Teach2Learn: Gamifying Education to Gather Training Data for Natural Language Processing

by

John J.D. O’Sullivan

S.B., Massachusetts Institute of Technology (2015)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Signature redacted

Author

Department of Electrical Engineering and Computer Science

February 3, 2017

Signature redacted

Certified by

Kalyan Veeramachaneni
Principal Research Scientist
Thesis Supervisor

Signature redacted

Accepted by

Christopher J. Terman
Chairman, Department of Electrical Engineering and Computer Science
DISCLAIMER NOTICE

The pagination in this thesis reflects how it was delivered to the Institute Archives and Special Collections.
Teach2Learn: Gamifying Education
to Gather Training Data for
Natural Language Processing

by

John J.D. O'Sullivan

Submitted to the Department of Electrical Engineering and Computer Science on February 3, 2017, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

Abstract

Teach2Learn is a website which crowd-sources the problem of labeling natural text samples using gamified education as an incentive. Students assign labels to text samples from an unlabeled data set, thereby teaching supervised machine learning algorithms how to interpret new samples. In return, students can learn how that algorithm works by unlocking lessons written by researchers. This aligns the incentives of researchers and learners to help both achieve their goals. The application used current best practices in gamification to create a motivating structure around that labeling task.

Testing showed that 27.7% of the user base (5/18 users) engaged with the content and labeled enough samples to unlock all of the lessons, suggesting that learning modules are sufficient motivation for the right users. Attempts to grow the platform through paid social media advertising were unsuccessful, likely because users aren’t looking for a class when they browse those sites. Unpaid posts on subreddits discussing related topics, where users were more likely to be searching for learning opportunities, were more successful. Future research should seek users through comparable sites and explore how Teach2Learn can be used as an additional learning resource in classrooms.

Thesis Supervisor: Kalyan Veeramachaneni
Title: Principal Research Scientist
Acknowledgments

Many people helped make this project come to life and need thanking:

Kalyan Veeramachaneni, for getting behind the original idea and working with me through its fits and starts.

Azan Abrar, for numerous previous freelance contributions and setting up the Collaborative Platform deployment.

Arash Akhgari, for making storyboards and animations to go along with the Lesson Modules, patiently bearing with my endless revisions.

Jolene Fong, Olivia Klinikowski, and William Lopez-Cordero for helping me with usability testing.

All my family and friends for pretending to listen to the thousandth version of the explanation of this thesis.

And Douglas Adams, for the apt saying:

I love deadlines.
I love the whooshing noise they make as they go by.

Douglas Adams
## Contents

1 Introduction ................................................. 13  
  1.1 Steps ..................................................... 15  
  1.2 Evaluation ............................................. 16  
  1.3 Outcomes ............................................... 17  
  1.4 Contributions ......................................... 18  

2 Literature Review ........................................... 19  
  2.1 Gamification Theory .................................... 19  
  2.2 Meaningful Games in Context .......................... 20  
  2.3 Guidelines for MOOC Content ......................... 22  

3 Prior Art ....................................................... 23  
  3.1 Zooniverse ............................................... 23  
  3.2 Fold.it .................................................... 24  
  3.3 reCAPTCHA .............................................. 25  
  3.4 Key Insights ............................................ 26  

4 Implementation .............................................. 27  
  4.1 Overview ................................................ 27  
  4.2 Previous Work .......................................... 28  
  4.3 Game Affordances ...................................... 31  
  4.4 Lessons User Interface ................................. 33  
  4.5 Lesson Content ........................................ 36
List of Figures

4-1 The user listing page without any filter applied. ................. 29
4-2 The user listing page after filtering for the Data Scientist tag. ... 29
4-3 The user profile page. ........................................... 30
4-4 The level bar shows the fraction of the total required points to advance
from the current level to the next. ................................. 31
4-5 The complete labeling interface, with level bar at bottom and pie
graphs on right. .................................................. 33
4-6 The Lessons page as viewed by a new user who has not yet unlocked
any lessons beyond the introductory offering. .................... 34
4-7 Detailed view for the Bag of Words lesson. ....................... 34
4-8 Admin form for creating and editing lessons. .................... 35
4-9 The introductory "What is Natural Language Programming?" lesson
video shows a "smart folder" learning to understand new samples from
previous labeled ones. .......................................... 36
4-10 The "Bag of Words" converting the sample sentence into its internal
representation. .................................................... 37
4-11 A visual demonstration of how singular value decomposition "magi-
cally" collapses similar terms into compound topics. ............ 37
4-12 Close-up of Disqus comment plug-in. ........................... 38

A-1 Collaborative Platform Entity Relationship Diagram ............... 54

B-1 The user finds our landing page via some referral source. ....... 62
B-2 If the user scrolls down, they will find more details describing the lessons in the Collaborative Platform.

B-3 If user tries to open Lessons without logging in, they're redirected.
   They click Sign Up to create an account.

B-4 User fills out the new account form with normal account details, a bio, and a list of self-descriptive tags.

B-5 User receives a confirmation email in their specified inbox.

B-6 User is now logged in, headline actions become links.

B-7 User goes to look at their available data science lessons.

B-8 User explores introductory content.

B-9 User exhausts content and starts labeling to unlock next lesson.
List of Tables

4.1 Costs per each level in the Collaborative Platform. .................. 32

5.1 Total views for each page as drawn from Google Analytics. .......... 42
5.2 Teach2Learn Visitor Nationalities & Languages ....................... 43
5.3 User Nationalities .................................................. 43
5.4 User Languages ................................................................ 43
5.5 Labeled Samples Per User .................................................. 44
5.6 Breakdown of Crowd-sourced Label Accuracy ............................... 44
5.7 User sign-ups per advertising medium. ..................................... 45

A.1 Project Upload Data Schema ............................................... 53
A.2 User Schema Pt. 1 ......................................................... 55
A.3 User Schema Pt. 2 ......................................................... 56
A.4 Lesson Schema ............................................................ 57
A.5 Upload Schema ............................................................. 58
A.6 Label Category Schema .................................................... 58
A.7 Label Tag Schema .......................................................... 59
A.8 Post Schema ............................................................... 59
A.9 Sentence Schema .......................................................... 59
A.10 Response Schema ......................................................... 60
A.11 Post User Schema ........................................................ 60
Chapter 1

Introduction

Natural Language Programming (NLP) studies how software can interpret written human language and derive meaning. Current applications range from determining the sentiment behind a tweet (i.e. "Did people generally like this new movie?") to assessing what the user wants from a request (i.e. "Siri, find me an enchilada, stat!"). Interest in the field has grown rapidly given the breadth of possible applications. The key relevant fact about the NLP algorithms performing these tasks is that they require pre-labeled examples in order to understand new ones. They can only recognize positive tweets after examining a few thousand tweets which a human being read and manually labeled.

This leads to a fundamental manpower problem. Labeling text samples falls into the broad category of Human Intelligence Tasks (HITs), mundane problems which humans can easily solve but are still impossible with current computational methods. Identifying objects in images is another common example of an HIT. The current best platform for completing these tasks in bulk is Amazon’s Mechanical Turk (MTurk). MTurk lets anybody around the world complete batches of HITs for a handful of cents apiece. This approach works well for companies gathering data, but many academic researchers can’t afford to pay per sample. Our research aims to help label more samples by leveraging three growing trends: crowd-sourcing, gamification, and online education. A brief introduction to their beginnings and growth provides necessary context for the thesis.
Crowd-sourcing is the practice of opening up a problem to let anybody on the internet contribute and help solve it. A June 2006 Wired article coined the term: "The productive potential of millions of plugged-in enthusiasts ... isn't always free, but it costs a lot less than paying traditional employees. It's not outsourcing, it's crowdsourcing." [4] The practice has grown rapidly since then, as the availability of many volunteer contributors online promises a quick solution to otherwise intractably large problems. This strategy seems promising, but it hinges on the "crowd" actually having an incentive to contribute. One growing strategy for motivating people is to make the task a game.

