Making Markdown into a microwave meal

Vít Novotný

Abstract

In today's academic publishing, many venues request LaTeX source in addition to or instead of PDF documents. This is often for the purpose of editing and improving full-text search. Services such as arXiv, Editorial Manager, and EasyChair require LaTeX source code that can be microwaved in a single run of pdfTeX without shell access and without invoking external programs. This requires that authors include auxiliary files and limits their selection of LaTeX packages. In this article, I will show how LaTeX documents using the Markdown, Minted, and BibLaTeX packages can be precooked and frozen to be later microwaved in a single run of pdfTeX.

1 Introduction

Academic publishers often require that LaTeX documents can be microwaved in a single run of pdfTeX without shell access and without invoking external programs. Precooking and freezing the documents to this form can be a daunting task for authors, who may opt out of using powerful LaTeX packages just to save themselves the hassle.

The Markdown package [3, 4, 5] allows the authors to mix the familiar lightweight markup of Markdown with LaTeX, but requires shell access and invokes Lua. The Minted package [7] provides a simple interface for typesetting listings with syntax highlighting, but also requires shell access and invokes Python. The BibLaTeX package [2] automates many aspects of bibliography management that require careful manual work in LaTeX, such as sorting and formatting, but requires the invocation of Perl.

In this article, I will show by example how a LaTeX document using the Markdown, Minted, and BibLaTeX packages can be precooked and frozen, so that it can be later microwaved in a single run of pdfTeX. This will allow the authors to quickly prepare documents using powerful packages without spending a thought on the publisher's demands.

2 Technical details

Here I describe the technicalities of precooking and freezing the outputs of the Markdown, Minted, and BibLaTeX packages before moving on to the example. The famished may skip to the following page.

2.1 Markdown

The Markdown package extracts markdown documents from the main LaTeX document to disk and passes control to the Lunamark Lua parser. Lunamark converts the markdown documents to LaTeX, saves the converted LaTeX documents to disk, and then passes control back to the Markdown package. The Markdown package then inputs the LaTeX documents into the main LaTeX document for typesetting.

Since version 2.9.0 (2020/09/14), the Markdown package supports the `finalizecache` option that makes Lunamark produce a LaTeX document `frozenCache.tex` (the frozen cache) mapping an enumeration of markdown documents to their converted LaTeX documents, and the `frozencache` option that makes the Markdown package use the frozen cache instead of Lunamark. The resulting LaTeX document becomes portable, but further changes in the order and content of markdown documents are not reflected. Appearance of markdown documents can still be adjusted by redefining the LaTeX macros that render markdown elements.

2.2 Minted

The Minted package extracts code listings from the main LaTeX document to disk and passes control to the Pygments Python package. Pygments converts the code listings to LaTeX, saves the converted LaTeX documents to disk, and then passes control back to the Minted package. The Minted package then inputs the LaTeX documents into the main LaTeX document for typesetting.

Since version 2.2 (2016/06/08), the Minted package supports the `finalizecache` option that makes Minted produce LaTeX documents `listing`⟨*listing number*⟩`.pygtex` (the frozen cache) containing the converted LaTeX documents, and the `frozencache` option that makes the package use the frozen cache instead of Pygments. The resulting LaTeX document becomes portable, but further changes in the appearance, order, and content of listings are not reflected.

2.3 BibLaTeX

The BibLaTeX package extracts citations from the main LaTeX document ⟨*document*⟩`.tex` to disk. The author then invokes the Biber Perl package, either manually or using automation software such as GNU Make, LaTeXMk, Arara [1], etc. Biber converts the citations and an external bibliography database into a precooked bibliography file ⟨*document*⟩`.bbl` and saves it to disk. The BibLaTeX package then inputs the precooked bibliography file for typesetting.

By including the precooked bibliography file with the LaTeX source, we remove the necessity of invoking Biber. The resulting LaTeX document becomes more portable, but further changes in the order of citations and the appearance and content of references are not reflected.

