Hacking OpenScholar

Kevin He

March 2019

OpenScholar is the content-management system powering scholar.harvard.edu websites. It is widely adopted by the Harvard Economics department — in particular, every job-market candidate maintains an OpenScholar site as their official department webpage. The appeal of OpenScholar stems partly from its intuitive graphical interface. But this feature comes at the expense of flexibility and control, as site owners can only indirectly modify the underlying HTML/CSS source through the intermediating layer of OpenScholar web-development tools.

This document describes several ad-hoc workarounds (or "hacks") that I discovered while building my job-market website using OpenScholar. These techniques allowed me to better control how my site looks and functions, and I hope they will be helpful for future cohorts of OpenScholar users.

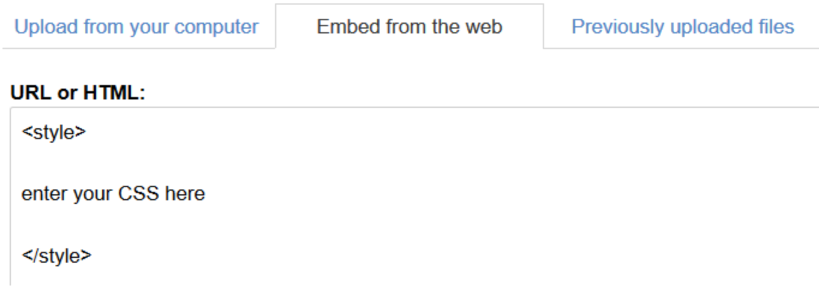
1 Custom CSS

OpenScholar lets you choose between several "themes," but does not let you to use an external CSS file. Nevertheless, the "Embed Media" widget can inject arbitrary CSS code into your page, thus allowing you to re-style any page element in any way you like.

1. Follow steps 1 through 4 here: https://help.theopenscholar.com/embed-media-widget
2. In step 5, Choose File ▷ Embed from the Web.
3. Enter your CSS code into the text box. Add the `!important` property to the end of every rule you write. For example,

   ```
   li a{ color: #8B0000 !important; }
   ```

4. The text box must start with `<style>` and end with `</style>`.

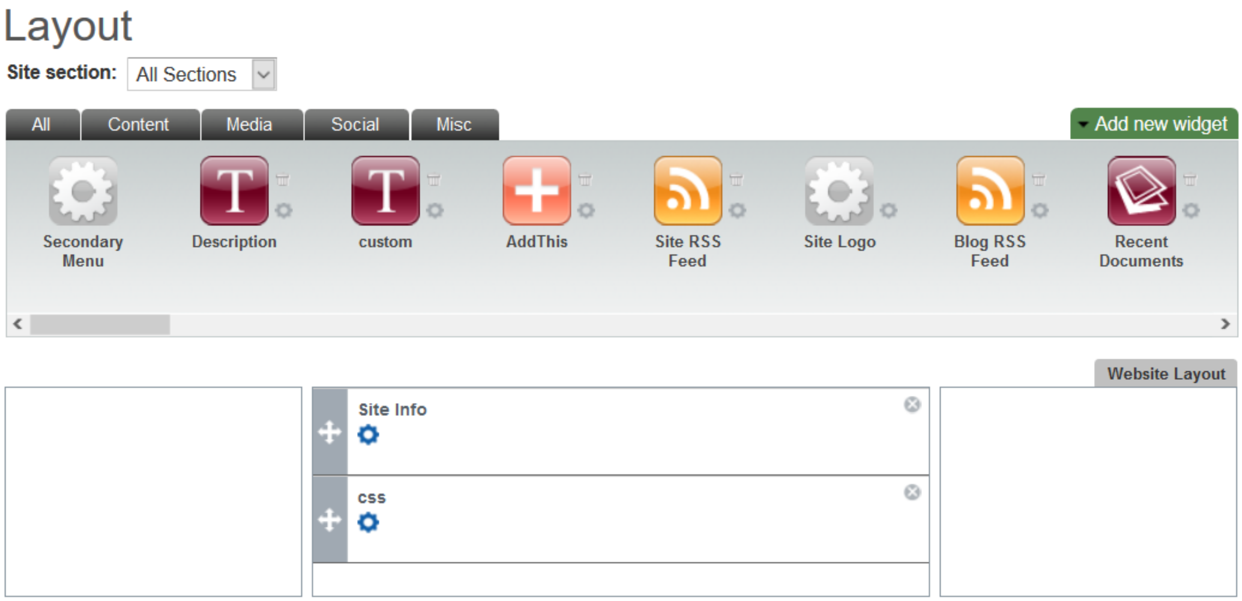


```
<style>

enter your CSS here

</style>
```

