count-para

Michael Cysouw

Adding a paragraph counter to text

[1] **Summary**: Use this Pandoc filter to add numbers to paragraphs of a text. These numbers can be used as stable identifiers for scholarly citations in an online world of adaptive design where page numbers are not useful anymore.

- Check out this readme in HTML and PDF.
- Refer to paragraphs by using a hash, e.g. (Cysouw 2021: #9), and add this hashtag to the end of a link to be automatically directed to the paragraph, e.g. link to the ninth paragraph of this readme like this: (Cysouw 2021: #9).
- There is explicitly no option to hide these identifiers in the final output, because the numbers are supposed to be 'hard-coded' into the text for them to function as stable identifiers.

Rationale

[2] Even scientific writing, notwithstanding its very conservative nature, is moving to new forms of publication. One of the ongoing changes is the appearance of a diverse set of **electronic forms of publication**. Articles and books are all (coming) online, and many are also available as html. Also ePub is clearly an avenue to consider. In general, any such new formats will have to be adaptive to different screen sizes.

[3] One of the central problems this poses for academia (besides the obvious infrastructure issues) is that one of the central scholarly traditions breaks down, namely the **reference to a page number** of a work. Currently, there is a strong tendency for online publications to always *also* make a PDF available that looks like a classical printed book with page layout. The main reason for these PDF-versions is that they still provide an authoritative separation of the work into pages for reference (less and less readers are actually printing them).

[4] It is time to move to a different approach to citation. Figures and tables can be numbered, and of course there are numbered sections. However, for a more fine-grained approach I propose to simply number **paragraphs**. A paragraph is a clearly defined part of a text, and it remains the same paragraph on different screen sizes with different wrappings (this does not hold, for example, for line

numbers). A good written paragraph has a clear message, so for precise citation this seems like an ideal match (and ranges of paragraphs can of course also be used).

[5] The idea is to simply put a number at the start of each paragraph, hard coded into the text so it cannot be changed (it has to be a **stable identifier**!). Depending on the electronic format, different ways to actually display these identifiers can be used. Being hard coded at the start of a paragraph does not necessarily mean that they should be obtrusive!

Practice

[6] There are many ways in practice to add the numbers. I provide here one possibility, a Pandoc Lua filter. Pandoc is a conversion system, mainly provided by John MacFarlane, that allows for a highly flexible conversion between many different output formats. The basic idea is to write text in Pandoc's extension of Markdown, and then the text can be published into different formats. Pandoc has an extension mechanism called 'filters', and the easiest to use variant are 'lua filters'. They basically consist of just one file that provides the extra functionality.

[7] The file `count-para.lua` in this repository is such a file. Used with Pandoc it will count paragraphs, add a number to the front, and provides (currently) nice outputs for HTML and Latex. As an example, this readme-document is provided with paragraph-numbering in HTML and as a PDF made with Latex by the following commands (much of this can be specified in so-called `default` files, which are much easier to handle):

```
pandoc readme.md \
    --to html \
    --output docs/readme.html \
    --lua-filter count-para.lua \
    --standalone

pandoc readme.md \
    --to pdf \
    --output docs/readme.pdf \
    --lua-filter count-para.lua
```

[8] For a more involved example, check out my manuscript about German grammar written in Pandoc Markdown. The HTML version can be directly accessed here.

Referencing

[9] Now we can refer to paragraphs! Of course we can simple write them in our citations, just like page numbers. I propose to use the hash '#' as an indicator of paragraphs, like this (Cysouw 2021: #2.7), to differentiate them from page numbers, like this (Cysouw 2021: 34). This hash is not just a typographic issue, but it is actually the method to link to the paragraph in question. If you add

#2.7 to the end of the link to my manuscript you will immediately end up at that paragraph, e.g. click here: (Cysouw 2021: #2.7). By the way, the number is 2.7 because I have restarted numbering at chapters. So this is actually the seventh paragraph of chapter 2.

In-document referencing

[10] It is also possible to refer to a paragraph inside your own document. Simply add a reference-label to the start of a paragraph that looks like this: `{#test}`. As you can see in this readme (in the GitHub version), there is such a reference at the start of this paragraph. Now you can refer to this paragraph by using the Pandoc citation format that looks like this `[@test]`. After Pandoc processing, it then looks like a reference to paragraph 10 on page 3 (this reference is replace by a number and hyperlinked after processing with Pandoc).

Options

- **`resetAtChapter`** To restart numbering this filter provides an option: by specifying `resetAtChapter: true` in the metadata for Pandoc (for an example, see below) paragraph-numbers will restart each chapter and a chapter number is added. The term 'chapter' is a slight misnomer, because it simply refers to the highest level of headings in the manuscript. Note that the chapter numbers are also added when there are no explicit chapter numbers.
- **`chapterSep`** By default, the chapter-number is separated by the running number by a full stop, e.g. `3.45`. Use this option to specify a different string to separate chapter and example number.
- **`enclosing`** By default, the numbers are enclosed in square brackets. This option allows for other enclosures. Typically, a sequence of two characters is provided, an opening and a closing character, e.g `"()"` or `"[]"`. When a single character is provided, this is reused, e.g `"|"`. Absence of any enclosure is achieved by providing an empty string, i.e `""`.
- **`refName`** How should in-document references by called? By default the string "`paragraph`" (with a space) is inserted in-text before a paragraph-reference number. But you could for example set this to simply `refName: "#"` to get a hash without space before the number. Or you can leave it empty and just type whatever you want before the number.
- **`addPageNr`** is by default set to `addPageNr: true`. This option is only relevant for page-based formats and it will add "on page X" after the paragraph number. Currently only implemented for latex.

Issues

[11] Because of the usage of the Pandoc Cite-format, this filter has to be applied before the `citeproc` filter that processes the literature references.

`---`

```
title: count-para
author: Michael Cysouw
resetAtChapter: false
enclosing: "[]"
chapterSep: "."
refName: "paragraph "
addPageNr: true
---
```