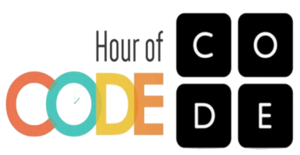
**Projects**



**What Will You Code Next? - Deep knowledge tracing with recurrent neural nets on open-ended student responses**

*Accepted as presenter into* [*Women in Machine Learning Workshop*](http://wimlworkshop.org/) *2016 (WiML 2016) in Barcelona, Spain December 2016*

Modeling student knowledge while students are acquiring new concepts is a crucial stepping stone towards providing personalized automated feedback at scale. This task is also referred to as “knowledge tracing” and has been explored extensively on exercises where student responses can only be correct or incorrect. However, knowledge tracing on open-ended problems where answers extend beyond binary solutions is still mostly unexplored. We believe a rich set of information about a student’s learning hides within their responses to open-ended problems. This is a challenging task, but with recent advances in machine learning, there are more promising techniques to take this on. In our work, we train [Recurrent Neural Nets](http://karpathy.github.io/2015/05/21/rnn-effectiveness/) (LSTMs) to predict a student’s performance and their next move while solving a coding exercise on code.org, an online learning platform for computer programming. Our work shows promising results and brings us a step closer to building automated feedback systems.

Automated feedback is a major challenge for open-ended questions since correct and incorrect solutions can take a variety of forms. This makes personalized feedback that is specific to each student’s answer even more critical, in order to allow the students to understand their performance and steps for improvement. With the inception of massive open online courses (MOOCs), educators from around the world can reach millions of students by disseminating course videos and content through online classrooms. However, in the current model of online courses, scaling feedback for these open-ended questions remains cost-inhibitive and difficult.

Robust knowledge tracing is a crucial step towards personalized feedback. [Piech et al.](https://web.stanford.edu/%7Ecpiech/bio/papers/deepKnowledgeTracing.pdf) applied deep learning to predict student performance on multiple-choice math exercises on [Khan Academy](https://www.khanacademy.org/), and found that RNNs are particularly suitable for this task. However, exercises with open-ended answers like coding problems are much harder to model. In our research, we extend Piech et al.’s work on “deep knowledge tracing” by modeling students’ learning trajectories as they solve [Code.org](https://code.org/)'s open-ended coding problems. More concretely, our deep learning model trains on a student’s history of code submissions and predicts firstly the student’s performance on the next problem and secondly the next line of code that the student will write. We train our model on code submissions because they contain rich information about a student’s knowledge of both code logic and style. Since code submissions are challenging to represent directly in a feature space, we also trained an RNN to generate embeddings for code submissions. These embeddings are then fed into a second RNN, which performs the final predictions over a series of exercises represented by the embeddings.

Resources:

* Source code: [Github](https://github.com/ange3/deepcode)
* Poster: [pdf](http://www.hrlarry.com/s/dkt_poster.pdf), [jpg](http://www.hrlarry.com/s/dkt_poster.jpg)
* Paper: *Publication pending*



**City of Heroes**

*Best Intern Hack, Yahoo! 2014 Q3 Hackday*

Made in 24 hours, this was a superhero-themed personalized, localized, and gameified volunteering app called City of Heroes to increase employee engagement in community service opportunities. It featured a mission control page, achievement badges, service hours tracking, location-based search for service opportunities, urgent "distress signal" functionality for nonprofits, and leaderboards against other employees and teams.

The front end used Javascript, HTML5, CSS, and [Bootstrap](http://getbootstrap.com/). The back end was built with MySQL, PHP, and AngularJS for integration. We leveraged data scraped from the [Benevity](http://www.benevity.com/) volunteering platform and the [Google Maps API](https://developers.google.com/maps/) for our map interface.

This was our entry into Yahoo's 2014 Q3 Hackday and it won the Best Intern Hack award. It was submitted under the category for Tech for Good, intended for hacks dedicated to social good. After winning the Hackday, we presented the hack at a company all hands meeting and recruited full-time employees to take up the reins after the interns leave. The product will be polished and deployed for their next community service cycle.

Resources:

* Source code: Yahoo! proprietary



**Don't Be Late**

*Finalist, Intern Hackday at LinkedIn*

An app that leverages user relationships to encourage punctuality. Groups of friends can opt into an event, and upon doing so commit their device for location tracking. Their location is broadcast to every other individual attending the event, and a agreed-upon punishment is given to the most tardy attendee. A leaderboard tracks average lateness or earliness to increase social aspect.

