Machine Learning Engineer Nanodegree

Capstone Proposal

Daniel Brandt
June 15, 2019

I. Domain Background

Automation of grading multiple choice exams is trivial. For many reasons, short answer tests provide a better tool for helping students assess their knowledge. So far, this area remains a challenge and machine learning has not yet provided usable solutions. This doesn't mean that machine learning can't add a lot of value today. A recent paper recommended that it be used to supplement the quality, provide automation in some areas of grading and help target areas where more human involvement is needed. *1

Because short answer tests are very tedious to grade, the success in automating related areas such as identifying plagiarism and essay grading has led to renewed efforts to attach this area. The problem is challenging because of the need to focus on identifying correctness in relatively short answers. Longer answers and essays require a broader set of criteria like grammar, ideas, structure and therefore provide more features for machine learning to work with.

If we had enough short answer test results graded by a human for a particular test, it would be relatively easy to model and grade a short answer test. Unfortunately, exhaustive student answer datasets rarely exist in the real world for a specific short answer test. The reality is that tests generally have to be refreshed regularly to reflect constantly changing content and, in some cases, to prevent unwanted distribution and cheating.

1. Some of the Challenges of Grading Short Answer include:

    - Teachers usually find the task of assessing respondents' answers very time-consuming.
    - Students may have to wait for a long time to receive feedback on their responses
    - When they finally get it, the grade can be different from another classmate's, who has given a very similar answer. *2

2. The challenges of grading Short-Answer compared to essays are:

- Response length. Responses in SAS tasks are typically shorter. For example, while the ASAP-AES data contains essays that average between about 100 and 600 to-kens (Shermis, 2014), short answer scoring datasets may have average answer lengths of just several words (Basu et al., 2013) to al-most 60 words (Shermis, 2015).
- Rubrics focus on content only in SAS vs. broader writing quality in AES.
- Purpose and genre. AES tasks cover persuasive, narrative, and source-dependent reading comprehension and English Language Arts (ELA), while SAS tasks tend to be from science, math, and ELA reading comprehension. *3

3. Defining correct:

From my analysis, I believe one of the most difficult challenges is defining what correct actually means. There is no absolute measure of correctness for this kind of test, rather, we can only look at how human graders make the judgement. Human graders tend to look for key words and pattern matching rather than the specific order of words when grading large numbers of questions.

- The number of correctly used words has more influence on marks than semantics or order of words.
- If a large number of responses are being graded, it is not unreasonable that a human would move towards pattern recognition via key words rather than “reading for meaning”.
- Identifying words gives an idea about grades and students misunderstanding to teachers. Such an approach allows time saving for scoring, and to provide rapid feedback to students by checking the words used from model vocabulary. *1

4. My Interest in this area stems from two areas:

- I’m very interested in Education technology and deeply involved in the develop a Competency Based Undergraduate degree platform. This is similar in some ways to Udacity approach while targeting an entire undergraduate or graduate degree. It was critical for this to have large numbers of automatically graded assessments to help the student learn basic courses material in a self-paced way. The final competency determination in a course was based on a project or multiple-choice exam.
- I’ve been very interested in memory and how recall vs. recognition differs. Multiple choice exams rely mostly on recall and/or some higher order problem solving based on the question. The more complex the questions, the more time consuming it is to write a good multiple-choice test suite. On the other hand, short answer is much easier to write and require the student to recall, which is more challenging, and then concisely articulate the answer. These are much

easier to generate than multiple choice but far harder to grade. Moreover, they currently cannot be used in a self-guided course without additional grader involvement. The ultimate solution will require some amount of human grading but ideally a large portion of the effort could be automated. Achieving this would have a considerable impact on the ability to deliver scalable self-paced courses.

*1 Suzen, Neslihan & Gorban, Alexander & Levesley, Jeremy & Mirkes, Evgeny. (2019). Automatic Short Answer Grading and Feedback Using Text Mining Methods. Page 1, 19.

