

Tools for Writing Conversation Analytic Papers

Saul Albert (saul.albert@gmail.com)

Contents

1 The Problem	2
2 The Simplest Solution: Markdown + Pandoc	2
3 For Longer Texts: Markdown + Pandoc + LaTeX	5
3.1 What you will need	6
3.1.1 A little explanation	6
3.2 A working example	8
3.3 Rendering your CA extracts using Pandoc	12
3.3.1 PS: Testing your Pandoc + Markdown setup:	12

NB: This is a markdown version of a [blog post](#) - it's slightly different from that page as it's intended to be a working example of how to use Markdown to write CA-added research papers. Some bits may render badly as PDF (they were intended for a web layout), but the examples should at least work.

A while back I wrote a [blog post](#) detailing why I chose Pandoc and Markdown to write papers including [Jeffersonian Conversation Analytic transcripts](#). It wasn't very detailed though, because a full explanation of how to set up [a compatible text-based writing workflow](#) was an onerous task - one happily now completed beautifully by Dennis Tenen and Grant Wythoff's guide to [Sustainable Authorship in Plain Text using Pandoc and Markdown](#).

So, I decided to update this how-to for anyone using Pandoc and Markdown to start including CA style transcriptions quickly and easily.

To go along with this how-to, there is also a [set of demo files](#) you can download to try out this approach. However, before you do that you probably want to [get a pandoc + markdown setup installed](#).

1 The Problem

There are great software tools out there for [CA-style transcription](#), my favourite is [CLAN](#) for [a number of reasons](#). However, I can't find any resources online about how to publish CA-style transcriptions without being [forced through some eye-bleeding LaTeX diddling](#) every time.

Of course I could just use a WYSIWYG text editor like LibreOffice - but now I've experienced [the power of LaTeX](#) for document preparation and publication, I really can't see myself going back.

When doing CA it seems particularly important to have transcriptions legibly in the body of the paper and visible during the writing process, because many of the analytical observations come, or get significantly modified at the point of writing about them, double and triple checking assumptions, and cross-referencing with the CA literature while tweaking citations.

2 The Simplest Solution: Markdown + Pandoc

[Markdown](#) is my favourite [lightweight markup language](#), a highly readable format with which you can write a visually pleasing text file, which you can then convert into almost any other format - HTML, OpenOffice, LaTeX, RTF, etc. using [Pandoc](#). There are many similar systems, notably [reStructuredText](#) and [Textile](#), all of which you can use to write your text file, and [other conversion tools/toolsets](#), but in my experience, Markdown and Pandoc are the most useful combination in an academic context [ref]Not all of these systems support bibliographical references with [BibTeX](#) - Markdown + Pandoc does this quite elegantly[/ref].

There are lots of great things about markdown:

- Just edit simple text files - no weird file formats to get corrupted or mangled.
- Less verbose and complicated-looking than LaTeX.
- Small files are easy to share/collaborate on with others (everyone gets to use their favourite editor).
- There are some great [pandoc plugins](#) for my favourite text editor [vim](#).

However, the best thing is that, used along with the [XeTeX](#) typesetting engine, it solves the problem with CA transcriptions being unreadable in LaTeX/pdflatex.

For example, in my first CA-laced paper, my transcriptions looked like this in my LaTeX source:

```

\begin{table*}[!ht]
\hfill{}
\texttt{
  \begin{tabular}{@{}p{2mm}p{2mm}p{150mm}@{}}
  & D: & \& 0:h (I k-)= \\
  & A: & \& =Dz that make any sense to you? \\
  & C: & \& Mn mh. I don' even know who she is. \\
  & A: & \& She's that's, the Sister Kerrida, \hspace{.3mm} who, \\
  & D: & \& \hspace{76mm}\raisebox{0pt}[0pt][0pt]{ \raisebox{2.5mm}{[ ]}'hhh \\
  & D: & \& Oh \underline{that's} the one you to:ld me you bou:ght.= \\
  & C: & \& \hspace{2mm}\raisebox{0pt}[0pt][0pt]{ \raisebox{2.5mm}{[ ]} Oh-- \hspace{42mm}\raisebox{0pt}[0pt][0pt]{ \underline{Ye:h} \\
  & A: & \& \hspace{60.2mm}\raisebox{0pt}[0pt][0pt]{ \raisebox{3.1mm}{\lfloor}}\underline{Ye:h} \\
  \end{tabular}
\hfill{}
}
\caption{ Evaluation of a new artwork from (JS:I. -1) \cite[p.78]{Pomerantz1984} .}
\label{ohprefix}
\end{table*}