5. Submit ▷ Fill in "Name of File" ▷ Save.

6. Fill in "Widget Description" ▷ Save.

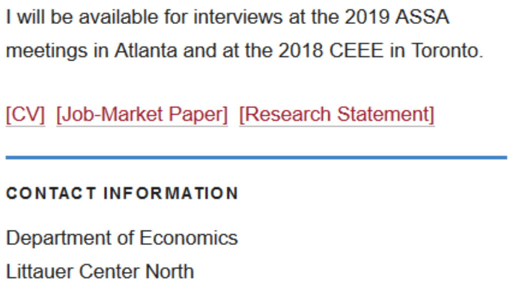
7. In Appearance ▷ Layout, place the newly created Embed Media Widget (I named mine "css") into the layout of "All Sections." (Make sure you are editing the layout of "All Sections" and not just that of "Front Page.")



The Embed Media Widget becomes part of the HTML source for all of your pages. The `<style>` tag causes web browsers to interpret the widget's content as CSS. The `!important` property allows your CSS rules to override any existing rules from OpenScholar's theme. To edit your CSS, click the "gears" button near the Embed Media Widget you created, then click the "Edit" button under "File or HTML."

2 Styling Specific Elements on Your Page: An Example

Let's say you want to change the color of this horizontal blue line on your page.



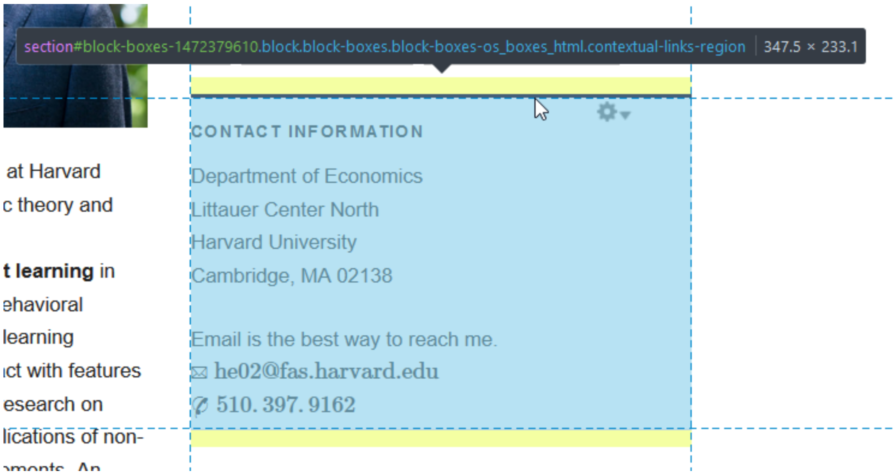
I will be available for interviews at the 2019 ASSA meetings in Atlanta and at the 2018 CEEE in Toronto.

[CV] [Job-Market Paper] [Research Statement]

CONTACT INFORMATION

Department of Economics
Littauer Center North

Modern web browsers all have an "inspect elements" function. (In Firefox, press F12.) It lets you look up the HTML source corresponding to page elements. Let's say you find that the blue line is the top border of an item with the id `block-boxes-1472379610`.



section#block-boxes-1472379610.block.block-boxes.block-boxes-os_boxes_html.contextual-links-region | 347.5 × 233.1

at Harvard
c theory and
t learning in
ehavioral
learning
ıct with features
esearch on
lications of non-

CONTACT INFORMATION

Department of Economics
Littauer Center North
Harvard University
Cambridge, MA 02138

Email is the best way to reach me.
✉ he02@fas.harvard.edu
✆ 510. 397. 9162

Now, simply add a CSS rule to change the color of this specific border.

```
section#block-boxes-1472379610{ border-top:  3px solid #e6e6e3 !important; }
```