Gamification is, roughly, the practice of building games to motivate some behavior in a non-game context. The term was originally coined in 2002, although some fields had already begun borrowing elements from video games before then. It truly took off in 2010 as app developers began to interpret it as adding social and reward elements to their baseline functionality. The landmark example of using social reward features in non-game software was Foursquare, a check-in app for telling friends where you are, which let you earn points for that location and climb its leaderboard. Early attempts which simply added points onto an activity were criticized when new research discovered that adding extrinsic motivation to an activity can sap intrinsic motivation. [9] New frameworks have been developed in response, because the fundamental practice of developing games for a purpose has proved useful and is here to stay. [7]

Meanwhile, interest in online learning has exploded in the last decade. In 2011, there were about 160,000 online learners at one university. Fast forward to 2015 and there are 35,000,000 online learners across 570 universities and twelve Massive Online Open Course (MOOC) providers. [1] It provides a cheap way to share educational resources with far more people, meeting learners where they’re at and letting them work on their own schedule. Many of these people are non-traditional students (e.g. business intelligence analysts) trying to expand their skill set by learning topics in computer science. 17% of all MOOCs are in computer science topics, the largest category other than business (19%), largely because MOOC sites are monetizing by selling certificates for applied skills. [10] NLP is of particular interest, as it is a
young field with direct business applications. This demand for course content in NLP presents an opportunity for researchers who can teach it.

In this thesis, we crowd-sourced the labeling problem using gamified education. We designed a system named Teach2Learn which brings researchers and learners into one community. Researchers share unlabeled text data sets (e.g. tweets, reviews, forum posts, etc.) and specify what the possible labels are for each sample. Learners label these samples in order to earn points and levels, which in turn unlock short, practical lessons about NLP. These labeled samples teach the algorithms how to recognize new samples, and in exchange, users learn about cutting-edge algorithms with content created by researchers. We enable a collaborative, hands-on learning style where people can learn while getting involved with active research in the field. The gamification system aligns the incentives of two different groups to help them both get what they want. Learners want to label in order to get lessons from top researchers, and researchers want to produce great lessons to inspire more people to label. The remainder of this introduction discusses the research in more depth, describing the final outcomes and contributions to the field.

1.1 Steps

The Teach2Learn platform leverages this need for educational NLP content to bring a crowd of volunteers into the labeling problem. There were three core components to its development: building the gamification system, implementing the lessons interface, and crafting the sample lessons.

We set out to use current best gamification practices when building system to motivate individuals into spending their time labeling text samples. People want to satisfy three core needs: competence, relatedness, and autonomy. Users earn points for each labeled sample, eventually earning levels which unlock learning content, increasing their sense of competence as they progress further. Community-oriented features like leaderboards and comment sections help learners feel like they are part of a growing team driving real research in the field, satisfying the need for related-
ness. The website also takes interface cues from modern video games, borrowing the persistent level bar used in many Massively Multiplayer Online Role Playing Games, to make the labeling and learning process more engaging.

We then developed a simple MOOC platform to deliver lessons to our users. Each lesson has a minimum level which dictates when a user unlocks it. The lessons contain short (i.e. 60-120 seconds) whiteboard videos explaining the idea behind an algorithm. They then walk the user through a sample use case with a blog post, Python script, and sample data. Users are able to download the data they labeled, directly connecting the learning experience to their labeling work.

Finally, we created three sample lessons to test the platform. The first lesson is a broad explanation of NLP and its purposes, along with a walk-through on how to set up their computer with the right software. The second lesson explains how the Bag of Words model transforms an input into a dictionary of how many times each word appeared, then shows learners how to find the top ten words used in the sample corpus. The third lesson explains how Latent Semantic Analysis merges related words into "topics", then shows learners how to find the top topics in a corpus and interpret them. Many students of data science express frustration when they spend a long time learning the mathematical proofs behind a technique, then find that applying it boils down to using a few pre-written functions in a software package. Our lessons are designed to be quick and easily digestible, summarizing the underlying mathematics to give the user an intuitive explanation of what the algorithms do, then introducing them to the software packages that implement them.

1.2 Evaluation

We solicited users to assess the effectiveness of the platform by advertising on two social media platforms, Facebook and reddit. We used both paid and unpaid posts, shared over the course of three days. Users indicated how they found Teach2Learn on the signup form, letting us track the effectiveness of each medium. The site also has Google Analytics installed, letting us track the number of views on each page over
time as well as where our users came from.

1.3 Outcomes

First results from testing Teach2Learn were encouraging, showing that 47% of signed-up users (9/19) made at least one label and 21% of signed-up users (4/19) created enough labels to unlock all of the available content. On the other hand, only 6% of users who found the site (19/315) ended up signing up. Tracking user engagement with lessons was harder to quantify, as opening a lesson does not create any record in the database. Google Pageview results showed that the Lessons page was viewed 13 times, suggesting 68% of users who signed up at least browsed to see what educational content was available.

Paid social media proved to be an ineffective marketing channel, yielding low click-through and sign-up rates. Facebook ads showed the first lesson to 1,618 users, but only one clicked through to the site and nobody signed up. Reddit showed the ad to 59,115 users, yielding 42 click-throughs and 1 sign-up. The unpaid post to related subreddits was the most effective strategy, yielding 11 sign-ups. Low user interest is likely because social media users are browsing for entertainment, rather than searching for learning content. However, the effectiveness of the unpaid posts to relevant subreddits suggests that targeting users who are seeking learning should be more promising.

This research was intended to act as a proof of concept with preliminary trials, so we did not explore more advertising options like other subreddits or social media platforms. We also constrained advertising to one unpaid post and one paid post on each platform, with duplicate advertising budgets ($25 apiece) for the paid post, in the interest of getting a baseline control result. Now that the platform exists and has been seeded with content, there is ripe research to be done on how to best spread it. Future outreach should focus on finding those people who are looking to learn, rather than broadly advertising.
1.4 Contributions

Beyond the direct contributions to natural language programming via gathering labeled samples, Teach2Learn adds to the bodies of knowledge surrounding gamification and education. We are the first platform which gamifies HITs using educational content as an intrinsic motivator. Previous for-profit and nonprofit tools crowdsourcing HITs, but we are the first nonprofit tool which appeals to users via the promise of education rather than altruism.

Teach2Learn also demonstrates a novel approach to education, connecting people who are newly interested in the field with researchers at the cutting edge. We bring together the community around NLP to create a positive feedback loop. Researchers will create lessons so users have more content to unlock, users will label more samples to learn from the best. The HITs are also directly connected to the content users are learning about, contextualizing the whole experience. This is more powerful than having users simply accomplish some HITs and then giving them lessons about unrelated topics. We are teaching them the fundamentals of the field while also sharing the work required to push the cutting edge forward.
Chapter 2

Literature Review

The Teach2Learn project required an understanding of current knowledge about (1) implementing gamification systems, and (2) online educational content. Creating games to motivate behaviors (gamification) has become a popular practice in the last decade for a variety of applications, spawning academic research about the most effective practices. Similarly, Massive Online Open Courses (MOOCs) became popular in the 2000s to try and get better education to more people as technological capabilities have grown. The key conclusions we drew from the literature were (1) games need to help people satisfy their drives for competence, autonomy, and relatedness, (2) games need to acknowledge their context and focus on supporting intrinsic motivation, and (3) online learning must be delivered in discrete units of multimedia content.

2.1 Gamification Theory

Previously in the field of gamification, there have been a wave of practitioners who approach it as adding points, badges, and achievements – a set of additional incentives on top of an activity which quantifies it[2]. There has been a backlash to this approach because studies have shown that adding extrinsic incentives to an activity saps intrinsic motivation. The leading motivational theory in the gamification space, Self-Determination Theory, seeks to unify intrinsic and extrinsic theories of motivation[9].
In Self-Determination Theory, the individual has three desires they seek to fulfill: autonomy, competence, and relatedness. Researchers found that gamification systems which preserve these three elements led to increased learning outcomes and synergized well with the optimal mindset for learning. Both of these activities produce, in a best case scenario, a state of "flow". Flow is a feeling of energized focus and deep involvement, the best sort of engagement. The key requirements are (1) being involved in activities with clear and structured goals or progress, (2) performing tasks with articulate and immediate feedback, and (3) a good balance between perceived challenge level and skill level.

Relatedness is a critical part of the equation, but it also has to be managed carefully. Researchers found that in some platforms, the affordances for social interaction distracted the learner from the core content and kept them from properly engaging with the game[11].

2.2 Meaningful Games in Context

The backlash to the "traditional" BLAP (Badges, Levels, Achievements, Points) gamification has produced a number of alternate theories, the most coherent of which is Scott Nicholson's "RECIPE for Meaningful Gamification."[7] His recipe describes six key elements which make up a meaningful gamification experience:
Play: Freedom to explore and fail within boundaries.

Exposition: Share a story with participants that is integrated with the real-world setting, inviting them to make their own story.

Choice: Put the power of what to do next in the hands of the participants.

Information: Display concepts from the real-world context and use game design interfaces where they are helpful.