```

which renders this:

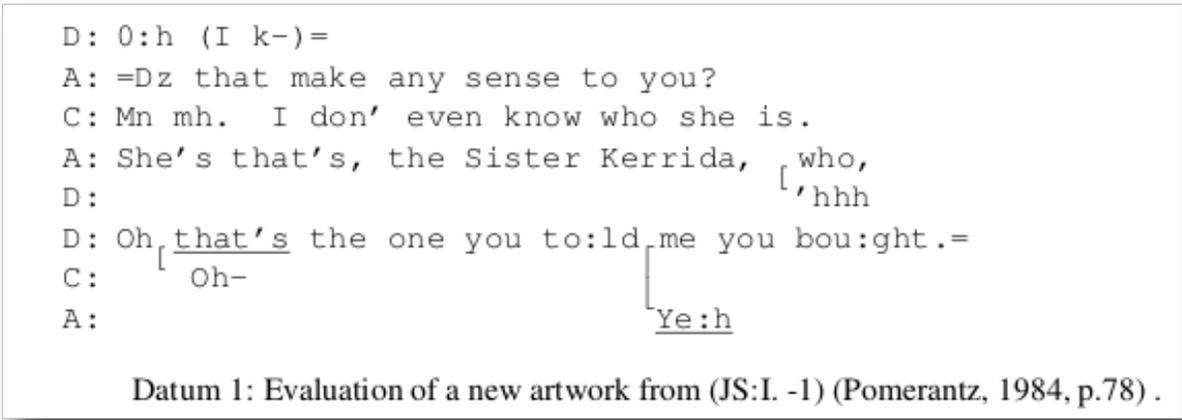


Figure 1: Nice LaTeX rendering

In Markdown for my latest paper, the CA bits look like this:

(3)

STE: Uo:h oh ugly things [he paints.]

KAT: [Really?]

(3.0)

STE: (°I think s[o-])°

KAT: [So you wouldn't sell any?]

STE: Uuh n[o]

KAT: [No?]

(1.7)

which renders this:

```
(3)

STE:  Uo:h oh ugly things [he paints.]
KAT:  [Really?]
      (3.0)
STE:  (°I think s[o-])°
KAT:  [So you wouldn't sell any?]
STE:  Uuh n[o]
KAT:  [No?]
      (1.7)
```

Figure 2: Reasonable Markdown rendering

Overall, I think the Markdown version represents a significant improvement in legibility while writing. I think it might be possible to do the same in LaTeX using the `{verbatim}` environment, but the fact that Markdown also lets me concentrate on writing without throwing errors or refusing to compile lets me spend longer on the writing than on endless text-fiddling procrastination.

When it comes to rendering, I feed my markdown file to pandoc (command all on one line).

```
$ pandoc --latex-engine xelatex --bibliography library.bib --csl default.csl -N  
-o paper_title.pdf paper_title.markdown
```

If you want to use the nicely stretched [ceiling characters](#) for overlap marking, or the raised full stop / [bullet operator](#) for inbreaths, you can do so, but you'll need to run Pandoc (see below) referencing a font that has those characters. For example, you could use [CAfont](#) and add:

```
--variable monofont=CAfont
```

to the pandoc command above.

The default.csl file is a [citation style language](#) file to customise how bibliographical references are rendered.

If you're only adding a few examples to your document, this will probably work fine. If you are writing a thesis or a longer document - read on.

3 For Longer Texts: Markdown + Pandoc + LaTeX

The above approach may work for writing a short paper with one or two examples, for a thesis or a longer piece where you may have many examples, you're going to have to take this a step further and use some LaTeX within your Markdown document. The bad news, you will have to use LaTeX, templates and some code to deal with:

1. Example Layout: you probably want your examples to be graphically separated from your text in a consistent way.
2. Document layout: you may need to make some stylistic tweaks to how your document prints out.
3. Referencing: you will want to use labels for your examples so you can cross-reference them automatically within the text and not have to re-label them every time you make a change.
4. Audio/video links: you may want to include links to audio/video examples in your files.

