Gregory Knapp

CS 401R

Evaluation and Lab Machine Execution Write Up

Link to code (Google Colab): <https://colab.research.google.com/drive/1RVZ7zsD6JX1iCCskiFlrioAfHY8i8ZQ4?usp=sharing>

Original Model: Character RNN using an LSTM

2 Baselines:

1. GPT2 finetuned on the WoW quest titles
2. Another Character RNN using a GRU instead of an LSTM

2 Comparison Metrics:

1. Human rated quest title quality: The survey is linked to in the code notebook. 7 participants were asked to rate 15 samples of quest titles for each model. The 1 sample of the 2 baselines and the original model were randomly paired together and were ranked from best to worst. The total ratings (how many best, mid, and worsts for each model) are graphed side by side on a bar chart in the notebook. Additionally, the data is averaged for the 7 respondents and the average ratings are graphed on a separate bar chart below.
2. BLEU Score: All the quest titles from the input file were read as sentences and split and saved as references for calculating the BLEU score. Since the different models required different epochs/steps to train, I decided to take 5 snapshots during each’s training process. (ex. My LSTM only took about 25 epochs to train, so every 5 epochs, I would sample output and take the BLEU score relative to the entire set of titles from the inputs, but for my GRU model which took 400 epochs to train to a decent level, I would take a sample every 80 epochs). This is not the best method, but was the best of both worlds I could find to still compare the models despite differences in training/finetuning.

Results:

The BLEU score results were not the most helpful things ever. I think that is mainly due to how inconsistent it is to try and standardize how the output is samples. I used the same prompt text and the same text length for the outputs, but still, there is so much variation and randomness involved with sampling text that sometimes the BLEU scores were very high at the start then would randomly drop off during training. The only thing that generally matched expectations of improvement was the GPT2 finetuning, which constantly improved as it fine trained more and more.

The human ratings were much more interesting. I originally only intended for my wife to grade them, but then passed it on to my brother as well. With just two input samples, the ratings looked as expected, GPT2 far outperformed the rest with the most “best” ratings, while the LSTM was generally second best, and the GRU was by far the worst. Then my wife thought it was funny, so she sent my survey to her friends, and I ended up with 7 respondents. The general trend was the same, but the results started to have more variance. The LSTM model and the GRU model had similar ratings for mid and worst outputs, and the gap was closed between them. I tried to advise people to rate them in terms of what is the most convincing WoW quest title, not what is the best English or the funniest, but I think some ratings were not entirely in line with those expectations. If anything, that is just evidence of the challenges of using human-centric metrics in rating machine learning models. That said, the ratings still met my general expectations and unsurprisingly, GPT2 did the best job of producing quest titles that people thought belonged to World of Warcraft.

Reflection on Part 2:

We talked about it a bit in class, but it was a funny experience trying to work on the same lab computer as everyone else. I have not had the experience of working in a CS research lab, so I’ve never had the problem of trying to log onto the same machine and coordinate with others, so it was amusing in a way each time that somebody would boot someone off the GPU on accident. In our case, it is a very low-pressure environment, so there is no issue with that, but it did impress on me the importance of ensuring what you are doing is in line with lab/work expectations. In our case, interrupting someone’s training is just a small annoyance, but in a grad school/research work situation, it could be a huge waste of time and resources. That was an unexpected but welcome take away from this assignment. Aside from that, the training went smoothly, though I do agree that the README was not very clear about how to use your own custom data sets to make the classifier/train the NPI etc. If this is every used in another 401R class, it is probably extra work, but if you clone the repo and then remove the default classifier/extra files added to make all the steps run, that might help the usage learning process. Otherwise, the code just runs and its hard to realize that it is just using those default files instead of the ones you generated.

I was able to do the construct data and train the classifier over the last weekend, but once we got to Monday, it sounded like a few people still needed to work on their training, so I decided to hold off on the NPI. I will check it after class on Wednesday, if there is space, I will start to train it. If not, then I will wait until the weekend and look again.