Engagement: Discover other people and learn from each other about the context.

Reflection: Help participants finding other interests or experiences to connect with what they just learned, deepening engagement and instilling intrinsic motivation.

A separate piece of commentary, "Six Invitations to Rethink Gamification"[2], seeks to reframe the idea of a game. Gamification shouldn't be about designing the right tool, but rather crafting a playful experience. Instead of seeing a game as a piece of software, it should instead include the person playing it and the situation they are in. For instance: a city installed pull-up bars in their subways to promote fitness, but found nobody used them. When asked, people said it would be weird to start doing pull-ups in the subway. Playfulness is only part of the social norm in certain situations.

An interesting consequence of norms around play is gamesmanship. It is assumed in a competitive game that each player will act solely in their own best interests, even though such behavior would be seen as psychopathic in other parts of society. The instinct to push the rules to their limits, "min-maxing" the penalties and rewards, is part of what we consider "playing a game". Researchers found that when an activity is gamified, users take that as a signal that self-centric min-maxing is a valid approach.

For our purposes, we concluded that gamification needs to be connected to the purpose of the underlying activity, rather than a set of extra strap-on design elements, in order to avoid its potential adverse effects. People need to be able to engage with the underlying activity; gamification should only be incorporated insomuch as it helps with that goal.
2.3 Guidelines for MOOC Content

This project required the creation of learning content, so we took cues from a study which performed a survey of the top 6 providers of Massive Online Open Courses to design a model for how the content should be presented[12]. Not all of the guidelines mapped neatly to our different scenario (i.e. one week learning modules were impractical for the early stage testing we were conducting), but they were used as goals nevertheless.

Based on the survey, educational material should include a video which is shorter than 20 minutes and contains most of the material. The video should come with exercises which check that the student is understanding the material, and projects should be peer-reviewed. Our lessons followed this model.
Chapter 3

Prior Art

The Teach2Learn platform follows in the footsteps of three key examples. Zooniverse is a large-scale labeling project which lets researchers upload datasets of images and have volunteers identify them. Fold.it is a competitive puzzle game which leverages human spatial reasoning to find optimal 3D folds of a protein structure. reCAPTCHA, developed at Carnegie Mellon University and purchased by Google in September 2009, repurposes the Captcha system for protecting websites from bots to also help with digitizing text. The key insight we draw from all of these is that crowd-sourcing is an effective strategy for labor-intensive research efforts, provided that users have some intrinsic motivation to contribute. The Zooniverse relies upon altruistic citizens, Fold.it capitalized on gamers who love puzzles, and reCAPTCHA piggybacked onto users who wanted to enter a website. We piggyback on people’s intrinsic desire to learn about NLP, augmented with a gamification structure.

3.1 Zooniverse

"The world’s largest and most popular platform for people-powered research" lets researchers upload their own datasets and have volunteers identify the images[14]. It is a collection of separate projects, where each page corresponds to a different researcher’s data set. The Zooniverse was created after the wild success of GalaxyZoo, started in 2007, which asks volunteers to examine photos of the sky and identify the
shapes and features of blobs which a machine vision algorithm suspects could be galaxies. There are now many projects in the Zooniverse, associated with a wide variety of disciplines; arts, biology, climate, history, language, literature, medicine, nature, physics, social sciences, and space are all represented.

The collective project has been a definitive success. Over the ten total years of data collection, GalaxyZoo yielded data for 53 research papers, other projects produced 55 papers, and the entire platform has been the subject of 26 meta-studies. All of the sites track how many labels you’ve made across the entire platform, but there is no leaderboard of any sort to compare against other players. The count is also displayed as a simple tally on your profile, missing from any of the other pages a user interacts with. Each project does have its own forum, though, increasing the sense of relatedness for the users. Offering a wide variety of content types also increases users’ sense of autonomy, letting them choose to classify content that they find interesting. These limited gamification affordances, supported by citizens’ intrinsic desire to contribute to research, were sufficient to produce successful engagement with the Zooniverse.

3.2 Fold.it

Proteins are made up of strings of amino acids which fold into three dimensional structures, and these structures dictate how they connect and fit together. Determining the shape of a protein is critical for designing treatments, but the laboratory methods to find the shape are expensive and slow. Strictly computational methods are promising, but have only found limited success. Fold.it is a competitive puzzle game released in May 2008 which leverages large-scale human spatial reasoning to have people find the optimal folds of simulated proteins[3]. The volunteer players all apply their spatial reasoning to find the best folds, essentially performing a very specific HIT which is far beyond the reach of computers. The game has proved effective, yielding a total of eleven research papers. Five papers used the best fold solutions to deduce knowledge about protein structures, and the others studied the interface from

24
a game design perspective to find ways of improving player results.

Fold.it is an excellent example of gamifying a problem. Protein folding mapped neatly to a scoring system which gave points for (1) how tightly the protein is packed, (2) how deeply hydrophobic side-chains are hidden inside the protein, and (3) how much distance exists between side-chains, to prevent clashing. The game is therefore intimately tied to the domain, implicitly teaching the players chemistry and biology while they played with the interface. The scoring system provides a sense of growing competence, as players can perform multiple attempts on the same protein to maximize their score. All of the players' solutions were posted to a global leaderboard, relating them to the surrounding community. Players could also build off of each other's solutions, finding further optimizations, reinforcing the fact that even as they compete, they are collaborating toward a shared goal. Fold.it players stuck with the game partly because it creates a competition for a good cause. Problems similar to protein folding, where there are quantifiable metrics of success but no good computational methods to optimize them, are ideal for gamification.

3.3 reCAPTCHA

Captcha are widespread tools used to detect whether a computer user is human, generally to protect websites from getting attacked by bots. The earliest versions took a string of characters and then warped them, making identifying the text difficult for vision algorithms but still possible for humans. The reCAPTCHA system was built at Carnegie Mellon University and used by Google starting in September 2009 to digitize printed text which couldn’t be processed using optical character recognition (OCR) systems[8]. A user is presented with two printed words, generally slightly mangled with text warping. One of the words is a control word, testing whether the user is a human. The other, however, is a printed word that Google’s two OCR algorithms disagreed on. reCAPTCHA collected the results from users identifying the unknown word and marked it as digitized when two users agreed.

This simple word-by-word system let Google completely digitize the New York
Times and Google Books archives by 2011. The system was revamped for identifying house numbers taken from Google's Street View project in 2012, and has been used to identify images (e.g. "Select the cats from this set of 9 images") since 2014. The reCAPTCHA system is one of the better examples of repurposing users' motivations. People were just trying to enter their desired sites, but in the process they were able to digitize massive quantities of text. Rather than creating some new motivation for people to label text samples, reCAPTCHA piggybacked onto something users were already interested in doing.

3.4 Key Insights

Teach2Learn builds on these prior examples, using a gamification system (like Fold.it's) to repurpose user motivations (like reCAPTCHA) towards solving the labeling problem (like Zooniverse). reCAPTCHA demonstrated the power of piggybacking onto prior motivations, Fold.it showed how game systems can drive crowd-sourcing efforts, and the Zooniverse highlights the broad need for data labeling. We improve these prior works by incorporating explicitly educational content as an intrinsic motivator for users. Our platform teaches interested users about natural language processing, while also performing some necessary labeling work in the process.
Chapter 4

Implementation

The core development for Teach2Learn consisted of adding game interface elements, building a lessons interface, and crafting the lessons themselves. We then performed two rounds of usability testing and integrated the feedback to get the site ready for general release. The following is a summary of the key areas of development on Teach2Learn. The reader can also find a screenshot-by-screenshot walk-through of a new user’s flow through the site in Appendix B.

4.1 Overview

The Teach2Learn system has two broad sections, the labeling interface and lesson interface. When a regular user navigates to the labeling interface, they select one of the available projects and are redirected to start labeling samples. The labeling page presents an initial set of popup instructions describing the data set and then lets the user label. The lessons page lists all of the lessons, and lets the user click on the ones they’ve already unlocked. Levels are unlocked when the user reaches the required minimum level. Each lesson detail page presents the user with the core lesson content: a short video about a technique, a blog post walking through its use, a Python script using it, and sample data to test with.

Each part of the site also includes an admin view for creating and editing content (i.e. uploading new data sets, updating lessons). When an admin views the page
listing all projects, they see additional interface elements for uploading new data sets, updating possible labels for existing data sets, and downloading all the user-created labels. The lesson list page also displays additional interface elements for admins, letting them create new lessons, update existing ones, or delete them.