This product was based on research (e.g. [1](http://link.springer.com/article/10.1007/s11109-010-9114-0), [2](http://www.sciencedirect.com/science/article/pii/0091743591900397)) that demonstrates that social pressures act as a strong source of intrinsic motivation, which can be stronger than a desire for self-improvement.

The front end used Javascript, HTML5, CSS, and [Bootstrap](http://getbootstrap.com/). The back end was built with Parse, the Play framework, and PHP. We leveraged the [Google Maps API](https://developers.google.com/maps/) for our map interface. This app was our entry into the Intern Hackday at LinkedIn and was a Finalist in the competition. It was made with [Eduardo de Leon](https://www.linkedin.com/pub/eduardo-de-leon/65/567/324), [Yinan Ding](https://www.linkedin.com/in/yinanding), [Linda Fayad](https://www.linkedin.com/in/lindafayad), and [Abhinav Khanna](https://www.linkedin.com/in/abhinavkhanna2016).

Resources:

* Source code: [Github](https://github.com/abhiInCalif/dont-be-late2)

[[](http://littlerandomhappiness.blogspot.com/2013/02/sweet-hearts.html)](http://littlerandomhappiness.blogspot.com/2013/02/sweet-hearts.html" \t "_blank)

**Find Your Food Soulmate - Community detection on food preference models using Yelp dataset**

Individuals in social networks are often unaware of people who share similar tastes as they do. From personal experience, we believe that people often bond over food. The goal of this project is to consistently identify clusters wherein users can both discover new friends, and also identify members of their own friend lists with whom they may share common food tastes with.

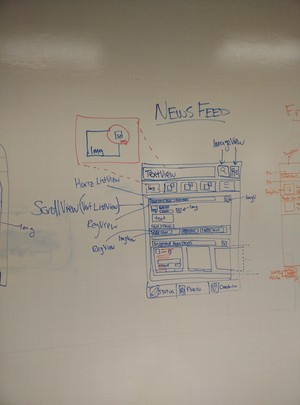
To achieve this, we took information from the [Yelp academic dataset](https://www.yelp.com/academic_dataset), identified network properties and generated a network model that recognizes food compatibility between Yelp users. We classify similarity in user attitudes towards restaurants (e.g. similar ratings and reviews of restaurants) as indicative of similarity in user food tastes as well. This allowed for the creation of an evaluation network against which we tested and optimised training on other factors of food-related experiences as indicative of similarity in food tastes (e.g. parking, cost, atmosphere). We finally run a clustering algorithm on a friends-specific subgraph of our model and compare this with the real-world network for the purposes of friend recommendations centered on food.

This project was written in Python, and leveraged the Stanford Network Analysis Project's Python package, [Snap.py](http://snap.stanford.edu/snappy/index.html), for network functions. This project was made with [Angela Sy](http://stanford.edu/%7Eangelasy/) and [Blanca Villanueva](https://www.linkedin.com/pub/blanca-villanueva/61/a3b/68b) for [CS 224w: Social and Information Network Analysis](http://web.stanford.edu/class/cs224w/).

Resources:

* Source code: [Github](https://github.com/ange3/food_soulmate)
* Paper: [pdf](http://www.hrlarry.com/s/Food-Soulmate-Paper.pdf)
* Poster: [png](http://www.hrlarry.com/s/poster_presentation_final_with_title.png)

**DOMiNO**



DOMiNO demo whiteboarding

A project that experimented with replacing the Document Object Model (DOM) when rendering pages in web browsers. This was my team's entry for [Treehacks 2015](https://www.treehacks.com/), a 36-hour hackathon at Stanford.

Instead of the DOM, we wrote our own library of canvas "Views" which we believe would be faster for displaying content that has a lot of dynamic components. Meanwhile, as far as the DOM is concerned, every single webpage would only have a single canvas object and fewer than 20 lines of HTML (see GitHub repo). All of the heavy lifting is abstracted into our Javascript library. Content is rendered based on a series of attributes common across all Views (width, height, position, background color, etc.) and type-specific content (e.g. text for textViews, images for imageViews, among others).

