Watch & Learn: Language Learning Application for Anime Enthusiasts

CS 160 Final Project | Summer 2018

Marinella Bohol
mbohol@berkeley.edu

Christian Cayabyab
ccayabyab@berkeley.edu

Helen Lee
ewoolee@berkeley.edu

Shirley Li
lishirley71@berkeley.edu

Sanket Padmanabhan
sanketq@berkeley.edu

アニメ
Watch & Learn
Thoughtless: without consideration of the possible consequences.
Correct Answers
1. What are the names of the main characters?
Incorrect. The correct answer is 1
Your choice: 0
Kirito and Nagisa
level 5

Figure 1: Watch & Learn, a mobile application for learning a language with anime

ABSTRACT

This paper provides a complete guide to our process, implementation and the final evaluation of Watch & Learn from conception to finished prototype.

CCS CONCEPTS

• **Human-centered computing** → **Interface design prototyping**;

KEYWORDS

Language, Machine learning

ACM Reference Format:
Marinella Bohol, Christian Cayabyab, Helen Lee, Shirley Li, and Sanket Padmanabhan. 2018. Watch & Learn: Language Learning Application for Anime Enthusiasts: CS 160 Final Project | Summer 2018. In *Proceedings of Final Project (CS 160).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Motivation

Learning a new language can be beneficial for a person's social life, career, and cognition. Being able to communicate with foreign language speakers increases the likelihood of new friendships. Employers will find polylingual people to be great assets when trying to negotiate with foreign companies. Lastly, cognitive benefits include a boost in creativity, memory, and intelligence.

Although learning a new language is clearly advantageous, it is a difficult task as many obstacles stand in the way. Language classes are costly and can conflict with schedules. Learning on one's own through textbooks requires patience, discipline, and motivation to go through them on a daily basis. People with busy schedules would find these methods hard to incorporate into their life. In addition, people would rather have their free time spent on activities that entertain and interest them. Among those who face these obstacles are people with an intermediate proficiency and a desire to achieve fluency in their non-primary language. As a result, many of these individuals develop self-paced methods of learning a language which have been tailored to meet their schedules and needs. Through our research, we learned that many people who fall under this category are also anime enthusiasts who use anime as an immersion tool to help them effectively learn a new language.

Other language learning apps like Duolingo and Busuu have lesson plans with quizzes and exercises that can be repetitive and dull, especially if the subject or topic is not interesting. Our app, Watch & Learn, is different in that it

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CS 160, August 2018, Berkeley, CA USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

combines immersion and quizzes. Users are able see and hear how words are used in a particular context that entertains them. Afterwards, material is immediately reinforced through the quizzes. In this way, Watch & Learn allows users to be motivated and productive when trying to reach their fluency goals.

Overview: The Design Process

Our design process consisted of three cycles.

The first cycle focused primarily on ideation, interviews, personas, and identifying our user group. During our ideation phase, we spent two days creating new ideas and extending existing ideas for a mobile application. From our brainstorm, we were able to categorize our ideas and narrow down what we wanted to pursue to language learning. Initially, our application idea was to have users post stories that would be of varying difficulty. After this, each team member interviewed someone that is learning a language and not within our age group. From there, we found that users preferred entertaining media such as videos or music over textbooks and classes for reasons regarding time and motivation. We then synthesized our results and created a persona based on our interview data. We wanted to use machine learning to determine the difficulty of the content that users would get. Because we imagined it would be much harder to use machine learning on pictures or videos, we had decided early on to use text as the medium. However, we were advised to keep our options open, and with another feedback to narrow down our user group, we decided to use anime rather than text.

For our second cycle, we created interaction sketches, paper prototypes, and wireframes. Our team sketched out several interaction ideas and integrated features from the refined sketches to help form a prototype. One member sketched out the interface wireframe according to the features from our refined sketches. We then made a paper prototype of the wireframe for the next cycle of evaluation.

In cycle three, we conducted Wizard of Oz trials and a cognitive walkthrough. These evaluation techniques gave us valuable insights on what to add to and subtract from our interface. For instance, users preferred for the quiz questions to have more variety. We also created a development plan and assigned roles for each portion of the implementation.

Summary of Solution and User Evaluation

We created Watch & Learn to supplement the primary learning method of intermediate language learners through immersive media, specifically anime. Our application recommends anime based on user proficiency and interests, and helps develop vocabulary, reading, and grammar skills through quizzes. Each episode features pop-up definitions of a key term to help guide the viewer. To motivate users and encourage friendly competition, we incorporated an avatar level-up system that makes the avatar image more interesting as the user progresses.

We conducted our user evaluation on users who were learning English through Google Hangout. There were some interface issues such as text being too small or pop-up definitions not being noticeable. Our definition of key terms was not clear to users in that it referred to difficult words. However, a user was pleased to have pop-up definitions conveniently placed at the top.