We populated the lessons page by creating three sample lessons covering: a broad overview of NLP, the Bag of Words model, and Latent Semantic Indexing. The lab contracted an animator to work with me and produce the whiteboard explanation videos, then I wrote up walk-through blog posts and Python scripts. The scripts used popular Python machine learning libraries, scikit-learn and gensim.

The majority of new implementation work on Teach2Learn was (1) adding gamification affordances like a persistent level bar, leaderboard charts, and comment boxes, (2) implementing an interface for creating, unlocking, and viewing lessons, and (3) creating sample lesson content. The remainder of this chapter will briefly describe the previous work which went into the platform and then discuss the novel development in detail.

4.2 Previous Work

Building upon the previous work on this project, I took three platforms built by earlier Master's students (LabelMe-Text, MOOCviz, and Feature Factory) and unified them into one application. Teach2Learn is a Ruby on Rails app backed by a MySQL\textsuperscript{1} database. It runs on Apache via the Passenger web app server, all hosted by DigitalOcean. The codebase is stored in a private GitHub repository which includes core documentation.

The original also includes a rich user profiles and a user directory which makes it easy to find users by interest. Each user's profile includes a 160-character bio and at least one self-descriptive tag, like Machine Learning Researcher or Sentiment Analysis, which will be displayed by their name in the list page. They can also optionally include their location and a personal URL. The user listing page then lets

\textsuperscript{1}The specific MySQL implementation is MariaDB.
wanted and there was no leaderboard to support competition within the site. They were an additional layer quantifying an otherwise boring activity, without adding in any elements of exposition, choice, or social engagement, as would be prescribed by Nicholson’s RECIPE.[7] The Zooniverse, the most direct predecessor of Teach2Learn, was successful partly because it included forums and a variety of content types to provide users a more meaningful game experience. On the other hand, the original Teach2Learn points, levels, and badges didn’t connect the user to the real-world context or engage them with the surrounding community, making them ineffective motivators for voluntary contribution.

We set out to remedy this by instead creating learning content about data science and using that as an incentive to give users intrinsic motivation toward reaching the next level. In order to test the core hypothesis of whether lessons can motivate labeling, we eliminated the MOOCviz and Feature Factory platforms from the interface. They are still available in the codebase and would make good additions in the future by expanding the set of ways users can participate on the platform. For now, however, they are hidden to keep the focus on the lesson and label interfaces.

4.3 Game Affordances

There were two key affordances added to help motivate users: a persistent level bar and a set of charts on the labeling page. The original gamification effort partly failed because it had insufficient affordances to make the user feel like they were progressing. Points were accumulating as they made labels, but there was no visual feedback from the label page. Users need to feel like they are progressing in order to satisfy their desire for competence, so it was critical to add a persistent reminder as they label.

Our first affordance for gamification was adding a persistent level bar, taking a cue from Massively Multiplayer Online Role Playing Games, which are centered around spending lots of time to progress in a virtual world. When a user is logged in, they are able to see their progress toward the next level at the bottom of every page. This helps cement the user’s sense of competence, as they watch their labeling experience
Users unlock lessons by accumulated labeling experience, earning a new lesson for each level. The number of experience points a user has is computed by multiplying their total number of responses with a constant. In the spirit of computer science, levels are zero-indexed — a new user begins at level zero and labels to reach level 1. The first four levels require an increasing number of labels to achieve, up to 50 labels, and then the level cost is constant. This is currently set to ensure that users create 100 labels in order to unlock the first four lessons, yielding a reasonable training set of 3000 labels given a group of 30 motivated individuals.

<table>
<thead>
<tr>
<th>Level</th>
<th>Labels Required</th>
<th>Points Required</th>
<th>Total Labels</th>
<th>Total Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>100</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>200</td>
<td>30</td>
<td>300</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>300</td>
<td>60</td>
<td>600</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>400</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td>(n \geq 5)</td>
<td>50</td>
<td>500</td>
<td>(100 + (n - 4) \times 50)</td>
<td>(1000 + (n - 4) \times 500)</td>
</tr>
</tbody>
</table>

Table 4.1: Costs per each level in the Collaborative Platform.

The other major affordance for gamification was adding a set of dynamic charts to the labeling page describing the ongoing contributions of users. The data for the charts is computed on the fly for each label and appears next to the labeling interface as shown in Figure 4-2. The first chart acts like a leaderboard, showing how many labels were contributed by each user. The following charts break down the percentage of labels for each tag within each category, such as how many posts were tagged as "Giving help or information" for the "Role of Poster" category.

The first chart adds an element of social competition and increases the user's feeling of competency as they see themselves rising through the ranks. The following charts expose players to the real-world information they're helping generate and also increases feelings of relatedness. Users are no longer labeling in isolation, they can
see how their contributions are connected to a greater community. They are able to both see that they are contributing and get a better understanding of what they’re contributing to.

4.4 Lessons User Interface

The lessons interface consists of three key components: a list page showing all the available lessons, a detail page showing the content of each lesson, and an admin form for creating and updating lessons. The lesson list page is accessible from the navigation bar and shows all of the lessons the user has unlocked. The lesson listings include a brief description of the content and a thumbnail of the video. Each lesson has a minimum required level and is disabled if the user has not unlocked it yet. The text overlay on the locked lesson was added after performing user testing, as described more thoroughly in Section 4.5.

Each lesson’s detail page includes:

- A short (60-120 seconds) animated whiteboard video explaining a topic
- A blog post demonstrating how the topic is used
What is Natural Language Processing?
A brief introduction to Natural Language Processing and a guide on setting up your computing environment.

You haven't unlocked this lesson yet; try making some labels!

You haven't unlocked this lesson yet; try making some labels!

Figure 4-3: The Lessons page as viewed by a new user who has not yet unlocked any lessons beyond the introductory offering.

- A comments section for users to discuss, share, and ask questions

What is the Bag of Words?
An explanation of the most fundamental machine learning model used to describe text, along with a tutorial on using it.

Sifting The Bag of Words

Key Links
- Sample data set: A set of anonymous bostonenses from the Fall 2012 offering of 6.002 (Circuits & Electronics).
- Sample script: A Python script which loads the data set, runs it through the Bag of Words model under a few different conditions, and prints the top 10 topics from each one.

Introduction
In this module we'll be experimenting with the 'Bag of Words' model, as described by the video. Luckily, the scikit-learn library provides us with a ready-to-run implementation of the model. Even better, it automatically removes punctuation and common English words like 'the', which would otherwise overwhelm more interesting words describing the data set.

Write your own Python file going through each of these steps as you work on the tutorial, as you'll get a better understanding of what's happening in the sample script. In other words, use the sample as a base to write your own version. You can check these scripts by

Figure 4-4: Detailed view for the Bag of Words lesson.

The blog posts are written using Markdown[6], a straightforward markup for writ-
ing HTML. Videos are stored in the MariaDB database and are attached to the Ruby ActiveRecord Lesson class using the paperclip gem. The comments section acts as an additional gamification affordance by helping learners interact with each other, boosting their sense of relatedness. It is provided by Disqus, a ubiquitous platform for plug-and-play comments in blogs.

The administrator interface for managing lessons is transparently baked into the interface with links that are only visible to users with the proper privileges.

![Figure 4-5: Admin form for creating and editing lessons.](image)

The list page includes a "Create New Lesson" link and each lesson listing has links to edit or delete it. The form for creating and editing lessons is identical, its fields are pre-filled with values in the case of an edit. The form contains the minimal interface required to fully describe a lesson. A file upload input lets the user supply a video, a text area holds the main blog post comment, two normal inputs for the lesson title
and description, and one number input to specify the minimum lesson level.

4.5 Lesson Content

The lessons (or Learning Modules) were roughly designed with the guidelines from Spyropoulou in mind, with the exception that each unit of content was meant to be one level rather than one week. [12] Teach2Learn is not an actual course, so following a specific pace is not as important. Each lesson includes a short (less than two minutes) video which explains the fundamental idea behind a machine learning technique. The videos don’t include mathematical derivations, but they intuitively explain the core principle which is at work. Each video was structured to answer three questions: (1) "What’s the technique for?", (2) "What does it do?", and (3) "How do I do it?" This immediately motivates the purposes of the knowledge, exposing users to the problems which natural language programming can solve.

The lessons covered a broad introducton to NLP, a description of the Bag of Words model, and a sample use of Latent Semantic Indexing. Each of the lesson videos hinged on a key visual which demonstrated the fundamentals of the concept, like an algorithm classifying samples, a sentence being converted to a dictionary, or similar words merging into topics.