The good news: your CA transcript examples will still be easy to read/edit, and actually this is all pretty straight forward once you've got it set up.

3.1 What you will need

First, you need a working [Pandoc + Markdown setup installed](#). You also need a nice monospaced font installed - I use [CAfont](#) by the amazing [CHILDES](#) project.

I've made a [downloadable archive](#) of the three files I use every time I create a new document. Download those. There is also a working demo (README.md) and some image files that you can use to edit/test things, or modify them to create your own.

Along with these examples inside the `camarkdown_files` folder you will find:

- `template.txt`: a LaTeX template that Pandoc uses when it renders PDFs - with macros etc.
- `apa.csl`: a [citation style language](#) file describing how I want my APA citations rendered.
- `margins.sty`: a little margins file I can use to tweak the overall page layout separately (US Letter vs. A4 etc.)

Whenever you start a new document, these three files into the same folder.

3.1.1 A little explanation

Without getting too geeky about it, here's a little explanation of how I use this setup:

Whenever I convert my Markdown to PDF using Pandoc, I add:

```
--template template.txt
```

to the `pandoc` command to make sure it uses this template. The template is based on the default LaTeX template Pandoc always uses to convert Markdown to PDF via LaTeX, but I've added a macro: `caextract`.

Basically the `caextract` environment sets the default monospaced font, and (optionally) creates a to an online media file referenced in the Markdown file (see working example below), it also formats the paragraph containing the example as a framed float to divide it from the body of the text, and changes the listings name to 'Extract', so references list it as 'Extract 1' rather than 'Figure 1'.

Here's the relevant bits from the header section of `template.txt`

```
\newcommand{\medialink}[2] { \begin{flushright} \href{#1}{#2}\ \end{flushright}
}
@if(highlighting-macros)$
$highlighting-macros$
$endif$
@if(verbatim-in-note)$
\usepackage{fancyvrb}
$endif$
\usepackage{listings}
\lstnewenvironment{extract}[1][]{
```

```
\renewcommand*{\lstlistingname}{Extract}
\lstset{frame=single,basicstyle=\small\ttfamily,keepspaces=true,#1}
}}
```

And this bit goes into the main section of the template:

```
\usepackage{float}
\floatstyle{ruled}
\newfloat{caextract}{htp}{lop}
\floatname{caextract}{Extract}
```

3.2 A working example

Here is a full example from a paper I'm writing at the moment that you can tweak and play with. It's all done in simple markdown, using a little bit of LaTeX embedded within the Markdown file to call the macro.

```
![Stopping postures \label{stopping-postures}](images/stopping-postures.png)
\ begin{caextract}[H]
\ caption{See https://www.dropbox.com/s/jnpf5pnxcy4dg8m/lexical-features.mov}
\ label{lexical-features}
\ begin{small}
\ begin{verbatim}

1 JIM:  .hhh 「opps sorry Hh hyeh °hyour head°, .HHh Hmhmhmhmhm hehheh
2 TEA:      LYE::AH! KAY >>LET's TRY it AGAIN< FIVE, (.) sfix? (.)
3 TEA:  ↓「five six se::vlen eight? Rock st「ep. (.) tri:ple, (.) tri:ple. 1
4 JIM:    「°five six shh°」          「°ep (.) tri:ple, (.) tri:ple.°」]
5 TEA:    G 0 : : 0 :「 : d! Lovely「::.. (.) 1 OKA::Y!
6 JIM:      「0:hhkay:::? °Hm thmhmhmhmhm°」
7          (1.3)
8 TEA:    LETS ROTATE PA:RTNERS!

\ end{verbatim}
\ medialink{https://www.dropbox.com/s/e960eu94ji7ncn3/lexical-features.mov}{Watch}
\ end{small}
\ end{caextract}
```

NB: The example as displayed above is not quite right: there should not be spaces after the back-slashes at the beginning of each line. According to the manual, Pandoc should treat this above extract (with 4 leading spaces on each line) as a preformatted text and shouldn't interpret it as Markdown/LaTeX. For some reason I can't fathom, it's not doing that, so I've added a space after every " to be able to show this as code. See the functional version in the README.md file.