*2 Galhardi, Lucas & Brancher, Jacques. (2018). Machine Learning Approach for Automatic Short Answer Grading: A Systematic Review: 16th Ibero-American Conference on AI, Trujillo, Peru, November 13-16, 2018, Proceedings. 10.1007/978-3-030-03928-8_31. Page 380

*3 Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. Investigating neural architectures for short answer scoring. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications. pages 159–168. Page 159

II. Problem Statement

Create a machine learning model and the supporting code to perform grading of an arbitrary short answer test given one or multiple possible correct answers and a limited set of actual test results.

For this project I will focus on reproducing some results from the literature and implementing an approach with Sagemaker and custom Sklearn models. The final model will be hosted with a basic web front end that will allow a user to answer test questions and return the prediction for correctness of their answer and the degree of confidence. This allow human testers to experiment interactively with real world variations to short answers see the quality of the model results.

The problem will be broken into two pieces.

First, I will try to replicate the results from the Riordan (*3) paper referenced above using Sagemaker with hypertuning and a basic LSTM model. The paper described several additional layers and techniques in addition to basic LSTM. They identify improvements to a basic model such as pretrained turned embedding and an attention layer may. These may or may not be attempted for this project based on time and complexity.

Second, the model created for the base line will be trained and tested on a larger data set a possibly some manually generated questions to see how it performs on a larger or arbitrary question set.

III. Datasets and Inputs

I plan to use two datasets.

1. For comparison to a baseline int Rioden (*3) paper, I will use one of the SciEntsBank (SEB) dataset used in their analysis. This data was taken from Dzikovska et al., 2012. The SciEntsBank (SEB) dataset consists of science assessment questions and I will work the set with 2-way labels (correct/incorrect). *1

2. For more comprehensive modeling and testing, I will use a University of North Texas short answer grading data set. It consists of ten assignments between four and seven questions each and two exams with ten questions each. These assignments/exams were assigned to an introductory computer science class at the University of North Texas. The student answers were collected via an online learning environment. The data set as a whole contains 80 questions and 2273 student answers. The answers were scored by two human judges, using marks between 0 (completely incorrect) and 5 (perfect answer). Data set creators treated the average grade of the two evaluators as the gold standard to examine the automatic scoring task.*2

*1 Myroslava O Dzikovska, Rodney D Nielsen, and Chris Brew. 2012. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

*2 Suzen, Neslihan & Gorban, Alexander & Levesley, Jeremy & Mirkes, Evgeny. (2018). Automatic Short Answer Grading and Feedback Using Text Mining Methods.