Each module also includes a short written exercise which they can implement themselves, essentially a blog post demonstrates how to use the technique in Python via the scikit-learn and gensim libraries. The post also comes with a script and sample data set which users can download to test and immediately see results on their own computers, letting them practice the knowledge they just learned. The sample data sets were drawn from the same corpuses that students were labeling. Finally, each lesson has a comments section (implemented via Disqus) for users to ask questions and relate what they did with the content.
Figure 4-6: The introductory "What is Natural Language Programming?" lesson video shows a "smart folder" learning to understand new samples from previous labeled ones.

Figure 4-7: The "Bag of Words" converting the sample sentence into its internal representation.

4.6 Usability Testing

After adding these affordances, we set out to perform some user testing. The user interviews were conducted by sitting next to the user, asking them to say what they were thinking as they use the application, and instructing them to otherwise use it
naturally. They each knew that they were in the role of a Facebook user who had arrived here because they wanted to learn about data science and went from there.

The first usability issue arose in the Lessons page. Two out of the three lessons start out locked, so their entries are grayed out and disabled. However, initially there wasn’t a clear affordance explaining that the solution was to go to the labeling UI and earn some points. This was resolved by adding a text overlay with a link to the labeling UI, as shown in Figure 4-3.

Within the lesson, the Disqus comment box was a mixed blessing. One of the interviewees navigated to the first lesson page and ended up engaging with the comment box for ninety seconds before looking at anything else. Disqus embeds a number of additional buttons and menus within their box which make it easy to navigate into more content on their network.

While it wasn’t a frequent issue, a simpler Rails comment library would probably be a good choice in the future. This is an excellent example where features which help the user relate can also distract them from the core content.

A few problems also arose in the Labeling interface. The key issue was that users found this site with no prior training or experience with reading these data sets. Thorough project instructions are critical to helping the users label successfully. The
meanings of the different labels were never clearly explained by the software in testing, it was left on the user to infer which ones would apply when. Unsurprisingly, a few of the first in-person testing labels were incorrect.

There was a two-pronged solution. The initial instructions were made to more clearly explain where these posts are from and what sorts of things they would be reading, helping the users contextualize the labeling process. The second aspect was to add more direct help within the labeling menu.

Each label and category is now augmented with help text which appears in a tooltip on hover. This text can explain the label and offer examples right as the
user needs them, providing information only when it’s wanted. In a follow-on user trial with new participants and these new affordances, users were able to label their samples correctly.
Chapter 5

Evaluation

We evaluated Teach2Learn by advertising it on social media and analyzing the resulting user engagement. Results showed that about 5% of users who found the site created an account and 20% of signed-up users created enough labels to unlock all of the available content. The platform does not seem to be a viable replacement to MTurk, as paid advertisements did not efficiently convert to labels, but it has much potential as an additional learning resource in a classroom setting.

5.1 Driving Growth

With Teach2Learn now ready to go, the last step was to recruit users and see if it could be a functional competitor to MTurk. I reached out to my personal social media network and ran paid advertising campaigns on Facebook and reddit.com. Facebook link tracking and Google analytics were both used to determine effectiveness. The paid Facebook campaign targeted English speaking men and women between the ages of 18-55, located in the US, who were marked as technology early adopters and followed a set of data-related interests1. The reddit campaign system does not allow

1Workers in the Computer and Mathematics, IT and Technical, and Sales industries; individuals with the job titles of BI Analyst, BI Developer, Business Intelligence, Business Intelligence Analyst, Business Intelligence Consultant, Business Intelligence Manager, Data Analyst, Data Entry Clerk, Data Entry Technician, Data Specialist; individuals who expressed interests in big data, entrepreneurship, management, marketing, science, digital marketing, display advertising, email marketing, online advertising, search engine optimization, social media, social media marketing,
for the same granularity, but targeted individuals interested in programming.

5.2 Quantitative Results

Teach2Learn was tested over a two week period via both paid and unpaid posts on Facebook and reddit.com. The site uses Google Analytics to track the total number of sessions and page views per site location. Google Analytics lets you track the number of unique pageviews (the number of sessions which viewed a given page) to each site location, letting us map how many people explored each part of the site. It also provides additional information like what nations users came from and what language their browser was set to use, giving us some idea of where the users recruited from reddit (as that platform had no way to select for language or location) came from. We were able to analyze label creation by inspecting the relevant database table. The only available way to track lesson engagement was checking how many page views each lesson received. Paid campaigns provided statistics about how many users they converted, letting us easily track their effectiveness.

5.2.1 Onboarding & User Characteristics

<table>
<thead>
<tr>
<th>Site Location</th>
<th>URL</th>
<th>Number of Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landing Page</td>
<td>/</td>
<td>315</td>
</tr>
<tr>
<td>Sign In</td>
<td>/users/sign_in</td>
<td>68</td>
</tr>
<tr>
<td>Sign Up</td>
<td>/users/sign_up</td>
<td>33</td>
</tr>
<tr>
<td>Labeling Page</td>
<td>/responses/new</td>
<td>15</td>
</tr>
<tr>
<td>Lessons Page</td>
<td>/lessons</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5.1: Total views for each page as drawn from Google Analytics.

?? shows the total number of pageviews for different locations on the site as pulled from the Google Analytics suite, showing results from all Teach2Learn visitors. The lesson and label pages redirect to the sign-in screen if the user is logged out, explaining why there were more hits to the sign in page than the sign up page. Approximately web design, web development, web hosting, and software.
20% of the users who found the site were interested enough to try and explore the core pages which triggered this redirect, and roughly 10% continued to the sign-up page. Inspecting the users table shows that 18 users completed the sign-up process, suggesting that 15 users found the sign-up screen but did not complete it. All of the users had their emails confirmed, so no users were lost in the "Confirm Your Account" process.

Table 5.2: Teach2Learn Visitor Nationalities & Languages

<table>
<thead>
<tr>
<th>Nation</th>
<th>User Count</th>
<th>Total %</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>171</td>
<td>51.98%</td>
</tr>
<tr>
<td>Netherlands</td>
<td>20</td>
<td>6.08%</td>
</tr>
<tr>
<td>Canada</td>
<td>17</td>
<td>5.17%</td>
</tr>
<tr>
<td>Brazil</td>
<td>13</td>
<td>3.95%</td>
</tr>
<tr>
<td>India</td>
<td>13</td>
<td>3.95%</td>
</tr>
<tr>
<td>Germany</td>
<td>10</td>
<td>3.04%</td>
</tr>
<tr>
<td>France</td>
<td>7</td>
<td>2.13%</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>7</td>
<td>2.13%</td>
</tr>
<tr>
<td>Australia</td>
<td>6</td>
<td>1.82%</td>
</tr>
<tr>
<td>Norway</td>
<td>5</td>
<td>1.52%</td>
</tr>
</tbody>
</table>

Table 5.3: User Nationalities

<table>
<thead>
<tr>
<th>Language</th>
<th>User Count</th>
<th>Total %</th>
</tr>
</thead>
<tbody>
<tr>
<td>English²</td>
<td>284</td>
<td>86.3%</td>
</tr>
<tr>
<td>German</td>
<td>10</td>
<td>3%</td>
</tr>
<tr>
<td>French</td>
<td>6</td>
<td>1.82%</td>
</tr>
<tr>
<td>Spanish</td>
<td>5</td>
<td>1.51%</td>
</tr>
</tbody>
</table>

Table 5.4: User Languages

Examining the breakdown of user locations showed that while the majority of users were from the United State, as expected, many users hailed from other nations around the world. The set of user languages was smaller, however, with a far larger majority of the users speaking English. We cannot easily determine whether any of the users who created labels were from non-US nations, but the interest from international users suggests that Teach2Learn has global appeal.

5.2.2 Label Creation

The 18 users created a total of 95 labels, most of them made by a small set of users. When these users found the site, they had to label 15 samples ³ in order to unlock all the available lessons. Four users completed enough labels to unlock all of the content,

---

²Including US, Great Britian, and Canadian dialects.
³5 labels for the first lesson, 10 for the second lesson.
<table>
<thead>
<tr>
<th>Number of Users</th>
<th>Number of Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.5: Labeled Samples Per User

one user unlocked the first lesson, and the majority made no labels at all. 4 of the 17 users, or 1.5% of all 315 site visitors, were willing to put in the amount of time required to unlock all of the content.

In order to examine whether the users were creating valid labels, I examined each response by hand to assess whether it was valid. User responses were scored with 1 point if they were entirely accurate, 0.5 points if only one of the labels was incorrect, and 0 points if all the labels failed to match the post.

