Text Classification for AI Education

Tejal Reddy
reddyt@mit.edu
MIT Media Lab
Cambridge, MA, USA

Randi Williams
randiw12@mit.edu
MIT Media Lab
Cambridge, MA, USA

Cynthia Breazeal
cynthiab@media.mit.edu
MIT Media Lab
Cambridge, MA, USA

ABSTRACT

To help middle school students explore Artificial Intelligence (AI), we built a text classifier extension into a block-based programming interface using Tensorflow's K-Nearest Neighbors and Universal Sentence Encoder libraries. After training a model, students can incorporate it into their own creations. In this paper, we discuss how we taught students the AI concepts behind the classifier and how students used the text classifier to build their own projects. Lastly, we touch on how our classifier works just as well as other text classification platforms. This text classification tool and curriculum is a powerful way to help students become more knowledgeable about the ever-growing field of AI and to raise their awareness about applications of AI within their own lives.

KEYWORDS

AI education, text classification, text tagging, machine learning

ACM Reference Format:
Tejal Reddy, Randi Williams, and Cynthia Breazeal. 2021. Text Classification for AI Education. In *SIGCSE '21: ACM Special Interest Group on Computer Science Education, March 2021, Toronto, ON*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM AND MOTIVATION

In recent years, Artificial Intelligence (AI) has become increasingly prevalent in our lives. Because of this, it is important for individuals of all ages to be aware of how AI impacts them. To help middle school students learn more about AI, we created the *How to Train Your Robot* curriculum to teach students about AI, how it's used, and ethical issues with the technology. One key topic of the curriculum is Text Classification. To teach students this topic, we provided them with a hands-on opportunity to experiment with text classification and apply it to their own projects. To enable this exploration, we created our own model-making application embedded within a block-based programming platform.

2 BACKGROUND AND RELATED WORK

As AI has become more prevalent, there has a rapid increase in work geared towards teaching students about AI [11]. Many AI platforms such as Cognimates [2], Machine Learning for Kids [8], Snap! [6], and AppInventor [13] use block-based programming languages catered towards students unfamiliar with programming. However, few tools allow students to create and use their own natural language processing algorithms.

Our extension is most similar to the Machine Learning for Kids [8] text classifier. However, their model is not directly built into a programming platform and requires students to generate their own API keys, which have limited free use. We provided students with a more streamlined platform that allows them to create models without limitations. Furthermore we created activities, similar to those used in other middle school AI curricula [9], to help students understand more about AI and its impacts.

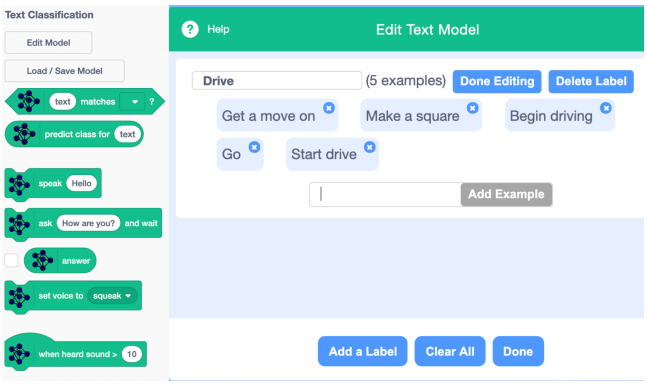

Figure 1: Text Classifier extension interface. Users input training data using the interface on the right and can then program with their model using the blocks on the left.

2.1 Curriculum Design

We wanted to ensure that students thoroughly understood all the steps of text classification. To do this, we emphasized the concepts of **(1) word embeddings, (2) the K-Nearest Neighbors (KNN) algorithm, and (3) classification bias**. Students then demonstrated their understanding in a **(4) programming activity** and their final projects.

1. **Word Embeddings:** Students were introduced to the concept of how words can be numerically represented with word vectors. We went through examples of creating a word vector with the word 'princess' and deciding whether the numbers in its vector corresponding to 'royalty', 'masculinity', 'femininity', and 'age' should be high or low.

2. **KNN Algorithm:** To better understand the KNN algorithm, students used a visual [4] of words plotted on a 2-D graph. They learned how the selection of the K parameter can impact the output of the algorithm.

3. **Classification Bias:** To illustrate classification bias, students used a word analogies website to plot jobs such as 'nurse', 'doctor',

SIGCSE '21, March 2021, Toronto, ON

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

'scientist', 'dancer', 'teacher', 'actor', and 'artist'. From there, they were able to observe gender biases in how word vectors represent some of these words.

4. **Scratch Activity:** Students put their new knowledge into practice by using a Text Classification model-making extension we built into a block-based programming interface. They started with a short tutorial that showed hem how to make a robot respond to three commands: Drive, Dance, and Speak (Figure 1).

3 APPROACH AND UNIQUENESS

The text classifier was designed to maximize ease-of-use and understanding for middle school students. The three main components of the text classifier are the **(1) translator, (2) sentence encoder, and (3) classifier.** The classifier is built into a block-based programming interface developed on top of the open-source Scratch Blocks repository [7] and can be accessed at *https://mitmedialab.github.io/prg-extension-boilerplate/intent-classifier/.*

1. **Translator:** Translates user input to the classifier into English. The language of each input is automatically detected and then translated to English before being added into the classifier model. We implemented this translator to ensure students anywhere in the world can use this tool.

2. **Sentence Encoder:** Converts user input to the classifier into a word vector using Google's Universal Sentence Encoder (USE) [3]. The version of USE used for the text classifier is trained on a deep averaging network (DAN) encoder, and the output is always a 512-dimensional vector.