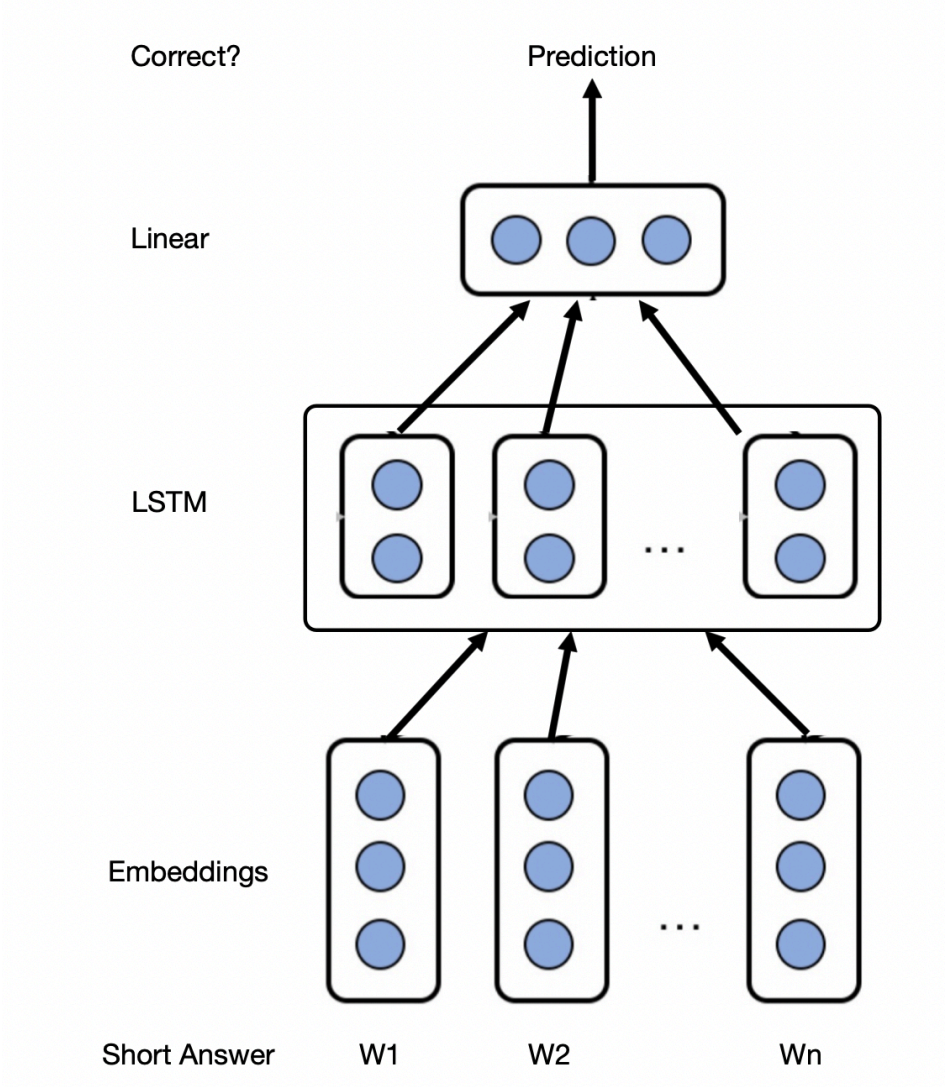
IV. Solution Statement

I propose to do a basic model variation which leverages from the Riordan paper which leverages LSTM to see the results for the SEB data. LSTM make sense here because the order and intent of the words is important and with language that is often well handled by adding memory to the model.
The model will be tested using Sagemaker with hypertuning. Once a baseline is established with a simple LSTM model and the results compared to those in the paper (about 60-75% accuracy), I will apply the same techniques to the Texas Data which is more in line with the type of short answer testing I am interested in addressing.

Finally, I will experiment with some of the additional modeling suggestions to see if I can improve on my results.

V. Benchmark Model

I will use a simplified version of the model from the Riorden paper removing the Convolution layer which was not deemed that helpful. In addition, I have also removed the aggregation/attention layer they used to manage the scope of this project. The attention layer and the use of pretrained embeddings with tuning were shown to improve results in general. I will explore these and additional enhancements after the basic modeling is stabilized and gives results in the expected range of 60 to 75% accuracy.



VI. Evaluation Metrics

The evaluation will be based on the three common metrics for a Boolean predictor: accuracy, precision and recall. The best results from the baseline was around 75% accuracy. I will also look at balancing precision and recall. From a grader's perspective, it's probably desirable to increase precision so that students are not receiving correct answers for incorrect work. They will eventually see the correct answer and can request a review. It's much more unpleasant to be downgraded then upgraded. Since we also expect this process to flag questions that are difficult to grade, we would want to look at questions that lots of students are getting wrong. They may be wrong because the grading is incorrect, or the question is poorly worded. Focusing on identifying these would indicate a bias toward precision over recall.

