

Project 3: A Lit Bot

e249 | SP19 | Electronic Literature | Jason Loan

Your Challenge

Compose a Twitter Bot that attempts, as all eLit does -- "to manipulate language, to transform the linguistic into the literary by means of computation..." ([Heckman and Sullivan](#))

What is a (Lit) Bot?

Here's what the resource [ilovepoetry](#) has to say about them:

The bot is an e-lit genre that goes as far back as 1966 with Joseph Weizenbaum's ELIZA, a chatterbot that engaged users in conversation through text entered and displayed in a computer terminal. This concept informed interactive fiction from the 1980s and has breathed life into video game characters ever since. Poetically, bots are also related to generative works, producing poetry in many forms (haiku, couplets, sonnets, and more), techniques (n-grams, Markov-chains, templates, variables, etc), and datasets (self-contained, data mining, streaming APIs, user-generated, dictionaries, and more).

The *ilovepoetry* site is a could starting point for exploring some model bots. The [Electronic Literature Collection](#) and [Botwiki](#) are also good resources.

How will we do this?

As non-programmers, we are for this project again going to rely on wonderful work of others a bit more knowledgeable in this area. To make your own bot, I point you to two possible resources/methods:

- GalaxyKate's [Tracery](#) in combination with V21's [Cheap Bots, Done Quick](#).

For help in using Tracery and Cheap Bots, Done Quick I used Shawn Graham's [tutorial](#) on The Programming Historian and was able to get this bot -- [@landlocked sushi](#) (thanks, Nick!) -- started in about an hour. At this point, [@landlocked sushi](#) needs a lot more work, but its proof that getting a basic bot started is doable. Note: Graham's tutorial is ultimately aimed at history teachers and using the tools to make something he calls a history bot, but I also found it generally useful. No doubt there are other tutorials on using these tools.

- Zach Whalen's ["How to Make a Twitter Bot With Google Spreadsheets."](#)

A while back I used this method to make my first ever bot called [@outlineOfoject](#). This bot is a mess, as I abandoned it before it was really working well. (Although its now churned out 141K tweets since 2016!) I

thought this might be a good time to revisit it or put it to sleep. Anyway, it's prrof that Whalen's resource can at least get you started!

A Couple of notes on resources/methods

1. Make multiple "throw-away" or "prototype" bots. At least one using each method above.

Each of these novice-friendly bot building methods offer different possibilities and limitations on what you can do. I strongly encourage you to build one potentially "throw-away" or "prototype" bot using each method in order to get a feel for what might be possible. This will also give you a chance to think through potential ideas while getting your hands dirty with the tools. For example, @landlocked sushi and @outlineOfObject where both quick and dirty ideas and making them both helped me get familiar with the tools and sparked other ideas.

2. Use paper in the process.

Like all our projects, computation, the technology is essential and potentially challenging. The technology, however, is of little use without interesting and compelling ideas. Don't hesitate to use paper in your process, to literally draft and sketh things out the most lo-fi of your writing tools. Then try to execute your ideas within the coinstraints of the technology.

3. Pay attention to workflow.

As with all our projects, develop a system that works for you for saving/archiving your work. Whalen's medthod utilizes Google Spreadsheets, so in some ways that is pretty straightforward. Using Tracery will be a little less automatic perhaps, so just be aware of the saving and archiving conventions of the tools you are using and how those will fit into your composing process.

Timeline/Deadlines

Monday [June 3] -- Submit [a link](#) to your working bot.