Easier LaTeX with Pandoc and Markdown

Nguyen, Duc Hieu

May 30, 2018

Markdown and Academic Writing

Academic writing involves writing, storing, sharing and retrieving documents which are essential for a workflow of academic research. In addition to your documents you have to manage external files like images or bibliography.

For that, many authors rely on Word Processors like Microsoft Word or Google Docs. Alternatively you could use an open source variant like LibreOffice Writer, but it is not commonly used as Word. So .odt format is not often used and sometimes .docx files are not rendered correctly in Writer.
In a group project this is much more frustrating. When working in collaboration with others the lack of organizational management for your project hinders your overall workflow. Tracking changes is very difficult. Different people would work on their part by creating different copies of the initial file. Even if Word has its Track Changes feature, it is still linear to one document. The different logs of changes are not connected to each other. Exchanging and merging these documents at the end very time consuming.

Because of those downsides of Word other authors like to use LaTeX . Essentially it is a markup language. The content is written in plain text and can be annotated with specific commands to describe how an element should be displayed in PDF file. Content and form are separated. So you can focus on your writing without breaking your workflow.
The whole process of writing can be managed through Version Control Systems like git. So logging and merging are tightly managed with the project.

Unfortunately it has a steep learning curve and thus is not very easy and intuitive for beginners. Sometimes you might have to work with someone who does not want to learn LaTeX . This is where Markdown shines. It has the same benefits of LaTeX , but its readability is much higher and it is easier to learn than LaTeX .

Markdown by itself is just a markup language and originally designed to generate HTML. With the help of Pandoc we can write professional LaTeX files converting it from Markdown.

LaTeX vs Markdown

A good comparison is provided in a table by Dominici in his paper: (Dominici 2014).

Pandoc

Pandoc is a command line tool created by John MacFarlane (MacFarlane 2013) in 2006. It is an universal converter for different document formats. It can take 28 input formats and can generate files for 45 output formats.

Software Requirements

Text editor: Any text editor will do. Be it Notepad++ for Windows or TextWrangler for MacOS. Multiplatform editors which support many kinds of syntaxes (Markdown too) are Sublime or Atom.

Command line Terminal: Pandoc is a command line tool. So it needs a terminal where it can be executed from.

Pandoc: For installing Pandoc just follow https://pandoc.org/installing.html.

LaTeX : For PDF output LaTeX has to be installed. (Example TexLive)

Command line usage

To use Pandoc you have to open in you terminal and execute it like any other command line program.

```
pandoc input.md -o output.pdf
```

Metadata for document

Every document written in Pandoc's Markdown needs a metadata block written in YAML. It contains informations like `title`, `author`, `date`.

```
---
title: Sample Title
author: Author Name
date: 01.01.2000
header-includes:
  - \usepackage{geometry}
---
```

Markdown extension

Besides basic Markdown Pandoc extends it further by adding the possibility of using raw Tex annotations. With that you can add TEX Math or more detailed elements which will not be generated by Pandoc.

Footnotes and citations are also supported. For citations the external filter `pandoc-citeproc` should be used.

```
pandoc --filter pandoc-citeproc input.md -o output.pdf
```

The `bib`-file is either specified in your metadata block or given to Pandoc as an argument.

Pandoc templates and filters

Templates and filters are the two most interesting features of Pandoc.

Pandoc uses for its PDF generation its default template. To create your own custom template you have to execute `pandoc -D latex > custom_template.tex`. `custom_template.tex` contains the default template which than can be changed according to your need.

Pandoc provides its users an interface for writing programs to manipulate documents while converting them. Those programs are called filters and they make us of the AST – "abstract syntax tree" – representation of the document. To use a filter you have to specify it on the command line with the `--filter`-flag.

```
pandoc --filter myfilter.py input.md -o output.pdf
```

Limitations of Pandoc and Markdown

Even if Pandoc and Markdown together seem very powerful, there still are some problems and limitations.

Tables generated by Pandoc are very simple. Lines can not be added between rows or columns. You can not span multiple cells of rows and columns. For that you would have to revert back to LATEX tables which would defeat the purpose of using Markdown as alternative for LATEX .

LATEX and Markdown can not be nested together. The following example would not work as expected. That block would be recognized as a whole LATEX block and everything inside the `center`-environment would then be considered as a normal paragraph and string.

```
\begin{center}
  - item 1
```

```
  - item 2
\end{center}
```

TEX Math is limited to inline or displayed expressions. Other math environments are not included in Pandoc generation.

Conclusion

Pandoc and its Markdown can be very useful when working on project requiring multiple output formats. It can be seen as a neutral way between Word and LATEX . Thanks to choosing as Markdown as initial format it is easy to read and write for everyone. LATEX users can still use some Math and other TEX magic.

As powerful as Pandoc is it still has its limitations. But if one is aware of them and maybe creatively solves them with filters, there are no problems with most projects.

References

Dominici, Massimiliano. 2014. "An Overview of Pandoc." *TUGboat* 35 (1): 44–50. https://www.tug.org/TUGboat/tb35-1/tb109dominici.pdf.

Gruber, John. 2004. "Daring Fireball: Markdown." https://daringfireball.net/projects/markdown/.

MacFarlane, John. 2013. "Pandoc: A Universal Document Converter." https://pandoc.org.