2 CYCLE I

Brainstorming

We wanted to create an application that utilizes storytelling that has and creates meaning. So we began by coming up with ideas from a variety of different user groups and subject areas. We grouped our ideas into categories: sports, learning, project managing, ocean exploration, film, creativity, scrapbooking, travel, astronomy, goals and motivation, pets, social good, and connecting people (figure 3). After internal voting, we narrowed down our top three categories into learning and language, stars, and film. In the end we decided on language learning and conducted our user interviews with people learning a new language. We felt that language learning was the best option to create an application that had the strongest use-case and meaning.

Observational Study

We conducted user interviews to find areas to improve upon traditional language learning methods and narrow our focus. We created a list of interview questions asking basic questions about the summary and history of their learning of a language including their method, time spent, motivation, measurement of progress, and goals. From our interviews, we found that language learners preferred to learn by watching television, movies or anime (i.e. consuming content) over going through formal vocabulary instruction and attending classes. The users felt more drawn to this method because it was fun for them. We saw a potential area for storytelling in the interviewees' desire to learn a language in order to communicate with and understand the stories of people around them.

Task Analysis

In addition to user interviews we found it necessary to perform a task analysis to better understand the user's activities in context to inform our design. We went through the standard list of Task Analysis questions: who is going to use the system, what tasks do they now perform, what tasks are desired, how are the tasks learned, where are the tasks performed, what the relationship is between the user and data, what other tools the user has, how users communicate with each other, how often tasks are performed, what are the time constraints on the tasks, and what happens when things go wrong. From our task analysis we found that we wanted to focus on users who want to learn a new language without spending any money and learn at their own pace. The activities that they currently do are translate words and sentences, read stories and get feedback from other people on Duolingo, but there is also a desire to be able to compete with oneself and friends. Users must learn through practice and repetition and need to be able to hear audio with existing applications. We found that an area of improvement would be making the application more immersive with media.

Refining User Needs

To get a better understanding of what core interactions we wanted to support with our application, we derived user needs from each of our interviews. From there, we chose user needs that we wanted to focus on: the need for an immersive, but accessible language learning environment, the need for some type of regular progress feedback while learning (levels), the ability for users to learn at their own pace, the need for entertainment through learning (anime, card games), and the ability to test their proficiency.

Interaction Sketches

Based on the refined set of user needs, We individually created interaction sketches to brainstorm what features we wanted to include, and how the user might interact with them. We did our interaction sketches separately from each other so that we could generate as many ideas as possible without influencing each other. We decided that each of us would pick three of our favorite interactions to present to the group. We decided that the core interactions of our language learning application would involve television episodes with translated subtitles along with quizzes on plot at the end of videos.

Refined Interaction Sketches

After discussing each of our initial interaction sketches with each other we sought to create some refined sketches to brainstorm more polished ideas of how the application would work. Each of us compiled and refined pieces of our sketches that we individually liked along with what the group favored of our sketches to create our own refined ideas. We found that we wanted to also include a self-assessment and in-video popups of key terms in addition to our television episodes with translated subtitles and with the quizzes on plot following videos. We refined our initial interaction sketches to support these areas (figure 4).

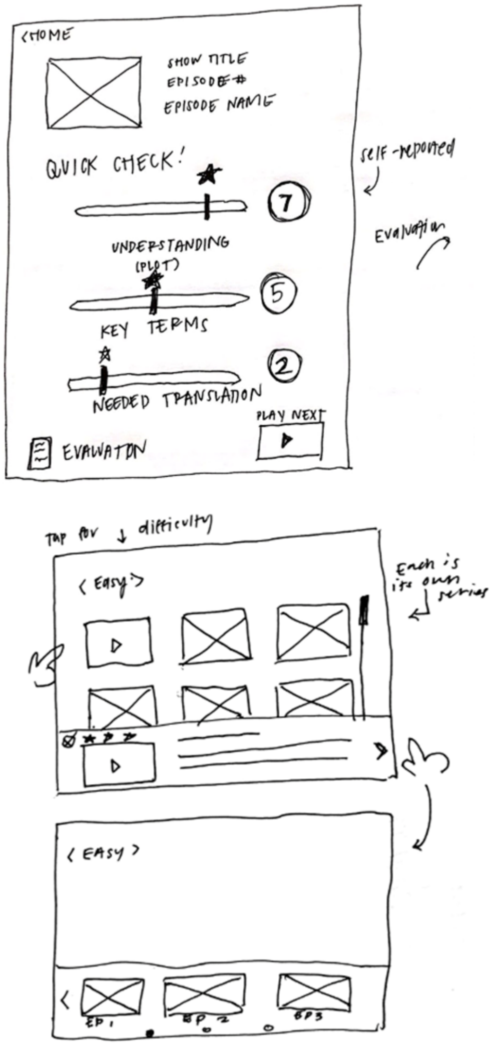


Figure 2: Initial interaction sketches

3 CYCLE II

Wireframes and Wizard of Oz Evaluation

To visualize our screens and user flow, we created wireframes from our refined sketches like the self-assessment screen, recommended show, playback, and quiz screen. Synthesizing the user need to track their progress and theme of storytelling, we added a leveling up screen to embed a story in their pursuit of fluency.