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Label Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entirely correct</td>
<td>67</td>
</tr>
<tr>
<td>Mostly correct</td>
<td>23</td>
</tr>
<tr>
<td>Entirely incorrect</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.6: Breakdown of Crowd-sourced Label Accuracy

This breakdown yielded a net labeling accuracy of 78.5/95 or ~ 83%. This is a mediocre level of accuracy, and is lower than desired. Some loss of accuracy is to be expected given that these users are seeing the data set for the first time and aren’t actively trained to label it, but an accuracy level above 90% would be preferable. Accounting for users mislabeling samples is a potential avenue of future work.

5.2.3 Lesson Engagement

Eight users visited the Lessons page, equal to the number of users who unlocked at least one lesson plus three more. The majority of those users unlocked both of the available lessons, suggesting that the educational content was a sufficient incentive
to drive sustained action among those users who were interested in the topic. None of the lessons received a comment from a user. A key limiting factor was the lesson volume, there were only two lessons beyond the introduction. The subset of users who unlocked all the lessons were emailed a follow-up survey to see what they thought of the lessons, but none responded.

The lessons seem to have sparked user interest, as a non-trivial number of people joined Teach2Learn and labeled samples in the interest of unlocking them. Many users, however, failed to follow through and try them. This aligns with a literature review on MOOC dropout rates which showed an average of 7.5% of users actually complete the MOOCs they sign up for[5]. It would be ideal if all the users engaged with the lesson content, but it’s encouraging that users were willing to perform labels based on the simple promise of lessons – people want to learn about this topic and are willing to spend time unlocking them, even if they never get around to using them. Whether or not people engage with the lessons, the platform is a success for researchers so long as users create labels.

### 5.2.4 Advertisement Response

<table>
<thead>
<tr>
<th>Referral Source</th>
<th>Money Spent</th>
<th>Impressions</th>
<th>Click-throughs</th>
<th>Sign-Ups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook Post</td>
<td>$0</td>
<td>Unknown</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Facebook Ad</td>
<td>$25</td>
<td>1,618</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>reddit post</td>
<td>$0</td>
<td>Unknown</td>
<td>119</td>
<td>11</td>
</tr>
<tr>
<td>reddit ad</td>
<td>$25</td>
<td>59,115</td>
<td>42</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.7: User sign-ups per advertising medium.

Paid advertising ended up being surprisingly ineffective for driving sign-ups. Advertisements on Facebook and reddit spent their budget on impressions, but Facebook yielded no sign-ups and reddit yielded only one. The most effective strategy was regular, unpaid posts to relevant reddit.com subreddits⁴. The bulk of user sign-ups came from these reddit posts, including all of the users who ended up creating labels. The

---
⁴/r/compsci, r/datascience, r/MachineLearning, r/python
reddit ads were placed in a similar set of subreddits as the unpaid posts, so the different levels of engagement were likely a consequence of users instinctively ignoring the paid posts.

5.3 Qualitative Insights

The following sections describe qualitative insights based on analyzing the quantitative results.

5.3.1 Growing the Collaborative Platform

Results show that the majority of labeling users made all of their labels in one sitting, not returning to the site. Ideally, the site would bring labelers back for future projects in a self-sustaining system. With the current small set of lessons, however, users were able to make enough labels to unlock them all in one go. Expanding the set of available lessons could help remedy this, increasing the total number of labels required to see all content. Teasing more valuable lessons, perhaps by demonstrating a practical skill they’ll teach the user, could also inspire continued engagement with the platform. Retention strategies like sending emails to users informing them about new content could also work to bring back users for more labeling.

This would not resolve the underlying problem, however, that the model of labels for lessons is not inherently self-sustaining. Users are incentivized to make the number of labels required to unlock all lessons, no more. This weakness hinges on the fact that lessons are an extrinsic incentive, they do not make the labeling work more intrinsically valuable. There will always be a finite supply of lessons, so some users will always reach the point where they have exhausted the available content. The most direct remedy to this issue is to advertise to groups which already think data science work is valuable, already possessing some intrinsic motivation to label. Academics and industry researchers could potentially fall into this category. An indirect remedy would be to add additional rewards for users who go above and beyond, creating more labels than necessary to unlock content. These could take the form of a tiered list
of contributors (e.g. "Gold Contributors", "Silver Contributors", etc.) recognizing the top users, granting them additional commendations such as acknowledgement in papers or signed thank you letters from researchers.

The results of paid advertisements suggest that individuals outside that group aren’t likely to sign up. Facebook ads turned out to not be useful, leading to no actual signed up users. This hearkens back to the question of situational context as described by Deterding[2]. People browsing their Facebook feed are typically bored and looking for something entertaining. On the other hand, posts on appropriate reddit.com sub-reddits ended up driving the bulk of our sign-ups. This made sense, as users who are looking at a forum for data science are actively engaged with those ideas and could be open to spending some of their time on them – exactly what this platform needs.

5.3.2 Viability Compared to Mechanical Turk

Teach2Learn boils down to a trade-off: spend less money, spend more time. The platform is a viable way to gather data, but it depends on users somehow being introduced to the site and choosing to spend their time on it. On the other hand, MTurk lets you pay people to complete work and get immediate results. This boils down to a slower total rate of data collection. One strong benefit of Teach2Learn, however, is that lesson content is free to share once it has been created. There is no marginal cost to teaching more users once the lessons are there, so the platform has potential for exponential returns on investment. As more lesson content is created, both new and old users will have incentive to create more labels. This stands in contrast to MTurk, where each and every label will cost the same amount of money.

The platform would be best suited for applications where there is a long-term period of data collection, such as longitudinal studies, because it allows time for more users to organically find the site. This makes it easy to get labeled data without spending money or worrying about running an MTurk job. Paid advertising on Facebook and reddit was ineffective, so it seems like there isn’t yet a good way to pump money into this platform and get results – MTurk is still the best solution if
researchers have the money to spend.

5.3.3 Viability as a Learning Tool

On its own, the Collaborative Platform is not well-equipped to act as a Massive Online Open Course. Standard features of a course, like quizzes, aren’t present. Accessibility features, like a colorblind mode for videos, are also missing. As with many MOOCs, the average user tended not to engage with the lesson content.

On the other hand, the platform could serve as an excellent accompaniment to a course. This tool could help students get a better understanding of a dataset while contributing to work on it. A professor could post lesson videos, have student complete labels to unlock them, and then use the dataset in a project at the end of the class. They could also set up a more direct incentive in the classroom, such as giving students study guides or extra credit based on the number of label they create. This strategy might end up being one of the best approaches for using the platform.
Chapter 6

Discussion

The core idea behind this research proved true. Users who engaged with the data labeling task produced exactly enough labels to unlock all the available content, suggesting that education was a sufficient incentive. Unfortunately, social media advertising was an ineffective strategy for finding new users. Testing developed a promising lead by identifying high rates of user interest among users in data science communities. It also demonstrated that crowd-sourced labeling efforts yield improperly labeled data, and that supplying on-the-spot instruction improved accuracy — a valuable insight for follow-on research. Future researchers should explore how Teach2Learn can be deployed in classrooms and how to account for the imperfectly labeled samples.

6.1 Key Take-Aways

The most important takeaway from this attempt to use gamified education to incentivize data labeling is that for a significant subset of the users, it worked. The experiment generated 95 labels, most of them created by 5 of the 18 users who signed up. The "Pareto principle" is at work here, where a small subset of the users (27%) produced a majority of the output (91%). These users created exactly enough labels to unlock all the available content, suggesting that accessing those lessons was their driving incentive (rather than altruistic goals of helping researchers). The fundamental question of this research, whether people will "pay" for educational content by
spending their time on human intelligence tasks, has borne out a positive result.

Efforts to grow the platform via social media advertising were not as successful. Paid advertisements converted to only one account and no labels. Teach2Learn is a valuable tool for individuals interested in learning about data science on their spare time, but that is a niche group. Even with precise demographic targeting tools, ads on social media platforms are ineffective as they target many users who aren’t looking for educational content at the time. The key factor driving useful results from the Teach2Learn is finding a user base of motivated learners. Lessons are only intrinsically motivating for individuals who are trying to learn. Google AdWords might produce better results by finding users looking to "learn data science", but the more direct approach is to work with classrooms and online learning environments. One strong benefit of Teach2Learn, however, is that the learning content is free to share with new users once it has been created. Whereas MTurk requires that researchers pay for each and every label, Teach2Learn requires an upfront cost to develop lessons and then reaps the reward of that investment with all future users of the site. MTurk is a quick way to gather labels, but Teach2Learn has the potential to keep generating labels after the initial investment is complete.