3 Click to Show Abstract

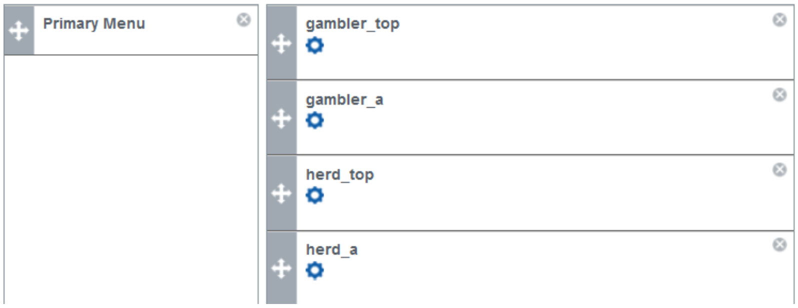
OpenScholar does not allow custom JavaScript code, making it difficult to implement a button that shows/hides a text block (e.g., the abstract of a research paper). The Embed Media hack does not work here because the platform seems to sanitize any function calls injected this way. But, OpenScholar's native Accordion Widget can be used for a decent approximation.

1. Instead of using OpenScholar's built-in "Publications" page, make an empty page called "Research" via Site Content ▷ Add ▷ Page.

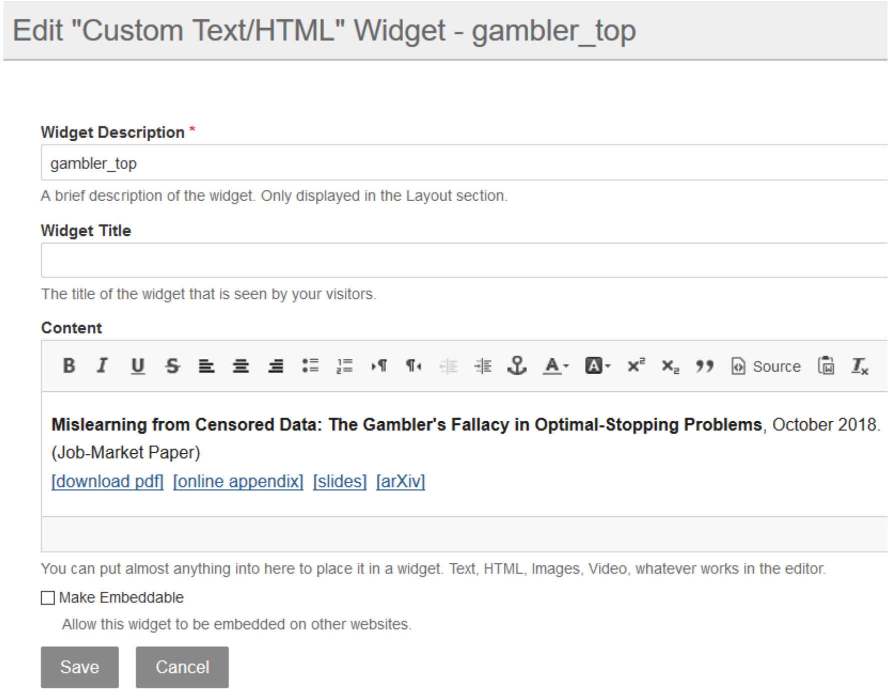
2. Navigate to this page while logged in.

3. Click the "gear" icon in the upper right part of the page, then click Layout.

Create two items on the research page layout for each paper you have written.



The first item is a Custom Text/HTML Widget, containing the paper's title and download links. These widgets are named `[paper name]_top` in the layout above.



Edit "Custom Text/HTML" Widget - gambler_top

Widget Description *

gambler_top

A brief description of the widget. Only displayed in the Layout section.

Widget Title

The title of the widget that is seen by your visitors.

