Re: [NICAR-L] Examples of computational thinking in journalism

Inbox x

Paul Bradshaw via po.missouri.edu 3:30 AM (5 hours ago)

to NICAR-L

Thanks for all those - a fantastic range of stuff. I'm going to add an update to the end of the original post with all of those examples and a few quotes from your messages, if that's OK. Here's the text - please let me know if you object to the use of any passages from this discussion (I've tried to exercise some judgement in ensuring there's nothing you wouldn't want public):

UPDATE: More examples of computational thinking

After this post was first published members of the NICAR-L mailing list suggested some other examples. Many of them are computational not just in thinking but in execution too. Here they are:

Max Lee highlighted an investigative series by **The Atlanta Journal Contitution**. "The reporters scraped through state medical board case decision websites and mined through PDF orders in order to find doctors who were found to have abused patients". You can read more about the methodology here, which involved an initial scoping phase with a legal expert (decomposition), a pubic records request phase, then scraping (pattern recognition, algorithms), and manual filtering (abstraction).

Jacqueline Kazil pointed to this Washington Post piece by **Jennifer Stark and Nicholas Diakopoulos,** on the discriminatory side-effects of Uber's surge pricing algorithm, in which they explain:

"We didn't want to miss any surges, so we chose three minutes, knowing that surges in D.C. are no shorter than three minutes. The surge-pricing data was then used to calculate the percentage of time surging. Data were analyzed by census tracts, which are geographic areas used for census tabulations, so that we could test for relationships with demographic information. Only uberX cars were included in our analysis since they are the most common type of car on Uber. (In the interest of making the data analysis transparent, all our code can be viewed online.)"

That GitHub repo is an excellent example of why sharing methodology is important. It allowed one user to submit an 'issue' (problem) with the script, that led to it being fixed and the text of the article improved to more accurately reflect the findings. The repo readme file shows a side-by-side comparison of how the original article text was changed as a result.

Peter Aldhous adds an example of his from **BuzzFeed**: *BuzzFeed News Trained A Computer To Search For Hidden Spy Planes. This Is What We Found*. Again there's a GitHub page explaining the methods employed, and a GitHub repo with the data and an R Markdown file.

The project includes examples of abstraction (in this case, filtering) and algorithms (in this case the random forest algorithm).

Another Trump tweet analysis came from **David Eads**, this one from **NPR** looking at sentiment. He writes:

"**Danielle Kurtzleben** and I took the Trump Twitter Archive and ran the tweets through VADER, a sentiment analysis algorithm tuned for short strings.

"It was a tiny project — a couple hours of reading the VADER paper and some other work on NPL/sentiment analysis, a few hours of Jupyter note booking (which we released), and a lot of talking through what it all meant.

"I'm a big fan of this kind of quick-turn computation work, and it's cool when it makes its way into a format like **radio** that people tend to say is hard to use data with."

And **Mark Lajole** told me about "Bhumika Can Speak for Herself" which "used voice recognition and natural language processing provided by IBM's Watson to create an interactive interview with a Nepali transgender rights activist". It won the SOPA award in Asia for journalistic innovation

On 10 October 2017 at 17:05, Marc Lajoie <manorapide@gmail.com> wrote:
Here's one where we used voice recognition and natural language processing provided by IBM's Watson to create an interactive interview with a Nepali transgender rights activist: https://projects.asiaweekly.com/bhumika-can-speak-for-herself/

The project, "Bhumika Can Speak for Herself," won the SOPA award in Asia for journalistic innovation, beating the New York Times' excellent Pulitzer-winning "They Are Slaughtering Us Like Animals."

On Tue, Oct 10, 2017 at 10:58 AM, David Eads <davideads@gmail.com> wrote:
A lightweight computational example is this Morning Edition story about sentiment analysis of Trump's tweets: http://www.npr.org/2017/04/30/526106612/what-we-learned-about-the-mood-of-trumps-tweets

Danielle Kurtzleben and I took the Trump Twitter Archive and ran the tweets through VADER, a sentiment analysis algorithm tuned for short strings.

It was a tiny project -- a couple hours of reading the VADER paper and some other work on NPL/sentiment analysis, a few hours of Jupyter note booking (which we released - https://github.com/nprapps/trump-tweet-analysis/blob/master/trump-tweets.ipynb, somehow they forgot to link it out from the story), and a lot of talking through what it all meant.

I'm a big fan of this kind of quick-turn computation work, and it's cool when it makes its way into a format like radio that people tend to say is hard to use data with.

On Sat, Oct 7, 2017 at 12:02 PM, Peter Aldhous <peter@peteraldhous.com> wrote:

You've already got one from BuzzFeed, but if you want more machine learning:

https://www.buzzfeed.com/peteraldhous/hidden-spy-planes (final in a series of three that relied on the machine learning - and other reporting, inc. public records)

Methods: https://buzzfeednews.github.io/2017-08-spy-plane-finder/

On 10/6/17 6:46 AM, Max Lee wrote:
The Atlanta Journal Contitution's investigative series on how doctors can often continue their careers years after medical boards find them guilty of sexually abusing patients relied heavily on scrapers, OCR, and machine learning tools. (As in, the reporters scraped through state medical board case decision websites and mined through PDF orders in order to find doctors who were found to have abused patients.)

More about methodology here: http://doctors.ajc.com/about_this_investigation/

On Friday, October 6, 2017, Paul Bradshaw <paulonhismobile@gmail.com> wrote:
I've pulled together 4 of my favourite examples of computational thinking in journalism - but I'd love to add more. Are there any particular examples that you think can help journalism students think computationally?

--

Paul Bradshaw

Out now - Finding Stories in Spreadsheets https://leanpub.com/spreadsheetstories
Snapchat for Journalists: http://leanpub.com/snapchatforjournalists/
Scraping for Journalists: http://leanpub.com/scrapingforjournalists
Data Journalism Heist: https://leanpub.com/DataJournalismHeist
8,000 Holes: How the 2012 Olympic Torch Relay Lost its
Way: https://leanpub.com/8g000holes (all proceeds to the Brittle Bone Society)
The Online Journalism Handbook: http://amzn.to/jEND3p

Please use secure email if you can: my public key is at https://pgp.mit.edu/pks/lookup?op=get&search=0x540D6E3F

Online Journalism Blog http://onlinejournalismblog.com
Help Me Investigate http://helpmeinvestigate.com - Shortlisted for
Multimedia Publisher of the Year, 2010; winner of Talk About Local investigation of the year 2010

Course leader, MA Data Journalism and MA Multiplatform and Mobile Journalism, Birmingham City University
Data journalist, BBC England data unit
Organiser, Hacks and Hackers
Birmingham http://meetupbirmingham.hackshackers.com/

http://twitter.com/paulbradshaw
LinkedIn profile and recommendations at http://bit.ly/paulbrecommendations