We then turned our wireframes into paper prototype to get insight on the look and feel of our application and to see if our core interactions were understood and appreciated by the user (figure 5).

We had two classmates do the self-assessment, select topics of interest, choose and watch a recommended episode and

Figure 3: All of our ideas from two days of brainstorming

take the quiz at the end. We then asked them their thoughts on the right length of the quiz, what they would like to see after getting a question wrong, pop-ups during the episode, gaining experience with an avatar, a list of vocabulary and definitions at the beginning of the episode, seeing series in succession vs. jumping around to episodes from different series, and the content of the quizzes. We found that the users wanted longer quizzes than we had planned, and an option to see a specific time in the anime where the answer for a question they answered incorrectly was. They both liked having pop up definitions, and one of them suggested having a compilation of all the key terms that would be displayed throughout the video displayed before the episode. So we decided to add a pre-episode vocabulary screen to the user flow. We also found that users would like to see the series they watch in succession and quiz questions focusing on each category that appears in our self-assessment (e.g. vocabulary, reading, writing, and grammar). Moving forward, we decided to prioritize implementing more questions in quizzes, pre-epsiode vocabulary screens, and pop-ups of terms and their definitions during episodes.

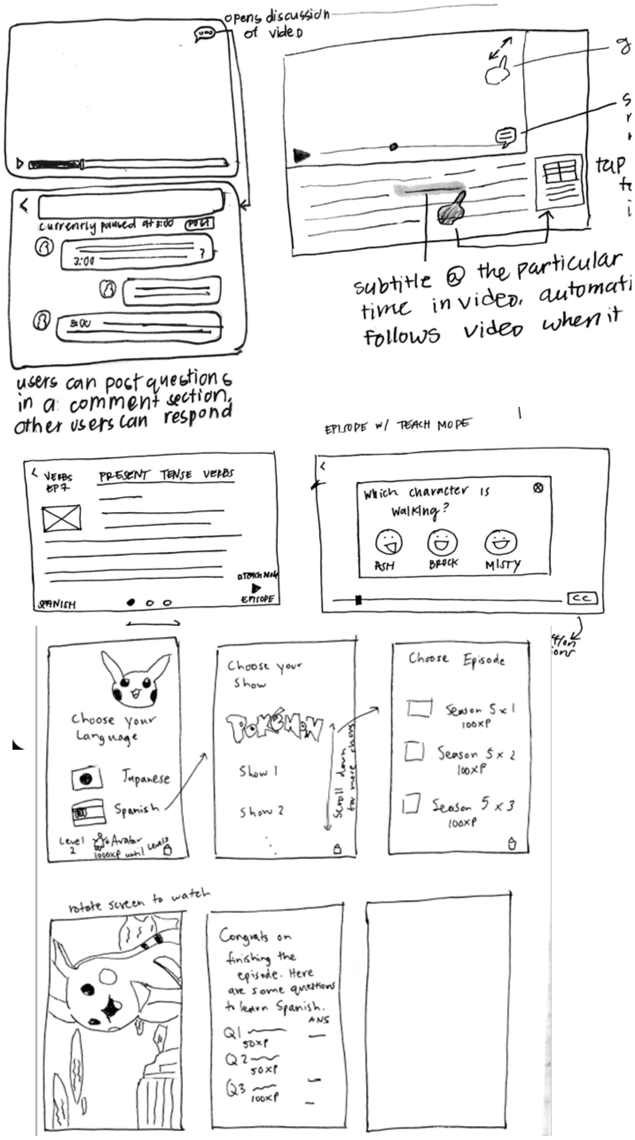


Figure 4: Refined Interaction sketches

Cognitive Walkthrough

After the Wizard of Oz evaluation, we conducted a cognitive walkthrough to simulate a user's problem-solving process at each step in using our application, to check if their goals and memory for actions can be assumed to lead to the next correct action. We broke down the steps of using the application into the following:

- completing a pre-assessment and inputting topic interests
- choosing a suggested show and episode
- navigating options in the playback screen
- taking a quiz to test comprehension