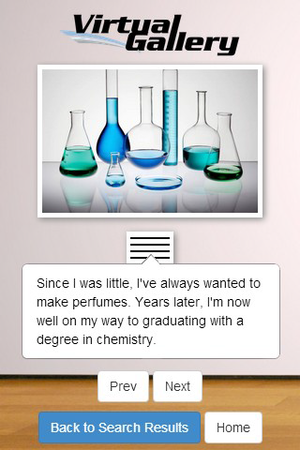
By confining content into individual views, we can expedite caching and animation demands on client computers. This model would see dramatic performance gains in contexts such as high-end screen-sharing, gaming, etc. As is, the products for such services need to be developed away from web browsers. Furthermore, styling the webpage is far simpler because of how intuitive the attributes are compared to CSS, so it is much more easily adoptable for users with minimal technical expertise.

As a proof of concept, we mocked up a couple of primitive pages in a [replica Facebook Mobile site](http://onion.firecaster.com/demo/) using DOMiNO (compare to [original Facebook Mobile site](https://m.facebook.com/)) to show how even sophisticated websites can be made using our library. One of the biggest challenges we encountered in over the two days we worked on this project was debugging when there was only one true "object." To combat this, we wrote our own debugger: by using the Ctrl+A shortcut in the Facebook Mobile demo, you can actually enter a debugging mode where you get debugging information at a View-level specificity rather than DOM Object-level specificity.

This project was written in HTML and Javascript. The demo leverages the [Facebook Graph API](https://developers.facebook.com/docs/graph-api) for retrieving live data (data in current demo site is static for privacy purposes). The project was made with Joel Einbinder, Matt Ho, and Bobby Pourkezemi.

Resources:

* Source code: [GitHub](https://github.com/JoelEinbinder/JS100)
* Demo site: [Facebook Mobile DOMiNO Demo](http://onion.firecaster.com/demo/) (Ctrl+A for debug view)



**VirtualGallery**

When you walk into someone's room, you can learn a lot about what kind of person they are, their personality, their idiosyncrasies, among others just by their belongings and how they're arranged. This sort of sensory stimuli is often more powerful than words, especially with regards to more intangible concepts like personality traits and passions. In this case, a picture *is* worth a thousand words!

This project was devised to help provide an alternative way to meet people. Users are able to upload images of things that are meaningful to them and describe them as if they were the curator of their own personal gallery. They can associate particularly salient keywords with each "exhibit," and based on these keywords we can find other people that share similar traits.

This project was written in JavaScript, Node.js, HTML5, and CSS. It leveraged MongoDB for database functionality and [Bootstrap](http://getbootstrap.com/) for off-the-shelf interface components. This project was made with Sophie Ye and Stephanie Palocz for [CS 147: Introduction to Human-Computer Interaction Design](http://hci.stanford.edu/courses/cs147/2014/au/index.html). The focus of the class was on rapid prototyping, end-user A/B testing, and usability.

Resources:

* Source code: [Github](https://github.com/hrlarry/VirtualGallery)
* Poster: [pdf](http://www.hrlarry.com/s/virtualgallery_poster.pdf), [pub](http://www.hrlarry.com/s/virtualgallery_poster.pdf)



**Movee**

Movee is a very primitive front-end prototype of an app intended to help involve family members in promoting elderly relatives' health by providing elderly individuals an engaging virtual environment for a more rewarding exercise experience. Family members can contribute to the experience by uploading packages (e.g. vacation photos, grandchildrens' recital footage, etc.) to unlock after achieving certain health milestones.

The prototype was made in two 2-hour ideation sessions in JavaScript, HTML, and CSS. The app was one of the solutions for improving elderly well-being, in this case physical health, for a project in PSYCH 102: Longevity, a class inspired by the [Stanford Center on Longevity](http://longevity3.stanford.edu/).

Resources:

* Source code: [Github](https://github.com/hrlarry/movee)

**Hunt the Wumpus 3D FPS**

*Winner (Classic Category), 2012* [*Microsoft Annual Hunt the Wumpus Competition*](http://microsoftjobsblog.com/wumpus-outreach-project/)

A modern recreation of the classic [Hunt the Wumpus](http://en.wikipedia.org/wiki/Hunt_the_Wumpus) video game first released in 1972. The objective of the game is to navigate the cave maze to kill the wumpus while avoiding traps. Our version involved a 3D environment that the user navigated in a First Person Shooter perspective. It featured different theme packs that modified the sound effects and graphics as well as a leaderboard and designated trivia minigame for the traps.

The project was written in C# and Microsoft XNA. Models and textures were created in Blender. The game was written to be compatible with both keyboard inputs and Xbox controllers.

Resources:

* Source code: not released to maintain integrity of annual competition, per rules
* Executable download: sourceforge link coming soon