={red!33!black}}
```

[2] Note that if `<optional>` field is not specified in the `\dumbibReferenceEntry` command, then it is automatically set to the default text '`??`'.

[3] This manual intervention is necessary, since `dumbib` cannot ascertain whether this "author-only" citation was used in the middle of a sentence or at its end.

4 Additional customization

In this section, we will discuss how the `<optional>` argument in the `\dumbibReferenceEntry` command can be used to vary how the references are printed in-text when citing them. We begin by describing yet another command called `\internalCite` in the `dumbib.sty` file, defined as follows:

```
\NewDocumentCommand{\internalCite}{smmm}{
  \IfBooleanTF {#1} {% starred version (used in \cite*)
    #2 , \ #3 % <author>, <year>                           <--- modify this!
  } {% unstarred version (used in \cite)
    #2 \ ( #3 ) % <author> (<year>)                        <--- modify this!
  }
}
```

The function `\internalCite` specifies how citations appear in-text when using the `\cite` command. Inside this function's definition (as shown above), '`#2`', '`#3`', and '`#4`' refer to the `<author>`, `<year>`, and `<optional>` fields respectively. To change the default behavior of the `\cite` and `\cite*` command, the user needs to modify two lines in the `\internalCite` function's definition (as highlighted above), to get the desired output. The following table shows some variations:

Variation	In-text citation is printed as:
`#2 , \ #3`	`<author>, <year>`
`#2 \ #3`	`<author> <year>`
`#2 \ ( #3 )`	`<author> (<year>)`
`#4`	`<optional>`
`[ #4 ]`	`[<optional>]`
`#4 \ ( #3 )`	`<optional> (<year>)`
`#2 \ [ #4 ]`	`<author> [<optional>]`

For instance, say that, the user wants to use the style "Lowry et al. 1951" with the `\cite*` and use the style "Lowry and colleagues (1951)" with the `\cite` command. In this case, the user can modify the `\internalCite` command to use the variations "`#2 \ #3`" and "`#4 \ ( #3 )`", and define the reference as follows (note the use of the optional argument):

```
\dumbibReferenceEntry[Lowry and colleagues]
{lowry_etal1951}{Lowry et al.}{1951}{
  Lowry, O. H., Rosebrough, N. J., Farr, A. L., Randall, R. J. (1951).
  Protein measurement with the Folin phenol reagent.
  \textit{Journal of Biological Chemistry, 193}(1), 265-275.
}
```

Another relevant use case is when the user wants to use the citation style "[XYZ+2025]", in favor of the using "X et al. (2025)". In this case, the users could set the optional argument in the `\dumbibReferenceEntry` to be "`XYZ+2025`", and modify the `\internalCite` command accordingly.

5 Laying out the bibliography section

The following command lays out the references in the bibliography section:

```
\dumbibCreateBibliography[*][<optional 'o'>]
```

And the following snippet creates a reference section similar to the one at the end of this page:[4]

```
\section*{References} % \addcontentsline{toc}{section}{References}
\dumbibCreateBibliography
```

The \dumbibCreateBibliography command raises an error if a bibliography entry were created in the database but never cited in-text. Also, as can be seen below, **the references are displayed in the same order in which they were declared using the \dumbibReferenceEntry command** during the creation of the dumbib database. Therefore, if you want the references to appear in an alphabetical order, you should declare them alphabetically. Also note that the style of references below is not consistent, because during their declaration, we did not follow a consistent style. Further, if a reference is cited multiple times (such as the book by Talagrand, 2022) on the same page, it will appear multiple times in the backward links as well (see the last reference below).

Using the starred version and the optional 'o' argument

The starred version of this command does not produce backward links to the places where the papers were cited in-text, whereas the unstarred version does. The optional 'o' argument is useful in certain cases, such as when the user is using the citation style "[XYZ+2025]", where each entry in bibliography needs to begin with some text like "[XYZ+2025]". In such cases, the user can use the optional 'o' argument, which will result in each reference entry beginning with "[<optional>]", where the <optional> field is the one specified while creating the bibliography database.

References

Lowry, O. H., Rosebrough, N. J., Farr, A. L., Randall, R. J. (1951). Protein measurement with the Folin phenol reagent. *Journal of Biological Chemistry, 193*(1), 265-275. (cited on page 5)

Van Noorden, R., Maher, B., Nuzzo, R. (2014). The top 100 papers. *Nature News, 514*(7524), 550. (cited on pages 2 and 5)

Bach F. (2023a). *Learning theory from first principles.* MIT press. (cited on page 5)

Bach F. (2023b). On the relationship between multivariate splines and infinitely-wide neural networks. *arXiv:2302.03459.* (cited on pages 5 and 5)

Talagrand, M. (2022). Upper and lower bounds for stochastic processes: Decomposition theorems. *Springer Nature.* (cited on pages 5, 5, and 7)

[4] Because of how LaTeX manages hyperlinks, you will need two compilations for the references to appear properly.