3. **Classifier:** Creates a text classification model utilizing Tensorflow's K-Nearest Neighbor's library [10]. The model determines the appropriate label for new input by comparing it with the phrases from the training dataset.

As users input more training data, the K parameter is dynamically calculated as the square root of the number of samples inputted. We chose this over a fixed K to allow the classifier to adjust if a larger training dataset was utilized. To combat having too small of a K, there should be at least five training examples for each label.

4 RESULTS AND CONTRIBUTION

4.1 Student understanding

We tested the classifier with students in the Summer of 2020 during an online class with 29 students. In their daily reflections, students demonstrated their excitement about the concept of text classification and the multitude of ways it could be used. "The coolest thing was the text classification" (Brant, age 13). "Today I got to create my own command for my robot! That was amazing"(Cacey, age 12).

In their final projects, students used their knowledge about text classification to implement their own projects. There were a total of eight final projects that used text classification. Projects tended to align with the theme of helping others and included a snake identifier, a TV show suggester, a dog food detector, an addition robot, a chat robot, a concussion tester, an animal classifier, and a healthcare robot (Figure 2).

4.2 Comparison with other text classifiers

To determine the effectiveness of our text classifier, we compared it against two similar text classifiers for children: the Machine Learning for Kids text classifier and UClassify's text classifier [12]. We generated two test datasets, one which contained phrases that could be classified as click-bait and not click-bait [1] and the other which did sentiment analysis on movie reviews [5].

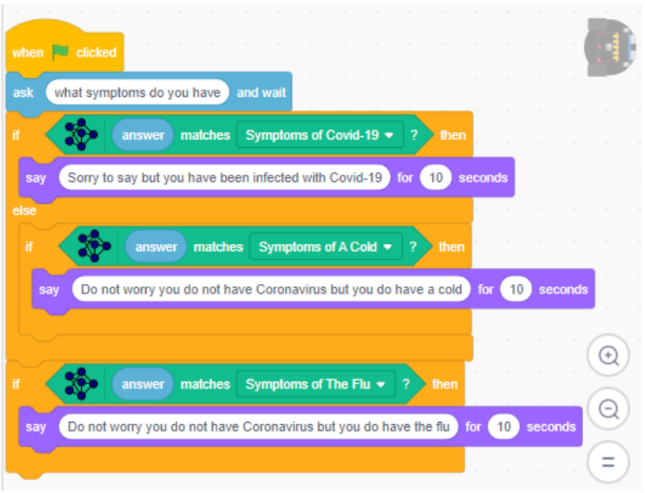

Figure 2: Code from Healthcare Robot final project

We conducted three rounds of testing with each dataset, training on ten randomly selected inputs. We only used five random inputs for each label to imitate how children used these tools. We then tested the effectiveness of the classifiers by testing them on four random phrases (two of each label).

On the clickbait dataset, our classifier classified the test data correctly eleven times out of the total 12 tests (91.7%) across all trials. ML4Kids classified 10/12 (83.3%) correctly. UClassify classified 5/12 (41.7%) correctly with 5/12 (41.7%) not yielding a definite classification. On the movie reviews dataset, our classifier classified 11/12 (91.7%) correctly. ML4Kids classified 7/12 (58.3%) correctly. UClassify classified 5/12 (41.7%) correctly with 1/12 (8.3%) not yielding a definite classification.

5 CONCLUSIONS AND FUTURE WORK

In this work, we described the process of creating and implementing a text classifier for the *How to Train Your Robot* curriculum. Through the use of final projects, we saw that the text classifier and related activities were effective in helping students understand how it worked as well as its uses. Many of the students used the classifier in their final projects to help others, and by doing this, were able to reinforce the concepts taught in class. In the future, we hope to improve this understanding by adding a KNN plot so that students can visualize the reasons behind their classifiers' decisions.

ACKNOWLEDGMENTS

We would like to thank the teachers and students who participated as well as Amazon Future Engineer for supporting the program.

REFERENCES

[1] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop Clickbait: Detecting and preventing clickbaits in online news media.

In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 9–16.
[2] Stefania Druga. 2018. Cognimates. http://cognimates.me/home/
[3] Google. [n.d.]. https://tfhub.dev/google/universal-sentence-encoder/4
[4] Italo José. [n.d.]. https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d
[5] Kaggle. [n.d.]. https://www.kaggle.com/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews/data
[6] Ken Kahn, Rani Megasari, Erna Piantari, and Enjun Junaeti. 2018. AI Programming by Children using Snap! Block Programming in a Developing Country. In *EC-TEL Practitioner Proceedings 2018: 13th European Conference On Technology Enhanced Learning, Leeds, UK, September 3-6, 2018.* http://ceur-ws.org/Vol-2193/paper1.pdf
[7] Lifelong Kindergarten. [n.d.]. https://github.com/LLK/scratch-blocks
[8] Machine Learning for Kids. [n.d.]. https://machinelearningforkids.co.uk/
[9] Personal Robots Group and I2 Learning. [n.d.]. https://aieducation.mit.edu/documents/i2educatorguide2019.pdf
[10] TensorFlow. [n.d.]. https://github.com/tensorflow/tfjs-models/tree/master/knn-classifier
[11] David Touretzky, Christina Gardner-McCune, Fred Martin, and Deborah Seehorn. 2019. Envisioning AI for K-12: What should every child know about AI?. In *AAAI*.
[12] Uclassify. [n.d.]. https://uclassify.com/
[13] Jessica Raquelle Van Brummelen. 2019. Tools to create and democratize conversational artificial intelligence.