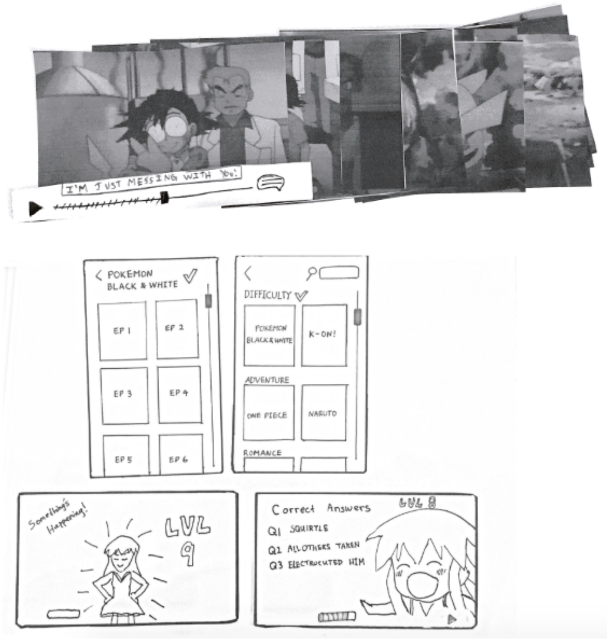


Figure 5: Original pop up with orange box

- viewing correct answers
- viewing progress including an avatar increasing their level.

For each question we asked the questions âĂIJWill the user know what to do?âĂİ, âĂIJWill the user notice that the correct action is available?âĂİ, and âĂIJWill the user interpret the application feedback correctly?âĂİ. We found that our application relies on cultural knowledge of using a mobile application for parts like using a slider bars to assign proficiency level on the self-assessment or tapping selections like the series or episode to select. For elements like choosing selected topics we saw that since we were using a skeuomorph of filling out forms or answering questions for the quiz in a regular format, these parts of our application would be understood and correctly interpreted. Correctly interpreting and using the playback controls like play/pause, time scrubbing and caption toggling would come down to whether the user has had experience streaming videos on various devices to understand what each button means (triangle to play, two vertical lines to pause, dragging the progress slider left to previous timestamps or right to go to future timestamps, and the caption icon to turn captions on and off). We assume that our users will have experience with this kind of playback interface. The concept of leveling up a character would be understood well if the user has played a video game that uses avatars leveling up to show progress.

Digital Mock-up

We wanted to replicate our prototype digitally, incorporating the feedback on interface features we got from the Wizard of Oz evaluation. Illustrator was used to create the mock up. We chose Proxima Nova as it was a legible and simple font. For our color scheme, we selected orange to be the primary color of our interface because it seems to encapsulate the bright and energetic concept of anime. Because users of our app would be in landscape mode majority of the time to view the video in full screen, we focused on displaying screens in that orientation. Those screens are the key terms, video, feedback, and level up screens. Digital mock-ups are great previews of our application and very useful guidelines for styling during implementation.

4 CYCLE III

Final User Evaluation

Our research question was âĂIJDoes Watch & Learn's content-centered assessment and annotated anime support users in developing their language proficiency?âĂİ In the user evaluation, we wanted to see if users can accomplish the following tasks:

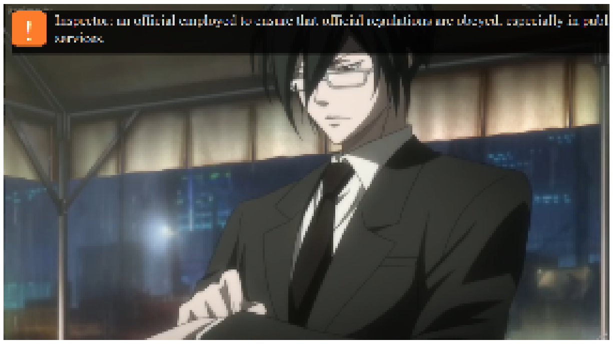
(1) select an episode to watch
(2) utilize the pre-episode vocabulary list
(3) interact with in-episode pop ups
(4) Open the word definition by tapping the alert, then exit
(5) Ignore the alert by waiting five seconds
(6) Complete a post-episode quiz and review the results

To make sure users know what to expect, we gave them a brief tutorial of how each interaction above would work. Because Watch & Learn was implemented in English, we found two users with intermediate proficiency in English. The evaluation was carried out remotely over Google Hangout, and we were able to observe their interaction with the app with the screen share feature. Both of them were able to complete all the tasks without any difficulty, but we were able to gain additional insights from their responses to the Google Form we had them complete after the interview. The first user told us that the pop ups are not prominent enough to grab their attention during the episode. This might be because the orange pop up box that we had in the mock up and earlier version of implementation(figure 6) had stopped working (figure 7).