Here's the same, unescaped example as actual, should-work-right-here markdown. If you render this README.md file, it should work:

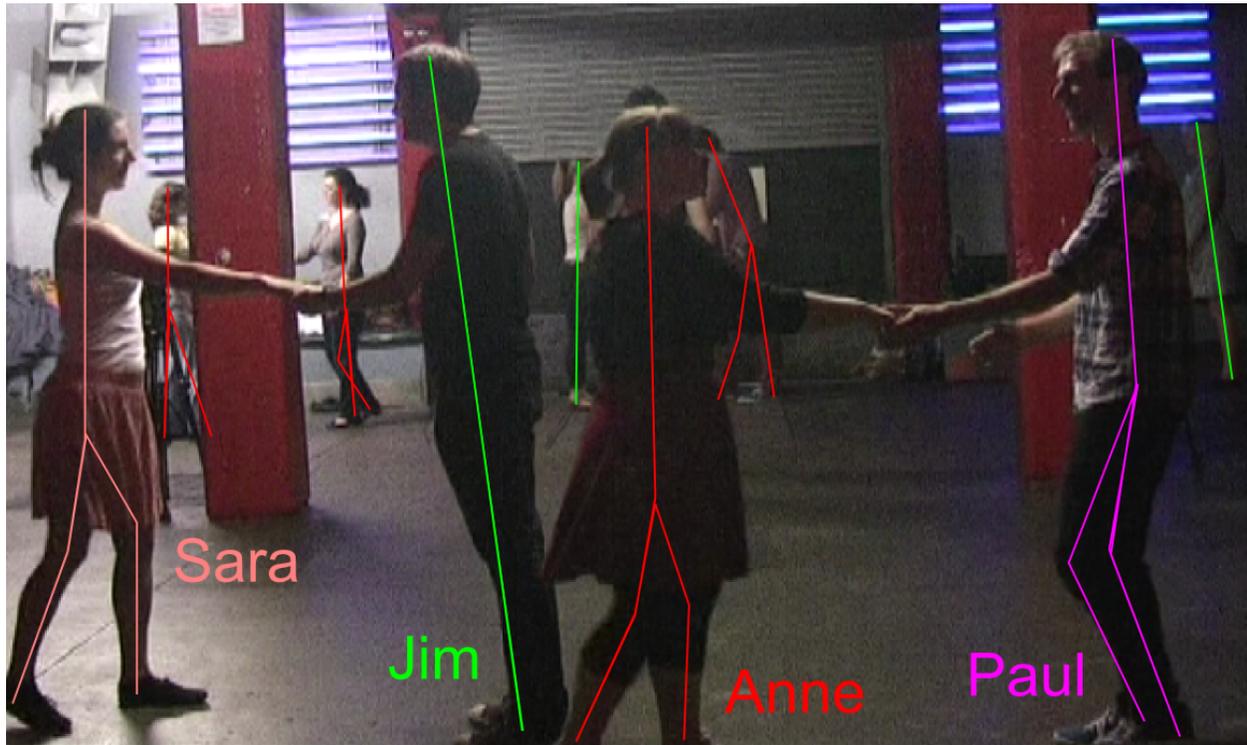


Figure 3: Different stopping postures between dancers

Extract 1 See <https://www.dropbox.com/s/jnpf5pnxcy4dg8m/lexical-features.mov>

```

1 JIM:  -hhh ʁopps sorry Hh hyeh °hyour head°, -HHh Hmhmhmhmhm hehheh
2 TEA:      ʎYE::AH! KAY >>LET's TRY it AGAIN< FIVE, (.) sʁix? (.)
3 TEA:  ʃʁfive six se::vʎen eight? Rock stʁep. (.) tri:ple, (.) tri:ple. ʎ
4 JIM:      ʎ°five six shh°ʎ      ʎ°ep (.) tri:ple, (.) tri:ple.°ʎʃ
5 TEA:  G 0 : : 0 :ʁ : d! Lovelyʁ:: (.) ʎ OKA::Y!
6 JIM:      ʎ0:hkhay:i? °Hm thmhmhmhmhm°ʎ
7          (1.3)
8 TEA:  LETS ROTATE PA:RTNERS!
```