Content

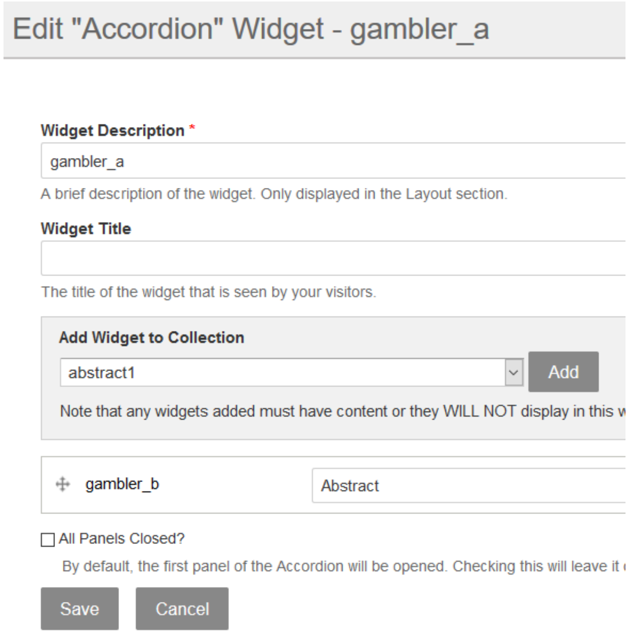
Mislearning from Censored Data: The Gambler's Fallacy in Optimal-Stopping Problems, October 2018.
(Job-Market Paper)
[download pdf] [online appendix] [slides] [arXiv]

You can put almost anything into here to place it in a widget. Text, HTML, Images, Video, whatever works in the editor.

□ Make Embeddable

Allow this widget to be embedded on other websites.

The second item is an Accordion Widget. These widgets are named [paper name]_a in the layout above.



Edit "Accordion" Widget - gambler_a

Widget Description *

gambler_a

A brief description of the widget. Only displayed in the Layout section.

Widget Title

The title of the widget that is seen by your visitors.

Add Widget to Collection

abstract1

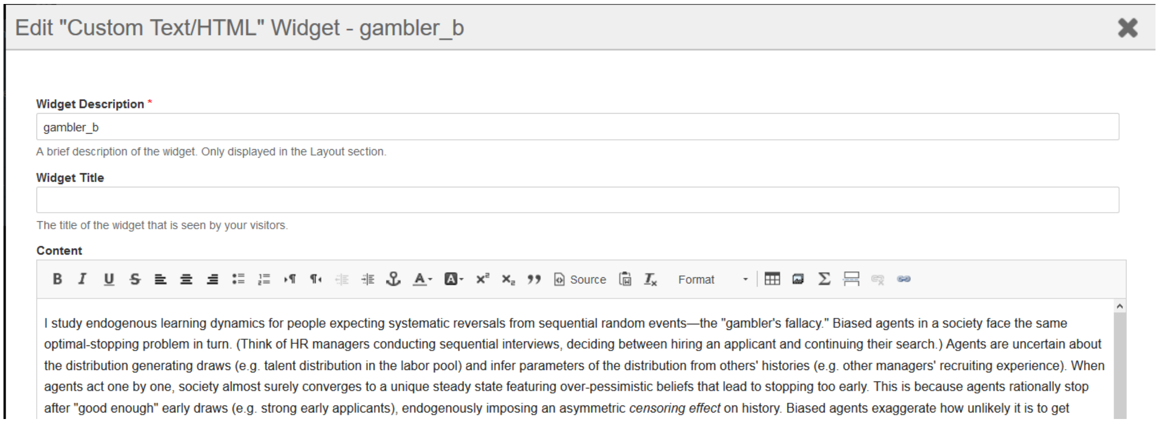
Note that any widgets added must have content or they WILL NOT display in this w

gambler_b Abstract

□ All Panels Closed?

By default, the first panel of the Accordion will be opened. Checking this will leave it

The Accordion Widget itself, in turn, contains one Custom Text/HTML Widget, which has the label "Abstract." Create this inner Custom Text/HTML Widget separately, writing the abstract of your paper in the widget. Then, add it into the Accordion Widget using "Add Widget to Collection." This embedded inner widget is named gambler_b in the screenshot above.



Edit "Custom Text/HTML" Widget - gambler_b

Widget Description *

gambler_b

A brief description of the widget. Only displayed in the Layout section.