5 SYSTEM

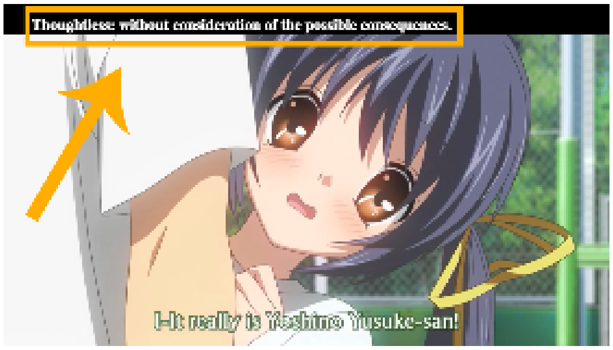
Difficulty Classification

It is important for users to feel challenged but not overwhelmed when using Watch&Learn. If a user attempts to learn a language outside of their ability, they may become



Inspector: an official employed to ensure that official regulations are obeyed, especially in publ
services.

Figure 6: Original pop up with orange box



Thoughtless: without consideration of the possible consequences.

I-It really is Yoshino Yusuke-san!

Figure 7: Pop up after unknown bug

discouraged and potentially give up on language learning entirely. To mitigate this, we must have some way of classifying each video by difficulty.

Proficiency Assessment

In order to cater content to a user's skill level, we must first learn the user's skill level. The first option to gain this information was to use an assessment. However, this kind of content wasn't possible to create given our skillsets, since training in linguistics and education would be required to create a quiz that could accurately determine a user's prior proficiency. Instead, we use sliders of each necessary statistic so that user's can self-report their level of comfort at each area of language learning.

Using Convolutional Neural Networks for Episode-Level Difficulty Classification

Our first solution, which we did not implement due to time constraints, was to use a convolutional neural network (CNN) for natural language processing (NLP). This spike was heavily influenced by the work of Yoon Kim of Cornell University [1]. Kim's CNN code can be found on github [2].

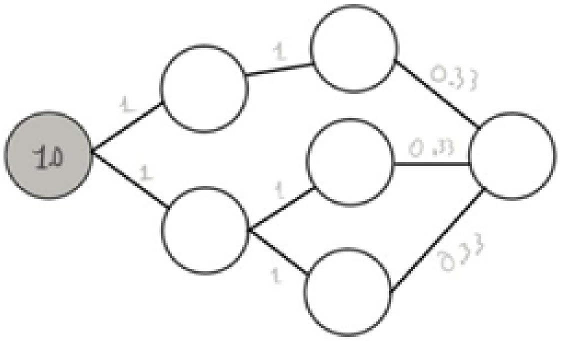


Figure 8: Graph Representation of Simple Neural Network

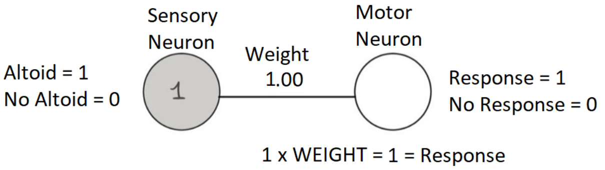


Figure 9: Dwight's initial reaction to Jim's offer of altoids

Neural Networks. A neural network is a structure in machine learning which mirrors the way the human brain functions by taking an input from a stimulus, processing it, and creating an output [3]. For example, a biological neural network (your brain) takes in visual stimulus of a brown torus object, converts that into electrical input to send to the visual cortex, processes that the object is a donut, and outputs that the object is edible. Neural networks are made up of neurons, information cells which form the basic unit of computation in the brain.

In an artificial context, a neuron is a mathematical function which receives inputs and produces outputs. We can string them together to create a neural network, as shown in Figure 8. In the graph, the nodes are neurons and the lines between them are connections, which facilitate forward movement of information. Each of these connections has a weight, and these weights can change during training. To understand what's going on, we take an example of classical conditioning. Jim offers Dwight an altoid. Since Dwight loves altoids, he always accepts the offer. We can model this sequence of events using the graph above (figure 9).

Jim also sometimes restarts his computer, triggering the Widows startup sound. Dwight is completely uninterested in the Widows sound, so he exhibits no response. We can use the following graph to model this (figure 10).