[Watch](#)

So, for reference, that last bit should render something like this:



Figure 1: Different stopping postures between dancers

Extract 1 See <https://www.dropbox.com/s/jnfp5pnxcy4dg8m/lexical-features.mov>

```

1 JIM:  -hhh ʔopps sorry Hh hyeh ʔhyour headʔ, -HHh Hmhmmhmmhmm hehheh
2 TEA:  ʔYE::AH! KAY >>LET's TRY it AGAIN< FIVE, (.) s'fix? (.)
3 TEA:  ʔf five six se::vlen eight? Rock st'ep. (.) tri:ple, (.) tri:ple. ʔ
4 JIM:  ʔ°five six shhʔ ʔ°ep (.) tri:ple, (.) tri:ple.ʔʔ
5 TEA:  G O ʔ: ʔ ʔ: d! Lovely!:: (.) ʔ OKA::Y!
6 JIM:  ʔ0:hkhay:ʔ? ʔHm thmhmmhmmhmmʔ
7      (1.3)
8 TEA:  LETS ROTATE PA:RTNERS!

```

[Watch](#)

Figure 4: Nice render of Markdown + template

A later paragraph refers to the figure like so:

By contrast, Sara, Paul and Anne - marked in red in figure [\ref{stopping-postures}](#) - step back, split their weight and stop dancing together with the onset of Teacher's "`\verb|G 0 : : 0 : : d!`". Without having space to analyse this method, it is worth noting in closing that the regularity of these methods and their interactional contingencies are shown in the [\[slow-motion sections of the video\]](http://bit.ly/1KBn5Tz)(<http://bit.ly/1KBn5Tz>) by how dancers who stop like Jim are all pulled off balance by dancers who stop like Paul, Sara and Anne.

Here's the non-escaped version that should render normally:

By contrast, Sara, Paul and Anne - marked in red in figure [3](#) - step back, split their weight and stop dancing together with the onset of Teacher's "`G 0 : : 0 : : d!`". Without having space to analyse this method, it is worth noting in closing that the regularity of these methods and their interactional contingencies are shown in the [slow-motion sections of the video](#) by how dancers who stop like Jim are all pulled off balance by dancers who stop like Paul, Sara and Anne.

Again, for reference, it should look something like this:

By contrast, Sara, Paul and Anne - marked in red in figure [1](#) - step back, split their weight between both legs and stop dancing together with the onset of Teacher's "`G 0 : : 0 : : d!`". While these analyses are fully developed in the paper itself, it is worth noting here that the embodied interactional contingencies of these different methods of stopping dancing are observable in the [slow-motion sections of the video](#). Dancers who stop like Jim, for example, are consistently pulled off balance by dancers who stop like Paul, Sara and Anne.

Figure 5: How the paragraph should look

3.3 Rendering your CA extracts using Pandoc

Finally, making sure you have your csl file (apa.csl), your images, your template.txt file and your margins.sty file all in the same folder with your example (I find that convenient), and making sure you have a nice monospaced font to use ([CAfont is great](#)) in place, run something like this (all on one line).

```
pandoc --latex-engine xelatex --csl apa.csl --variable monofont=CAfont
--variable mainfont=Arial --variable fontsize=12pt -H margins.sty
--template template.txt --bibliography /path/to/library.bib
-o README.pdf README.md
```

You can, of course, run this command from the terminal - swapping out the relevant variables as needed, but I use vim-pandoc's PandocRegisterExecutor function to run this whenever I type the local leader character twice (,,) followed by pdf. See <https://github.com/vim-pandoc/vim-pandoc> for documentation of that kind of thing.

I'm happy to answer any questions at [this post on my blog](#) or on [@saul](#).

3.3.1 PS: Testing your Pandoc + Markdown setup:

You can test your setup by reproducing this document. The command I used to create README.pdf was:

```
pandoc --latex-engine xelatex --toc -N --toc-depth=2 --csl apa.csl
--variable monofont=CAfont --template template.txt --variable mainfont=Arial
--variable fontsize=12pt -H margins.sty -o README.pdf README.md
```

NB: I did not use any references, so the bibliography bit isn't in there.