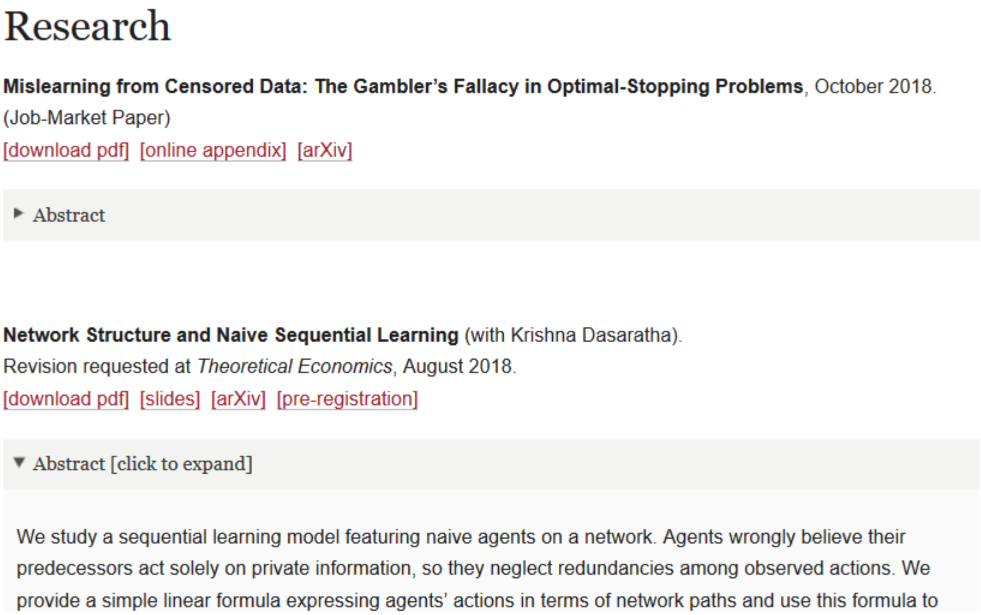
Widget Title

The title of the widget that is seen by your visitors.

Content

I study endogenous learning dynamics for people expecting systematic reversals from sequential random events—the "gambler's fallacy." Biased agents in a society face the same optimal-stopping problem in turn. (Think of HR managers conducting sequential interviews, deciding between hiring an applicant and continuing their search.) Agents are uncertain about the distribution generating draws (e.g. talent distribution in the labor pool) and infer parameters of the distribution from others' histories (e.g. other managers' recruiting experience). When agents act one by one, society almost surely converges to a unique steady state featuring over-pessimistic beliefs that lead to stopping too early. This is because agents rationally stop after "good enough" early draws (e.g. strong early applicants), endogenously imposing an asymmetric *censoring effect* on history. Biased agents exaggerate how unlikely it is to get

The net effect is that you have created an Accordion Widget containing just one item, the abstract of your paper, which can be expanded and collapsed by clicking the label "Abstract." Repeating this procedure for all of your papers produces something like this (you may wish to add a phrase like "[click to expand]" to the label inside the Accordion Widget, as to tell the site visitors that they can interact with the widget).



Research

Mislearning from Censored Data: The Gambler's Fallacy in Optimal-Stopping Problems, October 2018.
(Job-Market Paper)
[download pdf] [online appendix] [arXiv]

▸ Abstract

Network Structure and Naive Sequential Learning (with Krishna Dasaratha).
Revision requested at *Theoretical Economics*, August 2018.
[download pdf] [slides] [arXiv] [pre-registration]

▾ Abstract [click to expand]

We study a sequential learning model featuring naive agents on a network. Agents wrongly believe their predecessors act solely on private information, so they neglect redundancies among observed actions. We provide a simple linear formula expressing agents' actions in terms of network paths and use this formula to

4 MathJax as an Email-Address Obfuscation Mechanism

OpenScholar has built-in MathJax support. You can write your email address in LaTeX and render it with MathJax. For example,

```
\text{johnsmith}@\text{fas}_{\cdot}\text{harvard}_{\cdot}\text{edu}
```

becomes

$$\text{johnsmith}@\text{fas}_{\cdot}\text{harvard}_{\cdot}\text{edu}$$

The result is readable for humans but not for web-scraping bots, because bots probably can't parse LaTeX. (The use of the subscripted `\cdot` in lieu of the period further obfuscates.)