Quarto template for writing academic papers

Hai-Lin Wang[a,b,*]

[a]*Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China,*

[b]*Department of Geotechnical Engineering, College of Civil Engineering, Tongji University, Shanghai, China,*

Abstract

The quarto-template is a Quarto template for writing academic papers. The template is based on the Quarto document system, which is a document system that supports the entire research lifecycle, from initial exploration to final publication. The template provides a simple and clean layout for writing academic papers, which is suitable for researchers who want to focus on the content of the paper rather than the formatting.

Keywords: Manuscript, Template

1. Features

- Write academic papers in markdown, focus on the content rather than the formatting. See Quarto documentation for more information.
- Export to html, docx, and pdf formats using the Quarto document system (`make paper`).
- Make releases with markdown, tex, docx, and pdf documents archived (`make release-<tag>`). Generate a diff file compared with a previous release using latexdiff (`make diff previous=<previous-tag> current=<current-tag>`).
- Publish the HTML version of the paper to the web using GitHub Pages.

2. Prerequisites

To use the template, you need to have the Quarto document system installed on your computer. You can install Quarto by following the instructions on the Quarto website.

To render the paper to PDF, you also need to have a LaTeX distribution installed on your computer. You can install LaTeX by following the instructions on the LaTeX website. You can also use the tinytex package for a lightweight LaTeX distribution:

---

[*]Corresponding author

Email address: `hailin.wang@connect.polyu.hk` (Hai-Lin Wang)

```
quarto install tinytex
```

See [PDF Engines in Quarto](#) for more information.

The GNU Make utility is required to run the commands in the `Makefile`. You can install GNU Make by following the instructions on the [GNU Make website](#). If you are using Windows, you can install GNU Make from the [GnuWin32](#) project.

To clone the repository and run the commands in the `Makefile`, you need to have Git installed on your computer. You can install Git by following the instructions on the [Git website](#). If you prefer to use a graphical user interface for Git, you can install [GitHub Desktop](#).

3. Usages

3.1. Setup the project

The template is designed to be easy to use, with minimal configuration required. To use the template, simply fork the [quarto-template](#) repository. You can then clone the repository to your local computer using Git:

```
git clone https://github.com/<your-username>/<your-repository>.git
```

Or you can simply open the repository in GitHub Desktop from the browser.

3.2. Start writing

You can now start writing your paper in the `paper.md` file. For more information on how to write papers using the Quarto document system, please refer to the [Quarto documentation](#).

To configure the metadata of the paper, such as the title, authors, abstract, and article template to use, you can edit the YAML front matter at the beginning of the `paper.md` file and the `_quarto.yml` file.

3.3. Render the paper

To render the paper to PDF, run the following commands in the terminal:

```
make deps   # Fetch the extensions for journal articles
make paper  # Render the paper to docx, html, and pdf files
make clean  # Clean up the intermediate files
```

3.4. Making releases

You can use the following to make a release, markdown, tex, docx, and pdf documents will be archived in the directory `releases/<tag>`:

```
make release-<tag>
```

You can also use the following command to generate a diff file compared with a previous release using latexdiff:

```
make diff previous=<previous-tag> current=<current-tag>
```

3.5. Commit and push

After writing the paper, commit and push the changes to your repository. You can then share the link to the repository with your collaborators or submit the paper to a journal for publication.

A html version of the paper will be published to the `gh-pages` branch of the repository after every commit. You can turn on the GitHub Pages feature in the repository settings to publish the html version of the paper to the web. You can then view the paper online by visiting the link provided in the repository settings.

[!Note] You need to turn on the **Read and write permissions** for the **Actions** in the **Settings** of your repository to grant the permission for the GitHub Actions to upload the rendered files.

References