MDtoLongPDF

Table of Contents

About The Project

Pagination in PDF is deprecated. The vast majority of PDF documents nowadays were never intended to be nor will ever be printed. Yet, page breaks continue to split sections, break tables, move figures around leaving huge chunks of empty space, etc, all to serve a function that is no longer needed.

So, this script aims to provide a workaround when converting from inherently unpaginated formats, namely, Markdown and HTML into PDF by putting all of their rendered content onto a single very long PDF page. See README.pdf for an example.

Getting Started

Prerequisites

- Python 3.x (tested on Python 3.9.7)
  - `numpy`
  - `pdfminer`
- `pandoc` for md -> html
- `wkhtmltopdf` for html -> pdf
- `qpdf` to clean up after `wkhtmltopdf`

Installation

Below is an example of how you can instruct your audience on installing and setting up your app. This template doesn't rely on any external dependencies or services.

1. Install `pandoc`, `wkhtmltopdf`, and `qpdf`
2. Clone the repo
   `sh git clone https://github.com/Breedoon/MDtoLongPDF.git`
3. Install `requirements.txt`:

```
pip install -r requirements.txt
```

Usage

1. To use the script, run:

```
python3 md2longpdf.py
```

2. Then you will be prompted to enter path to you `.md` or `.html` file, for example:

```
Enter location of your md/html file (e.g.:
/Users/breedoon/Drive/CS166/Assignment 1.md)
>? README.md
```

3. Then you will be asked where the PDF should be generated (or leave empty to use the same folder and name as the input file):

```
Paste path to the output pdf file (will be overridden if exists)
or leave blank to generate in the same folder with the same name
>? resources/README.pdf
```

4. Done! Find your PDF: resources/README.pdf

Default Parameters

The stylesheet used to generate HTML is resources/pandoc.css based on this stylesheet. The simplest way to change scale is to modify `html { font-size: 120%; ... }` and `.math { zoom: 120%; }`.

The default PDF margins used are 15mm from each side, and can be changed in modules.py in the parameters of `HTMLtoPDF` class.

Known Issues

Rendering LaTeX

`wkhtmltopdf` is the only html-to-pdf tool I found that allows to specify arbitrarily large page sizes, which is necessary. A disadvantage of `wkhtmltopdf` is that I couldn't get it to compile MathJax, or MathML, or KaTeX, or any other pandoc-supported way to export LaTeX into HTML. So, the only working solution was to render LaTeX into SVGs with latex.codecogs.com which did slow things down (~2 seconds per equation compared to 3-5 seconds required for the whole process) but was still working fairly well.

A possible alternative would be to use GladTeX, also supported by `pandoc` and thus generate SVGs locally, but GladTeX is a pain to set up so I didn't even bother.

Rendering directly from markdown into LaTeX with `pandoc` allowed for paper size to specified as `-V geometry:paperheight=1000mm,paperwidth=210mm,margin=0pt` but that way, the footnotes appear at the bottom of the page which makes calculating the most optimal page size an unreliable mess.

Wikilinks

At the moment of writing, pandoc still does not have support for markdown [[wikilinks]] so I had to write a short script of my own to preprocess my markdown files containing wikilinks, which might not work in some edge cases (e.g., double [[]] inside of code blocks, etc), though it has worked well for most of my use cases.

`pandoc` will likely soon implement wikilinks, after which the entire `MdToMdWithoutWikilinks` part of the script will be unnecessary and the pandoc command will need to be modified to include `pandoc ... -f markdown+wikilinks_title_after_pipe ...`.