# Problems Overview

Currently, the proposed record structure for individual white matter structures ("tractDepiction") features fields for links to sections of text ("descriptions") and links to images ("figures"). The use of links as opposed to downloading and storing local representations is to avoid copyright infringement issues, which have typically seen "hotlinking" to be within the realm of fair use (though it does get a bit complicated). This "solution", and our planned approach more broadly, are not without their problems.

## Problem 1: automated text extraction

Below we see an example of a html link to a section of a paper hosted by a Springer-owned journal

https://link.springer.com/article/10.1007/s00429-019-01907-8#:~:text=The%20posterior%20arcuate%20(pArc%3B%20Fig,Supplementary%20Fig.%C2%A03%20region%2016).

This component:

<https://link.springer.com/article/10.1007/s00429-019-01907-8>

selects the article itself by its doi (10.1007/s00429-019-01907-8), while the latter component:

# :~:text=The%20posterior%20arcuate%20(pArc%3B%20Fig,Supplementary%20Fig.%C2%A03%20region%2016).

Corresponds to a chrome extension capability (described [here](https://arstechnica.com/gadgets/2020/06/google-pushes-text-fragment-links-with-new-chrome-extension/)) which can be used to highlight specific bits of text via the url itself.

Currently, I do not know if this capability can be extended to facilitate the automated extraction of those text sections. This would be an ideal method to facilitate the curation and extraction of relevant bits of text for machine learning / text analysis purposes. At the very least it can be used to demarcate pertinent bits of text, which is at least a partial solution.

## Problem 2: images

The case for images is a bit different. It is quite possible to link directly to an image:

https://media.springernature.com/full/springer-static/image/art%3A10.1007%2Fs00429-019-01907-8/MediaObjects/429\_2019\_1907\_Fig6\_HTML.png

However, such a link is divorced from its figure description, which may be essential to interpreting the figure. One solution is to link to the standalone page provided by the journal for the figure:

https://link.springer.com/article/10.1007/s00429-019-01907-8/figures/6

This works in a sense, but may present a challenge for our attempts to provide users with a database interface which allows them to directly interact with the article components of interest, which hints at problem 3.

## Problem 3: displaying web content

It is currently unclear to what extent an interactive website based off of Juypter Notebook / Sphinx / Jupyter Book will be able meet our use case. Minimally, the python package [*itables*](https://github.com/mwouts/itables)provides some interactive and search functionality with tables which could, along with [*pandas*](https://pandas.pydata.org/), do a lot of what we are looking for.

The challenge arises when the actual information sought isn't stored locally in the database (as would be the case with the targets of our text and image links). [ipython appears to have some methods for this](https://ipython.org/ipython-doc/3/api/generated/IPython.display.html), which appear to be able to handle things like [images](https://ipython.org/ipython-doc/3/api/generated/IPython.display.html#IPython.display.Image), [html](https://ipython.org/ipython-doc/3/api/generated/IPython.display.html#IPython.display.HTML), and [JSON](https://ipython.org/ipython-doc/3/api/generated/IPython.display.html#IPython.display.JSON). Some experimentation will be necessary to see how this can take us with jupyter notebook output windows.

## Problem 4: authorization tokens

Journals typically restrict access to documents. In order to access them you would need to be authenticated as a valid user from a university (or other related organization). To my understanding this is likely achieved via IP addresses (as evidenced by the use of university credentials to access journal articles via a proxy service from the library). If this is the case, a user interacting with our resource using an IP connection from a university should be able to access the relevant html based content. However, if this process requires browser-based authentication, some sort of API or authentication token may be necessary.

# Solutions Overview

Independent of the aforementioned challenges, there also appear to be ways to automate or streamline certain parts of the "distillation" process. Below I describe some of these.

## Automated DOI based metadata extraction

Using Crossref and Datacite doi metadata APIs (as documented [here](https://project-thor.readme.io/v2.0/docs/accessing-doi-metadata)). This could be used to automatically extract many of the relevant fields for articles (e.g. everything in the current template except species, method, and of course tractDepictions) , simply based on entry of the doi. It may even be possible to enter a list of dois and have some code generate the relevant folder structure and basic "articles" excel files as described in the "ExcelFiles" readme.

## Use of Excel spreadsheets as intermediary format

For contributors that may be less familiar with JSON formatting or other programming conventions it is possible to use [pandas](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_excel.html) and associated JSON conversion functions in python to convert excel spreadsheets containing the relevant information to the desired JSON format. Indeed, even in cases where the user is familiar with JSON formatting, it may be simpler to use a template spreadsheet.

## JSON merging and convertibility

Currently, JSON format is being proposed as the storage format for tract and article information. These components can be merged both on an article and corpus level in a programmatic fashion, allowing for the construction of a composite database from individually stored records. Moreover, in the event that it is deemed appropriate, the JSON formatted files/database can be straightforwardly converted into XML format for use with SQL based applications/approaches.

## Web interface for contributions

Currently, it is presumed/proposed that contributors will log article and track information in excel spreadsheets (as described in the excel folder readme). However, it is also conceivable that an interactive webform could also be used to streamline this process. For example, the user could begin by entering the doi of a given article to auto populate an entry (as described above) and then enter the desired information via checkboxes or text entry fields. It is currently unclear if this can be done with jupyter notebooks. It may be simpler and more straightforward to use some sort of online survey platform.