Finally, testing the platform highlighted a key concern for other endeavors seeking to crowd-source data labeling problems. Users who find the site have no prior experience with the data set and need to be coached on how to properly understand it. The need for training has to be balanced against minimizing friction for users who are first starting out, so the best approach is to embed training content directly into the interface where users are doing the labeling. This explanation also needs to include the general context surrounding the samples, rather than simply describing the possible tag types, as users often require an understanding of the greater context to accurately identify a sample. An introductory summary describing the data set works well for that purpose, but there’s no way to guarantee that a user reads it. If there is too much training content before users reach the labeling task, they will lose interest and go elsewhere. The best way to account for the user who rushes through screens to try and start labeling is to put the training content in front of them right
as they need it, when they’re clicking the label buttons (i.e. using tooltips).

### 6.2 Future Work

The platform is now stable and ready for public use, but the testing up until now has only scratched the surface of its potential. Researchers interested in education and data science could find fertile questions by exploring how Teach2Learn can be used in classrooms and finding ways to account for noisy training data.

Classrooms teaching computer science, both brick and mortar classrooms and digital learning platforms, are likely the optimal place to use Teach2Learn. There will be a readily available pool of students who are motivated to learn, making the lessons intrinsically valuable. This is particularly true for online learners in the developing world who don’t have access to in-person courses on this content but still want to learn about cutting-edge techniques. The platform also opens a number of opportunities for educators who want to perform hands-on data science. Researchers are able to download the labels for their data set, so an educator could upload a data set for the course, have students label it over a semester, then use the resulting training set for a final project on supervised algorithms. Ideally, instructors would be able to submit their own lessons and have students unlock those via labeling. This would require (1) giving researcher accounts the ability to also upload lessons, and (2) an alternate system where users choose which lesson they unlock at each level. Giving users the decision on what lesson they unlock next would also be beneficial as it would give them a greater sense of autonomy, controlling their path of learning.

Opportunities within the classroom aside, one outstanding issue with Teach2Learn is that the labels produced by an arbitrary set of users are not perfect. Some of the forum posts are ambiguous and some of the label tag names can be confusing (e.g. "Missing Data" was misinterpreted in a small set of responses), so mislabeled samples are inevitable. One possible strategy to remedy this would be presenting the users with "test" samples at predetermined intervals. These samples would be pre-labeled by the instructors, preferably at least one from each label category/type pair, and
would let us assess whether the user tended to give the correct response. This could then be used to build a confidence rating of that user's responses, which could in turn be fed into an appropriately designed supervised learning algorithm.
Appendix A

Data Schemas

This appendix describes the structure and source of all the data within the Collaborative Platform.

A.1 Sample Data

First, the sample labeling data was drawn from the Fall 2012 offerings of 6.002x and 6.00x on edX. The data from these courses had been cleaned and transitioned into the MOOCdb data schema [13], such that the forum posts were now located in the Collaborations table.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>String</td>
<td>&quot;Thank you! I have to confess, I'm a bit intimidated by the course, but I'll work as hard as I can to success at it!&quot;</td>
<td>The body text of one post from an edX class forum.</td>
</tr>
</tbody>
</table>

Table A.1: Project Upload Data Schema

Each forum post was one row in the table, and its text was contained in the collaboration_content column. This field was exported from both databases into a one-column CSV file. The MOOCdb schema separates personally identifying user data from the rest of the content such that we're guaranteed not to have the posts point back to a user.
A.2 Collaborative Platform Data

Figure A-1: Collaborative Platform Entity Relationship Diagram
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing, table-unique primary key.</td>
</tr>
<tr>
<td>email</td>
<td>varchar(255)</td>
<td><a href="mailto:johno@mit.edu">johno@mit.edu</a></td>
<td>User sign up email.</td>
</tr>
<tr>
<td>encrypted_password</td>
<td>varchar(255)</td>
<td>$2a$10$NO...</td>
<td>Encrypted password.</td>
</tr>
<tr>
<td>reset_password_token</td>
<td>varchar(255)</td>
<td>$2a$10$NO...</td>
<td>Code to confirm password reset.</td>
</tr>
<tr>
<td>reset_password_sent_at</td>
<td>datetime</td>
<td>2017-01-21 21:01:27</td>
<td>Time of last reset email.</td>
</tr>
<tr>
<td>sign_in_count</td>
<td>int(11)</td>
<td>5</td>
<td>Number of times signed into site.</td>
</tr>
<tr>
<td>last_sign_in_at</td>
<td>datetime</td>
<td>2017-01-21 21:01:27</td>
<td>Time of last session sign-in.</td>
</tr>
<tr>
<td>current_sign_in_ip</td>
<td>varchar(255)</td>
<td>18.111.25.80</td>
<td>IP of current session sign in.</td>
</tr>
<tr>
<td>last_sign_in_ip</td>
<td>varchar(255)</td>
<td>73.234.153.121</td>
<td>IP of last session sign in.</td>
</tr>
<tr>
<td>name</td>
<td>varchar(255)</td>
<td>John O’Sullivan</td>
<td>Full name of user.</td>
</tr>
<tr>
<td>confirmation_token</td>
<td>varchar(255)</td>
<td><em>2ygCL4Um_L9u4t-Drq</em></td>
<td>Token to confirm account.</td>
</tr>
<tr>
<td>confirmed_at</td>
<td>datetime</td>
<td>2017-01-13 00:21:03</td>
<td>Confirmation timestamp.</td>
</tr>
<tr>
<td>confirmation_sent_at</td>
<td>datetime</td>
<td>Short example</td>
<td>Time of confirmation email.</td>
</tr>
<tr>
<td>role</td>
<td>int(11)</td>
<td>1</td>
<td>Role of user, where 0 is normal and 1 is admin.</td>
</tr>
<tr>
<td>approved</td>
<td>tinyint(1)</td>
<td>0</td>
<td>Whether user is approved to upload projects.</td>
</tr>
<tr>
<td>current_post</td>
<td>varchar(255)</td>
<td>81936:0</td>
<td>Current selected post in LabelMe-Text.</td>
</tr>
<tr>
<td>current_upload</td>
<td>text</td>
<td>19</td>
<td>Current selected upload in LabelMe-Text.</td>
</tr>
<tr>
<td>username</td>
<td>varchar(255)</td>
<td>johno</td>
<td>Unique, case insensitive, letters, numbers, and underscores, username.</td>
</tr>
</tbody>
</table>

Table A.2: User Schema Pt. 1
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile_picture</td>
<td>varchar(255)</td>
<td>square.jpg</td>
<td>Filename of profile picture.</td>
</tr>
<tr>
<td>_file_name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>profile_picture</td>
<td>varchar(255)</td>
<td>image/jpeg</td>
<td>MIME content type.</td>
</tr>
<tr>
<td>_content_type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>profile_picture</td>
<td>int(11)</td>
<td>923481</td>
<td>Filesize in bytes.</td>
</tr>
<tr>
<td>_file_size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>profile_picture</td>
<td>datetime</td>
<td>2017-01-18 00:26:12</td>
<td>Last profile picture update timestamp.</td>
</tr>
<tr>
<td>_updated_at</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>location</td>
<td>varchar(255)</td>
<td>Cambridge, MA</td>
<td>Geographic location of user. (Optional)</td>
</tr>
<tr>
<td>url</td>
<td>varchar(255)</td>
<td>example.com</td>
<td>Personal homepage of user. (Optional)</td>
</tr>
<tr>
<td>display_tag</td>
<td>varchar(255)</td>
<td>Data Scientist</td>
<td>Tag selected to display on Users screen.</td>
</tr>
<tr>
<td>streak_count</td>
<td>int(11)</td>
<td>2</td>
<td>Number of consecutive days labeling.</td>
</tr>
<tr>
<td>streak_date</td>
<td>datetime</td>
<td>2017-01-19 12:42:00</td>
<td>Time of last streak update.</td>
</tr>
<tr>
<td>referral_source</td>
<td>varchar(255)</td>
<td>reddit</td>
<td>How user found Collaborative Platform.</td>
</tr>
<tr>
<td>unconfirmed_email</td>
<td>varchar(255)</td>
<td><a href="mailto:johno@mit.edu">johno@mit.edu</a> (or) NULL</td>
<td>Stores email until confirmed, null afterwards.</td>
</tr>
</tbody>
</table>