Note that the main difference between Figures 9 and 10 is the weight of the connection to the motor neuron. Now,

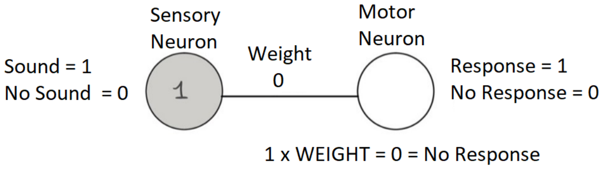


Figure 10: Dwight's initial reaction to the startup sound

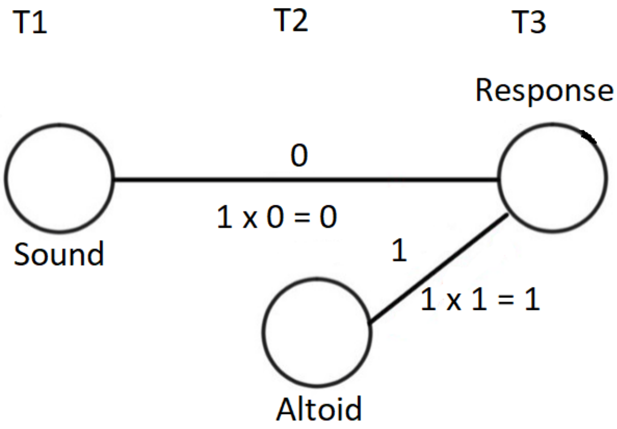


Figure 11: Initial state for classical conditioning experiment

Jim starts the training phase of classical conditioning. Every time Jim turns on his computer and the Windows startup sound plays, he offers Dwight an altoid. We can model this interaction by combining the two graphs (figure 11).

So now Jim enters the training phase. As we can see in Figure 12, each trial slightly increases the weight of the sound. After a while, the sound weight will be high enough that the sound will elicit the "Wants Altoid" response, even when the "Offers Altoid" input is removed.

Classification Problem. The problem that we want to solve for Watch&Learn is a classification problem. The idea is to classify each of our inputs into some discrete amount of labels. Specifically, given the caption file for an unknown anime, we want to give it a label 1-10 based on how difficult it is to understand. As an example, let us imagine a garage door remote with four buttons. Three buttons don't do anything, and one opens the garage door, but we don't know which one is which. We can model this as the following neural network (figure 13).

We press button1, so the inputs are [1,0,0,0] with the default weights [0,0,0,0] and the output node returns the sum of [inputs] * [weights] which is 0. The garage door does not open, so the correct answer was 0, therefore the model was correct. The model is correct for button 2 as well (input: [0,1,0,0]). When we press button 3, the output node returns

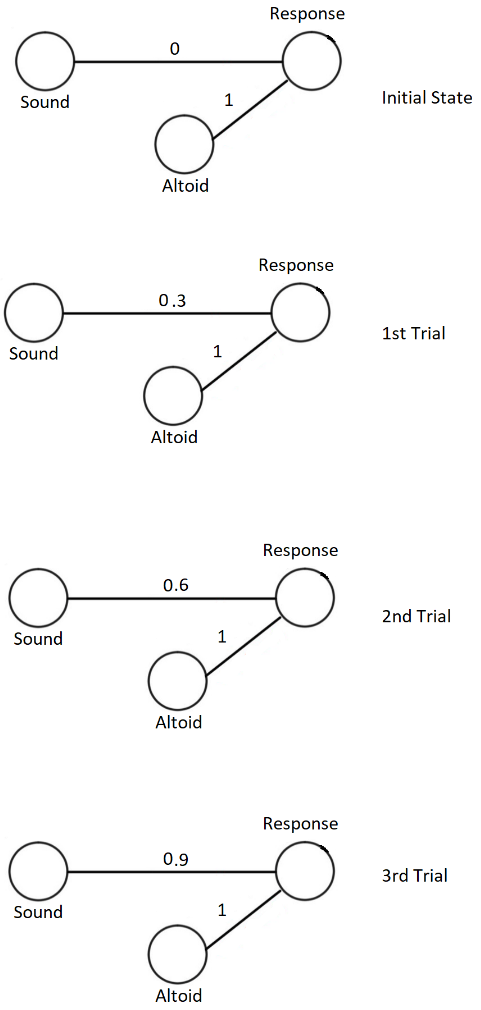


Figure 12: Training for classical conditioning experiment

0 just like before. However, this time the garage door does open, so the correct answer was 1. We use the error function [Error = Desired Output - Neural Net Output] To find an error of 1, so we were 100% wrong about the output to button 3. We then update weight 3 so that, next time, the network will be slightly less wrong. After enough trials, the neural network will correctly guess that button 3 will open the garage door.

Convolution. We take a brief aside to discuss the concept of convolution. Convolution is an integration function which measures how the shape of one function is modified by the