VII. Project Design

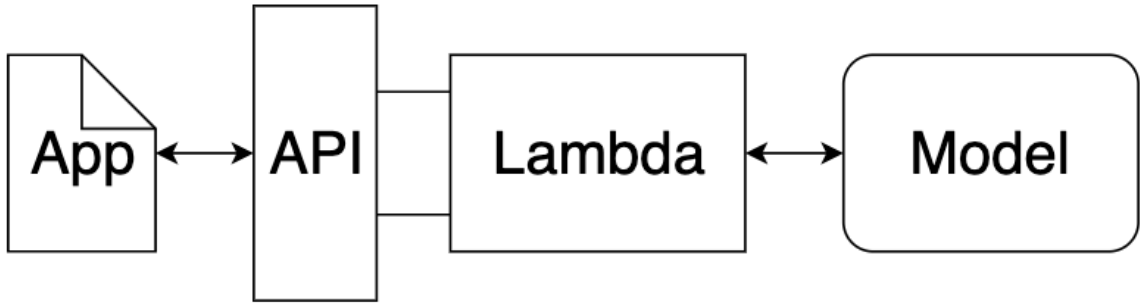
The project will use Sagemaker with a custom Keras deep learning model. I will start testing with the baseline model proposed in the Riorden paper based on Embedding, LSTM and linear regression. The testing will involve hypertuning that model and then looking at some variation to the model. Most importantly, I will experiment with the input layer and look at what if any additional features I can deduce from the problem domain.

The first step prior to modeling will require data cleansing to remove data quality issues from the test datasets. The next step is to train a custom Keras model in Sagemaker. I did some quick prototyping to validate the syntax of a baseline model which is below and serves as a prototype.

Model Code:

```
model = Sequential()
model.add(Embedding(len(from_num_dict), 30, input_length=max_length))
model.add(Dropout(0.2))
model.add(LSTM(100, return_sequences=False, input_shape=(max_length,)))
model.add(Dropout(0.2))
model.add(Dense(1, activation="linear"))
# compile the model
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['acc'])
# summarize the model
print(model.summary())
# fit the model
model.fit(test_answers, test_labels, epochs=200, verbose=0)
# evaluate the model
print(f"test_answers shape: {test_answers.shape}")
loss, accuracy = model.evaluate(test_answers, test_labels, verbose=0)
print('Accuracy: %f' % (accuracy*100))
print('Loss: %f' % loss)
```

The basic structure of the application will be the standard AWS flow:

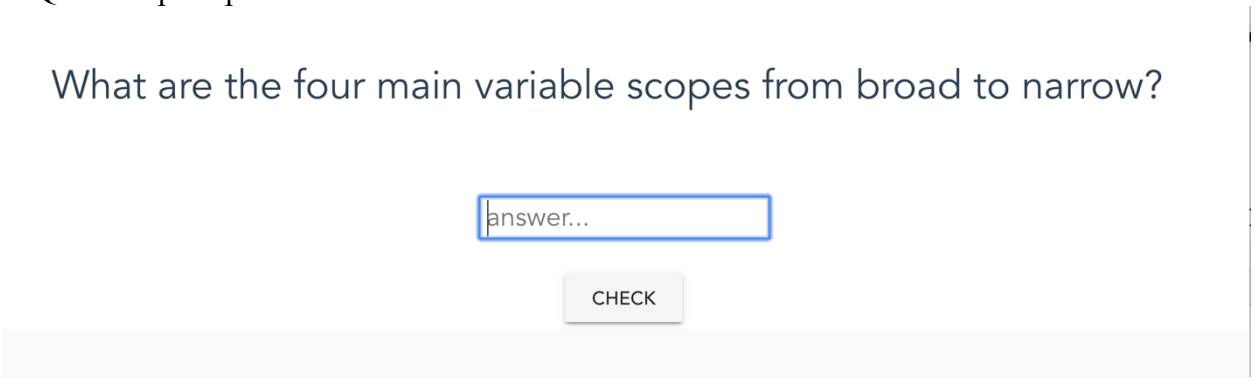


The app will be a single page JavaScript app that renders the questions and accepts the short answer input. The user submits the answer which it sent to the API.

The response from the API will include whether the answer is correct and the probability calculated.

Prototype Screens:

1. Question prompt



2. Answer Submission

The user can agree, disagree (Nope) or hit skip if they gave no answer as in the case below. Ultimately that interface will allow you to record human evaluation of the model decision.

What are the four main variable scopes from broad to narrow?

answer...

CHECK

Global, Class, Instance, Local - Incorrect. (0)

CORRECT NOPE SKIP