Table A.3: User Schema Pt. 2
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>title</td>
<td>varchar(255)</td>
<td>What is Natural Lan-</td>
<td>Name of lesson.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>guage Processing?</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>varchar(255)</td>
<td>A brief introduction to Natural Language Processing and a guide on setting up your computing environment.</td>
<td>Brief Tweet length description of lesson.</td>
</tr>
<tr>
<td>help_text</td>
<td>text</td>
<td># Setup Help Text ...</td>
<td>Long form lesson blog post.</td>
</tr>
<tr>
<td>minLevel</td>
<td>int(11)</td>
<td>0</td>
<td>Level required to access lesson.</td>
</tr>
<tr>
<td>status</td>
<td>varchar(255)</td>
<td>1</td>
<td>Whether lesson is published.</td>
</tr>
<tr>
<td>video</td>
<td>blob</td>
<td>...</td>
<td>Video binary.</td>
</tr>
<tr>
<td>video_file_name</td>
<td>varchar(255)</td>
<td>What_is_NLP.mp4</td>
<td>Filename of video.</td>
</tr>
<tr>
<td>video_content_type</td>
<td>varchar(255)</td>
<td>video/mp4</td>
<td>MIME type of video.</td>
</tr>
<tr>
<td>video_file_size</td>
<td>int(11)</td>
<td>65418520</td>
<td>Video size in bytes.</td>
</tr>
</tbody>
</table>

Table A.4: Lesson Schema
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>csv_file_name</td>
<td>varchar(255)</td>
<td>corpus_6_002x_2012_fall.csv</td>
<td>Name of CSV samples file.</td>
</tr>
<tr>
<td>csv_content_type</td>
<td>varchar(255)</td>
<td>application/vnd.ms-excel</td>
<td>MIME content type of samples.</td>
</tr>
<tr>
<td>csv_file_size</td>
<td>int(11)</td>
<td>5038778</td>
<td>Size of samples in bytes.</td>
</tr>
<tr>
<td>csv_updated_at</td>
<td>datetime</td>
<td>2016-02-12 19:46:28</td>
<td>Timestamp of last update to samples.</td>
</tr>
<tr>
<td>header</td>
<td>tinyint(1)</td>
<td>1</td>
<td>Binary boolean to detect csv header row.</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>25</td>
<td>ID of upload user.</td>
</tr>
<tr>
<td>column</td>
<td>int(11)</td>
<td>0</td>
<td>Column (zero-) index of samples.</td>
</tr>
<tr>
<td>instructions</td>
<td>text</td>
<td>The following are posts from</td>
<td>Brief description</td>
</tr>
<tr>
<td>sentences</td>
<td>tinyint(1)</td>
<td>0</td>
<td>Binary boolean to label corpus by post (0) or sentence (1).</td>
</tr>
<tr>
<td>name</td>
<td>varchar(255)</td>
<td>6.002x Forum Posts</td>
<td>Title of upload.</td>
</tr>
<tr>
<td>description</td>
<td>text</td>
<td>Our specific research question for this work was...</td>
<td>User-facing upload description.</td>
</tr>
</tbody>
</table>

Table A.5: Upload Schema

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>upload_id</td>
<td>int(11)</td>
<td>18</td>
<td>Foreign key to label category’s upload.</td>
</tr>
<tr>
<td>content</td>
<td>varchar(255)</td>
<td>Topic</td>
<td>Text of label</td>
</tr>
<tr>
<td>helptext</td>
<td>varchar(255)</td>
<td>What was the poster talking about in this post?</td>
<td>Text for label tooltip.</td>
</tr>
</tbody>
</table>

Table A.6: Label Category Schema
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>content</td>
<td>varchar(255)</td>
<td>Content</td>
<td>Text of tag.</td>
</tr>
<tr>
<td>label_category_id</td>
<td>int(11)</td>
<td>188</td>
<td>Foreign key to tag's category.</td>
</tr>
<tr>
<td>helptext</td>
<td>varchar(255)</td>
<td>Discussing concepts from the course, e.g.</td>
<td>Text for tag tooltip.</td>
</tr>
</tbody>
</table>

Table A.7: Label Tag Schema

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>content</td>
<td>text</td>
<td>I would agree with the first and third case. I have to disagree with ...</td>
<td>Text of post sample.</td>
</tr>
<tr>
<td>upload_id</td>
<td>int(11)</td>
<td>22</td>
<td>Foreign key to post's upload.</td>
</tr>
</tbody>
</table>

Table A.8: Post Schema

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>content</td>
<td>text</td>
<td>Yes, it makes sense.</td>
<td>One sentence from sample post.</td>
</tr>
<tr>
<td>post_id</td>
<td>int(11)</td>
<td>219725</td>
<td>Foreign key to sentence's post.</td>
</tr>
</tbody>
</table>

Table A.9: Sentence Schema
Table A.10: Response Schema

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>38</td>
<td>Foreign key to labeling user.</td>
</tr>
<tr>
<td>sentence_id</td>
<td>int(11)</td>
<td>NULL</td>
<td>Foreign key to sentence, NULL if label on post.</td>
</tr>
<tr>
<td>label</td>
<td>text</td>
<td>- Topic @ Content - Topic @ Course Website - Technology - Role of Poster @ Giving help or information</td>
<td>Labels applied to sentence or post, &quot;- [Category] @ [Tag]&quot;, separated by newlines.</td>
</tr>
<tr>
<td>post_id</td>
<td>int(11)</td>
<td>46539</td>
<td>Foreign key to response's post.</td>
</tr>
</tbody>
</table>

Table A.11: Post User Schema

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>42</td>
<td>Auto-incrementing table-unique primary key.</td>
</tr>
<tr>
<td>user_id</td>
<td>int(11)</td>
<td>25</td>
<td>Foreign key to user.</td>
</tr>
<tr>
<td>post_id</td>
<td>int(11)</td>
<td>4088</td>
<td>Foreign key to post.</td>
</tr>
</tbody>
</table>
Appendix B

User Experience Flow

The following series of figures documents the average new user’s path through the website, with captions describing the transition actions.
Label Text Samples

Computers often need training data to learn from, but only humans can read and label that data. You can help researchers around the world by labeling their datasets.

Figure B-1: The user finds our landing page via some referral source.
researchers around the world by labeling their datasets.

Learn Data Science
In return, you'll unlock data science lessons which teach practical knowledge. Each one briefly explains a key tool, gives you some sample data, and shows you how to use it.

Figure B-2: If the user scrolls down, they will find more details describing the lessons in the Collaborative Platform.

Figure B-3: If user tries to open Lessons without logging in, they're redirected. They click Sign Up to create an account.
Figure B-4: User fills out the new account form with normal account details, a bio, and a list of self-descriptive tags.

Confirmation instructions

Collaborative Platform: <alyssa.p.hacker1@gmail.com>

Jan 19 (2 days ago)

Welcome to the Collaborative Platform, alyssa.p.hacker1@gmail.com!
You can confirm your account email through the link below:

Confirm my account

Happy Learning,
Alyssa & the Collaborative Platform Team

Click here to: Reply or Forward

Figure B-5: User receives a confirmation email in their specified inbox.
Label Text Samples
Computers often need training data to learn from, but only humans can read and label that data. You can help researchers around the world by labeling their datasets.

Figure B-6: User is now logged in, headline actions become links.

Lessons

What is Natural Language Processing?
A brief introduction to Natural Language Processing and a guide on setting up your computing environment.

You haven't unlocked this lesson yet; try making some labels!

You haven't unlocked this lesson yet; try making some labels!

Figure B-7: User goes to look at their available data science lessons.
What is Natural Language Processing?

A brief introduction to Natural Language Processing and a guide on setting up your computing environment.

Installing Python

These lessons are going to use Python 3.5, the most up-to-date version of Python. If you already have a Python installation, make sure to check which version you’re on. If you’re a Windows user, make sure to check the version number of your Python installation.

Installing Packages

These lessons and their follow-ons will be using four major libraries: `nltk`, `texy`, `smtk`, `nltk`: The installation process is slightly different, depending on whether you’re a Mac or Windows user.

Unix (Mac & Linux)

Figure B-8: User explores introductory content.

6.002x Forum Posts

It means you have to include enough decimal places in the answer such that it is within 1% of the exact (infinite precision) answer. So if the correct answer is x, your answer has to be between x-0.001*x and x+0.001*x.

Role of Poster

Seeing help or information

Getting help or information

Other

Progress to Next Level

Figure B-9: User exhausts content and starts labeling to unlock next lesson.
Bibliography