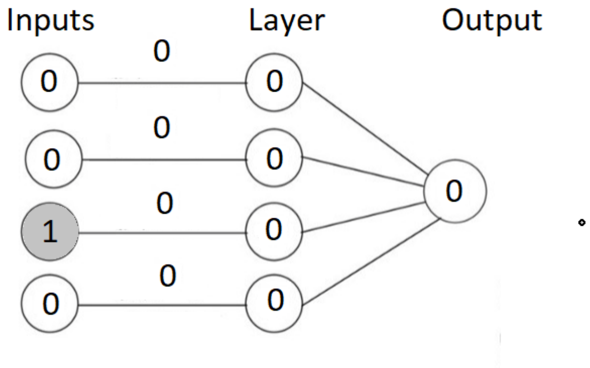
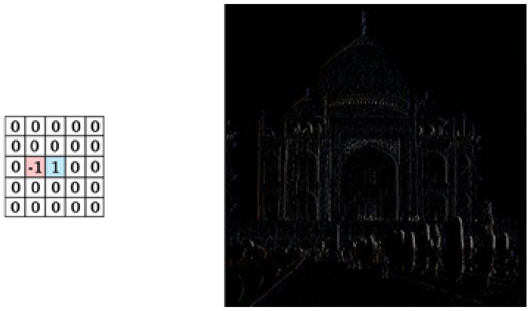


Figure 13: Graph representation of a garage door remote



0	0	0	0	0
0	0	0	0	0
0	-1	1	0	0
0	0	0	0	0
0	0	0	0	0

Figure 14: Convolution-based Edge Detection of the Taj Mahal

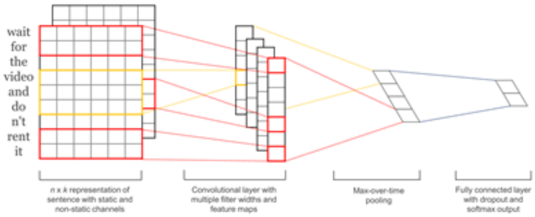

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 15: Text Classification CNN

other [4]. They are used heavily in the field of image processing, for example in edge detection. One could use a convolution function to subtract each adjacent pixel. Side by side pixels would have similar values, so the difference would be nearly zero. Meanwhile, pixels on an edge may have radically different values, and thus would show up as white.

Convolutional Neural Networks. It's fine to talk about a neural network with only five possible options, but these aren't particularly useful. The neural network used by Watch&Learn would include thousands of inputs, and it wouldn't make sense to train each of these individually. Instead, we use a convolutional neural network. A convolutional neural network (CNN) is a neural network that uses many identical copies of the same neuron. This increases the scalability and efficiency of the neural network because there are less parameters to memorize [5]. More specifically, instead of each input node being connected to an output node on the next layer, the input node is convolved with the each subsequent layer and is only actually connected to the overall output node [6]. Let's say there is a caption file for an episode of anime. We can take each word of the caption file as our inputs. The naiive solution would be to directly connect these inputs into fully-connected layer, which is connected to the output, similar to the garage door example. A better solution would be to attempt to find some symmetry between words or groups of words, by looking at some local property of the file [5], for example word usage. This layer of neurons is the convolutional layer, the output of which is then fed into the full-connected layer.The network can have a group of neurons whose job it is to compare the word usage in the file. Since these are local properties, the layer will look at groups of words of some window size. For example, with a window size of two, the nodes would look at every group of two words. We can also stack another convolutional layer on top of the first one, to detect higher level features based on the results of previous layers. For example, going back to our convolution example (Figure 14), one layer could learn which groups of pixels are edges, the next layer could learn which groups of edges make up shapes, and the third could learn which groups of shapes make up faces [6].

Text Classification CNNs. And this finally brings us to the CNN used in Watch&Learn. Step 1 is to assemble a data set of thousands of labeled caption files [7]. To make this easier, we assume that episodes from the same series have the same difficulty. So if we label one hundred animes, we have a bit more than 1000 training samples. For each caption file in the training data, the single words are used as inputs. These inputs are put into a convolutional layer, which looks at different window sizes, filter widths, and feature maps to move from words to sentences. The next convolutional layer moves from sentences to lines, and then a third from lines to acts. From here, the data moves into the fully-connected layer which uses a soft-max normalization function to generate an output.

Using SMOG Readability Index for Episode-Level Difficulty Classification

Due to time constraints, we used the SMOG readability index as a stop-gap measure. The SMOG index estimates the years of education a person needs to understand a piece of writing, based on syllable counts [8]. We used regex to remove all the timestamps, to avoid confusing the algorithm. It's a decent estimate for difficulty, but it's less accurate than the CNN because it uses less parameters and is much more generic.

Using English Profile for Word-Level Difficulty Classification

An important part of learning languages is learning vocabulary. Since the app focuses on visual media, the vocabulary should be presented in that context. This way unnecessary focus doesn't fall onto presenting vocabulary, it appears only as supplementary information to further a user's understanding of the content already being presented.

English Profile. English profile is a branch of linguistics focused on classifying words by acquisition difficulty. The scale is from A1-C2, using the Common European Framework of Reference (CEFR)[9]. EnglishProfile provides a free dataset on their website, with English words paired with one of six corresponding levels of difficulty.