3 Example

We have the following LaTeX document `document.tex` using Markdown, Minted, and BibLaTeX:

```
\documentclass{article}
\usepackage[citations,fencedCode]{markdown}
\markdownSetup{renderers={inputFencedCode=%
  {\inputminted{#2}{#1}\write18{#2 #1}}}}
\usepackage{minted,biblatex}
\addbibresource{bibliography.bib}
\begin{document}
\begin{markdown*}{hybrid,underscores=false}
The following code in *Python* shows how to
produce iid r.v.'s $W_1, W_2, \ldots,  W_k$
s.t. $\prod_i W_i\sim N(0, 1)$:  [@pinelis]
``` python
def pinelis(k, size, iterations=100):
  import numpy as np, numpy.random as npr
  arange1 = np.arange(iterations) + 1
  def eps():
    return (
      (npr.uniform(size=size) < 0.5) *
      2.0 - 1.0
    )
  def gamma(size):
    return npr.gamma(1.0 / k, size=size)
  def rv():
    return eps() * np.exp(
      np.log(2.0) / (2.0 * k) -
      gamma(size) -
      np.sum(
        gamma((size, iterations)) /
        (2.0 * arange1 + 1.0),
        axis=-1,
      ) + np.sum(
        np.log(1.0 + 1.0 / arange1) /
        (2.0 * k)
      )
    )
  return [rv() for _ in range(k)]
import matplotlib.pyplot as plt
fig, A = plt.subplots(1, 3)
W1, W2, W3 = pinelis(k=3, size=10**7)
W = (W1, W1 * W2, W1 * W2 * W3)
D = ('$W_1$', '$W_1W_2$', '$W_1W_2W_3$')
for ax, w, desc in zip(A, W, D):
    ax.hist(w, 'auto', density=True)
    ax.set_title(desc)
    ax.set(xlabel='value', ylabel='pdf')
plt.savefig('plot.pdf', dpi=300)
\end{markdown*}
\includegraphics[width=\textwidth]{plot}
\printbibliography
\end{document}
```

In addition to the LaTeX document, we also have the following bibliography database `bibliography.bib`:

```
@article{pinelis,
  author = {Pinelis, Iosif},
  title = {The exp-normal distribution is
           infinitely divisible},
  journal = {arXiv},
  year = {2018},
}
```

Our LaTeX document may seem simple, but it requires shell access and invokes Lua, Python, and Perl. A publisher is likely to reject it.

First, we remove the directories `_markdown_document` and `_minted-document` if they exist. Second, we change line 1 of our LaTeX document to `\documentclass[finalizecache]{article}` and invoke Python, pdfTeX, and Biber from the shell:

```
$ pip install numpy scipy matplotlib
$ pdflatex -shell-escape document.tex
$ biber document.bcf
```

Third, we change line 1 of our LaTeX document to `\documentclass[frozencache]{article}` and we remove `\write18{#2 #1}` from line 4. Finally, we create a ZIP archive `document.zip` with the files `document.bbl`, `document.tex`, and `plot.pdf` and with the directories `_markdown_document/`, and `_minted-document/`:

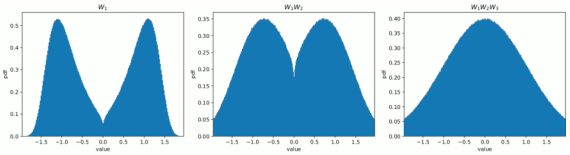
```
$ zip -r document document.{bbl,tex} \
>   plot.pdf {_markdown_,_minted-}document/
```

If our document contained forward references, we would also include file `document.aux`. After the publisher has unpacked the ZIP archive, they only need to microwave the document in a single run of `pdflatex document.tex` before savoring it and wolfing it down.

After typesetting, our LaTeX document produces the following output. Listing was trimmed for brevity (and colors have been grayscaled throughout in the printed article):

The following code in *Python* shows how to produce iid r.v.'s $W_1, W_2, \ldots, W_k$ s.t. $\prod_i W_i \sim N(0,1)$: [1]

```
def pinelis(k, size, iterations=100):
  import numpy as np
  [...]
plt.savefig('plot.pdf', dpi=300)
```



References

[1] Iosif Pinelis. "The exp-normal distribution is infinitely divisible". (2018).
```

4 Conclusion

Powerful new LaTeX packages are created every day. However, authors in academia often avoid them to steer clear of the publisher's fury. In this article, I have shown how authors can have their cake and eat it too: by precooking and freezing, the authors can quickly prepare mouth-watering documents using the Markdown, Minted, and BibLaTeX packages without having to worry about the publisher's demands.

Acknowledgements

This work was funded by the South Moravian Centre for International Mobility and the Brno Ph.D. Talent project.

References

[1] P. R. M. Cereda. The bird and the lion: arara. *TUGboat* 36(1):37–40, 2015. tug.org/TUGboat/tb36-1/tb112cereda.pdf

[2] P. Kime, P. Lehman. BibLaTeX: Sophisticated bibliographies in LaTeX, 2020. ctan.org/pkg/biblatex

[3] V. Novotný. Using markdown inside TeX documents. *TUGboat* 38(2):214–217, 2017. tug.org/TUGboat/tb38-2/tb119novotny.pdf

[4] V. Novotný. Markdown 2.7.0: Towards lightweight markup in TeX. *TUGboat* 40(1):25–27, 2019. tug.org/TUGboat/tb40-1/tb124novotny-markdown.pdf

[5] V. Novotný. Markdown: A package for converting and rendering markdown documents inside TeX, 2020. ctan.org/pkg/markdown

[6] I. Pinelis. The exp-normal distribution is infinitely divisible. *arXiv*, 2018. arxiv.org/abs/1803.09838

[7] G. Poore. Minted: Highlighted source code for LaTeX, 2017. ctan.org/pkg/minted

◇ Vít Novotný
Nad Cihelnou 602
Velešín, 382 32
Czech Republic
witiko (at) mail dot muni dot cz
github.com/witiko