1 Introduction to MD2PDF-Batch Project

This project aims to make the workflow to write a batch of manuscripts / reports with a same template easier.

- Write in markdowns, get papers. Help maintaining a clean paper repository.
- It will make a latex project easy to read:
  - separate templates and contents
  - itemize, emph, headers... expressed in Markdown!
  - future work: easier figures / subfigures, and tables.
- Compatibility to latex:
  - You can use any latex environments in our markdown file.
- Support batch workflow easier by input arguments in a list:
  - You can specify input arguments in the configuration file, either applied to all, or to a single document. These arguments are passed into your LaTeX compiler — this gives flexibility in the batch process!

1.1 Workflow description

1. Prepare template file in `src/`. It should always only include a temp.tex file in the document:

```
\begin{document}
\input{temp}
\end{document}
```

   - For any arguments you want to specify in configuration file rather than fixed in template, just write `\SomeName` inside template, and specify `Somename` in configuration list.

2. Specify your configuration list in `list.yml`, within Yaml format.
   - For arguments that apply globally (to all sub-documents), specify it in `default` object.
   - For arguments that apply to a single document, specify it in the document id object, e.g. `hw0`.
   - You can add a cover page by specifying *coverpage* argument.
3. Put all your source files in `src/`. The filename should be always `DocID.md` for system to recognize, e.g. `hw0.md`.

4. Run `python configure.py` to compile a Makefile.

5. Run `make all` to make all documents into `build` directory, or make a single document regarding to the ID in list.yml, e.g. `make hw0` to make `build/hw0.pdf`

1.2 Dependencies

- pandoc
- makefile
- a latex compiler

2 Source File Format

2.1 Getting Started

Be sure to look into this markdown file to learn use cases of markdown.
Paragraphs are split by an empty line.
Section headers can be specified:

- section: `====` or `#`
- subsection: `----` or `##`
- subsubsection: `###`

A good thing compared to LaTeXis that you can directly use quotation marks: `"`: “Hello world”.
Anything that begins with a `\` will have identical function with in LaTeX.
Other useful formats:

- **Strong**: `**Strong**`
- *Emphasis*: `*Emphasis*`
- TEXTSC: `\textsc{TextSc}`
- Comment: `<!-- Something you want to comment -->`

2.1.1 A sample unordered list

Be sure to have 1 empty line before the list.

- Fruit
  - Apple (Be sure to use 4 spaces for each indent!!!)
  - Orange (either $\star$ or $-$ is OK)
  - Strawberry
- People
  - Alice
  - Barack
  - Cathy

2.1.2 A Sample ordered list

1. Fruit
   (a) Apple (Be sure to use 4 spaces for each indent!!!)
   (b) Orange
   (c) Strawberry
2. People
   (a) Alice (`#.` also gives ordered list. This is useful when you do not include *enumerate* package.)
   (b) Barack
   (c) Cathy
3. OtherPeople
   1. Xlice
   2. Yarack
   3. Zathy

2.2 Environments

Any LaTeXenvironment is supported. You can directly write it in MarkDown.
e.g. `\begin{someEnvironment}...\end{someEnvironment}`

Table 1: Some table

Name	Some number	Some other number
Alice	5.5307	5.5576
Bob	5.5305	4.8091
Cathy	5.5284	15.8686

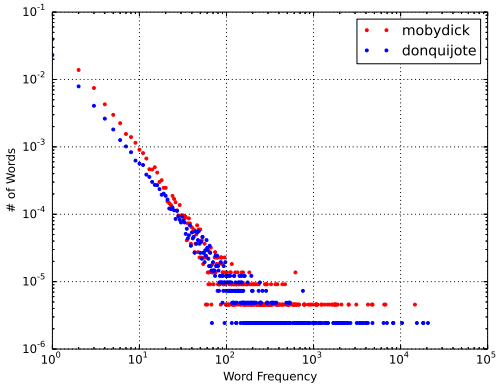


Figure 1: Sample Figure

2.2.1 Tabular

Just use latex environment `\begin{tabular}{lr} ... \end{tabular}` and you will get the tabular.

(a) Number of nodes in the network	7115
(b) Number of nodes with a self-edge	0
(c) Number of directed edges in the network	103689

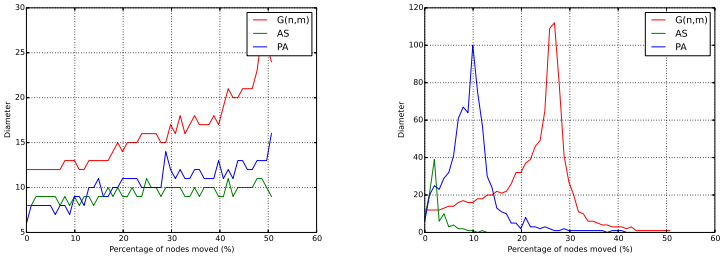
2.2.2 Tables and Figures

You have to specify and refer to tables and figures as you do in LaTeX. You can refer to Table 1, Figure 1: `Table \ref{table:1}, Figure \ref{fig:1}`.

You can use and refer to subfigures like following: Figure 2(a): `Figure \ref{fig:sub1}`. Be sure to include have specific packages.

2.2.3 Code Blocks with specific language

```
#!/usr/bin/python
print 'Hello World!'
```



(a) Some Subfigure (b) Some Other subfigure

Figure 2: Sample figure with subfigures

```
lan = 'Python'
print 'This is a sample code block in %s' % lan!
```

You can use environment `\begin{lstlisting}[language=python]...`, and craft your python/C++/Matlab code blocks...

2.3 Bibliography

We believe that it is not common that you include bibs in this batch project.

Be aware that our generated Makefile does not include a bibtex command. You can try to modify *configure.py* if you want. Note that the proper process should be *pdflatex, bibtex, then pdflatex twice*.

If you really want this, please send me a note. Actually we will have bib support in the coming non-batch version of our workflow.

Acknoledgement

Big thanks to Winnie Liu, for coming out the ideas and initial efforts!

Thanks for Scott Cheng for pushing this project for release.