Web Scraper Spider. The English Profile dataset is not downloadable, so to obtain the data we implemented a web scraper. Web scrapers autonomously go through the web, find desired content, and parse that content into a usable form. Our spider, made in Python Scrapy, went through each page of the dataset, lifted the pertinent information and transferred it into a csv file. From there, the information could be imported into the Django database using django-import-export functionality.

Context-based Key Terms Generation. Key terms are chosen based on contextual criteria. The vocabulary score of the user, the acquisition difficulty of the word, and the time the word was used in the anime. The acquisition difficulty is gotten from English Profile. The time the word was used is obtained by scrubbing the subtitle file for timestamps. The engine looks through each level from level 6 (the highest) to a minimum level dictated by the vocabulary score. For each of the words on the list, it looks through the subtitle file to see if it contains the word. If it does, it takes the timestamp and adds it to the list of words to display. When we had the key terms list populated, we used the YouTube API to display each word in context.

Limitations. The chief limitation to the English Profile method is that English is the only language with an equivalent dataset. For this reason, we decided to stick with teaching English for this implementation, but further languages would require a different dataset. One possibility would be finding a frequency distribution of every word used in every book written in a particular language on Project Gutenberg, an open-source digital library. Project Gutenberg has 57,000 ebooks, with at least 50 books in 16 languages [10]. Words that appear less often could be given a higher difficulty, since it is less likely that a user would have come across that word before. Another limitation is the existence of homonyms. Given a word used in context, there is currently no way of telling how the word should be used. It would be possible to use another machine learning algorithm to figure this out, but we just used the first definition given in case of a homonym.

Measuring Understanding and Feedback

After watching the episodes, users should be properly tested to check that the app is working for them. Users want to be able to check their progress in language learning, and have clear feedback that they are moving forward with their education.

Quizzes. We use comprehension quizzes to test that the user understood what was going on in the episode that they just watched. Preferably, we would also like to test other parts of language learning but we ran into the same block as we did with the assessment. Linguistics and education aren't in our skill set, so we were unable to create this type of content.

Feedback. On top of quiz scores, we wanted to leave the users with a visual feedback of their progress within the app. To this end, we implemented a level system so that users could gain levels and unlock rewards as they progress through the app. Users gain experience for getting after-episode questions correct. When enough experience is obtained, the user will increase in level and be rewarded by a new player icon.

6 CONCLUSIONS

Watch & Learn provides an exciting opportunity to use anime, a globally popular form of media, as a language learning tool while preserving the entertainment aspect that draws people into media consumption. We believe the machine learning aspect will give it a unique edge when competing with existing video streaming services by providing a tailored experience and a content-driven curriculum for each user. While our implementation of the application was limited to English language, we would like to have more languages available so that anyone who enjoys watching anime can experience the added benefit of learning a language as they tune in.

7 LINKS

- GitHub: https://github.com/cs160su18/final-lishirley71
- Slides: https://drive.google.com/open?id=1R1zhbl47vxzR_n-NE4CKZ-JVVy1OkUKcqCYp3_VpA90

- Medium Article: https://medium.com/@mbohol/watch-learn-2768467b8143?source=linkShare-26580c4b648-1533960548
- Final Video: https://www.youtube.com/watch?v=Io9hJ-KWaYU
- Poster: https://drive.google.com/open?id=1hjyVru--bh2YQK-uRCc8QNXHqqEuLM0h

8 REFERENCES

(1) arXiv:1408.5882 [cs.CL]

(2) dennybritz, cnn-text-classifcation-tf, (2018), GitHub repository

(3) Leon, Keno. "Making a Simple Neural Network âĂŞ Becoming Human: Artificial Intelligence Magazine." Becoming Human: Artificial Intelligence Magazine, Becoming Human: Artificial Intelligence Magazine, 9 Apr. 2017, becominghuman.ai/making-a-simple-neural-network-2ea1de81ec20.

(4) Colah. "Understanding Convolutions." Understanding LSTM Networks – Colah's Blog, 13 July 2014, colah.github.io/posts/2014-07-Understanding-Convolutions/.

(5) Colah. "Conv Nets: A Modular Perspective." Understanding LSTM Networks – Colah's Blog, 8 July 2014, colah.github.io/posts/2014-07-Conv-Nets-Modular/.

(6) Britz, Denny. "Understanding Convolutional Neural Networks for NLP." WildML, 10 Jan. 2016, www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/.

(7) Various. "Subscene." Subscene, 2018, subscene.com/.

(8) THE SMOG READABILITY FORMULA, a Simple Measure of Gobbledygook, www.readabilityformulas.com/smog-readability-formula.php.

(9) "English Profile." English Profile, 2018, "http://vocabulary.englishprofile.org/staticfiles/about.html"

(10) "Project Gutenberg." Project Gutenberg, 2018, "https://www.gutenberg.org/"