Meta-Turing Test by Joe Howarth and Dan Pechi

The development of Generative Adversarial Networks (GANs) by Ian Goodfellow in 2014 marked one of deep learning's more recent milestones. The framework consists of two neural networks: a generator and a discriminator. The generator is tasked with producing new examples of a particular entity, whether that be faces, words, or animals. The job of the discriminator is to then learn to distinguish fake, network-generated images from ground truth images of the entities attempting to be reproduced by the generator. In a GAN, the discriminator's feedback is fed back into the generator, such that it attempts to improve the quality of its output. GANs have shown promise in a variety of domains, possibly the scariest of these domains is producing of fake news: https://www.youtube.com/watch?v=9Yq67CjDqvw.

The idea of GANs inspired our initial attempts at a language model, however after digging around, it became apparent that resources for GANs for NLP were very sparse. This motivated us to take the idea in a different direction, essentially uncoupling the discriminator and the generator to act independently. While cool, this configuration left us without any sort of actually interesting user experience.

The uncoupled discriminator and generator can be framed in the context of the Turing test as the judge and the artificial intelligence, respectively. Neural language generators have been applied to the Turing test in the past to achieve state-of-the-art results on the task (Vinyals and Le, 2015). Important to note are the ways non-neural generators have handled the Turing test in the past, often opting for purposefully inaccurate language to mimic linguistic patterns of children or those just learning the language. While good at mimicry, these algorithms lack any real intelligence. Training a conversation model and corresponding discriminator however would

have required the configuration of the discriminator to look over the entire conversation to assess humanness, so the problem was modified such that the generator would only produce sentences as opposed to conversational responses.

The problem was thus designed as a competition between this artificial discriminator and a human discriminator to better distinguish human from artificial intelligence. The framing of the Turing test in this fashion meant we could not only assess the humanness of the generator, but the ability of the discriminator relative to the human. To pass the normal Turing test, the generator would be able to trick both the human and the discriminator into thinking it was human. An equally interesting outcome would have been if the human was able to figure out the machine-generated sentences, but the discriminator couldn't. This would point to higher levels of linguistic understanding distinguishing the discriminator from the human. For instance, the discriminator might be able to pick up on syntax, but would miss out on pragmatics, something a human judge would, presumably, be able to pick up on. By far, the most interesting outcome of the experiment would have been if the discriminator were able to better distinguish human and machine-generated sentences than the human. This would suggest that the artificial neural network was able to pick up on features of human language that human neural networks cannot or did not encode. There's been lots of discussion in the deep learning community about how deep learning compares to our learning with most acknowledging that deep learning lacks human capabilities of local neural weight updates and generalization (Hassabis et al., 2017). As such, it's to be expected that artificial neural networks will 'understand' human language differently than humans. Whether this understanding is more efficient or even better than our own will be tested through this experiment.

We planned to build a recurrent language generator with long short term memory for the generator and transition to a convolutional model if time permitted. Using a base model of a single layer LSTM network, initial tests showed low perplexity, but the resulting language was pretty bad. Link to this base model's GitHub is provided below. The model was tweaked to be made deeper and have dropout to better generalize to the data. This improved the model's output to some extent while keeping perplexity at about the same levels. Attempts were made to add additional components like attention, residual connections, and bidirectionality, but this proved difficult in practice. As progress was being made, a paper out of DeepMind showed that LSTM networks actually proved to be the most effective at language generation tasks, so attempts to create a convolutional model were subsequently scrapped (Melis et al., 2017). The models were trained on the provided Penn Treebank Data, but seeking diversity in language input, we also used Google's Billion Words dataset in conjunction with preprocessing scripts (replacing numbers and rare words) to test out the models. Google Billion Words made for worse models, likely due to the relatively less structured nature of the text, making it harder for the neural network to generalize. The latest model has two layers, each consisting of an LSTM and a linear layer of 1024 neurons. The network was trained for 5 epochs with a batch size of 20 sentences, dropout rate of 0.1, and a learning rate of .002. Words were embedded in 128 dimensions. Training was done on 2 Tesla K80 GPU's.

The discriminator was based off of a convolutional text classification model. Kernel widths of 3, 4 and 5 were used. The model was trained on output produced by the generator and ground truth data from both Google Billion Words and Penn Treebank. The latest model has one layer with a hidden size of 1024. The network was trained for 5 epochs with a batch size of 20

sentences over an output of about 50,000 sentences from the generator. Pre-trained word2vec vectors were used with an embedding size of 100 dimensions. The network had a dropout rate of 0.5 and a learning rate of 0.2. This network was trained on a single Tesla GTX 1060 GPU.

The generator ended up producing pretty bad sentences, although we found some to be particularly human-like. An appendix of a few 'good' sentences is provided below. Perplexity, while useful for assessing the confidence of the model in its output, by no means represents any real quality of the model as it relates to intelligence, let alone human similarity. The fact that the sentences produced by the generator are near state of the art indicates there's a lot more to be done to enhance these models' abilities to reproduce human language.

GitHub Link: https://github.com/JoeHowarth/meta_turing

Website Link: https://joehowarth.github.io/meta_turing/

Presentation Link:

https://docs.google.com/presentation/d/1AaH52AAMrXyaKGn65uQ12n7TF_3mwK8-tocBtwnTHyg/edit?usp=sharing

Appendix 1: 'Good' Sentences

1 Layer Output, No Dropout:
everyone knows they still have a trusting and skilled workforce .
he has also made good advantage of drugs and credit .
tipsarevic services have provided a second price stance option for the banks that was typically possible to end the sale of the nfl .
we 're talking about years ago before anyone heard of asbestos having any questionable properties
but the prosecutor 's office official says he does not see state council , but the cross-border moves upon thinking about changing production .
anheuser started N N of the nation 's N billion shares at N

in addition georgia-pacific had purchased about N million shares of oil electronics construction losses and is convertible into one common share held at the earliest campeau department for a big <unk> contractor
some analysts generally cite moscow <unk> in ways to bolster credit
it was a socialist construction problem and it means <unk> <unk>
westridge capital inc. fell N to N and sony lost N to N data while united air lines slipped N N in august

2 Layer Output with Dropout:
<unk> pharmaceutical advanced N to N N on the american petroleum exchange which includes a full start of N shares
in national over-the-counter trading yesterday <unk> closed at $ N up N cent
we 'll admit a loss for the split says robert <unk> a trader for donaldson lufkin & jenrette
the company also expects a $ N billion loss for the general in N from $ N million
the nikkei index was down N at N after plunging $ N in composite trading on the u.s. over-the-counter market
jaguar shares closed lackluster in core trading

LSTM Language Model for Generator:

https://github.com/yunjey/pytorch-tutorial/tree/master/tutorials/02-intermediate/language_model

Convolutional Text Classifier for Discriminator:

https://github.com/pravarmahajan/cnn-encoder-nmt/blob/master/cnn_encoder_nmt.ipynb

Hassabis, D., et al. Neuroscience-Inspired Artificial Intelligence,

http://www.cell.com/neuron/pdf/S0896-6273(17)30509-3.pdf

Melis, G., et al. On the State of the Art of Evaluation in Neural Language Models,

https://arxiv.org/pdf/1707.05589.pdf

Vinyals, O., and Le, Q. A Neural Conversation Model, https://arxiv.org/pdf/1506.05869v2.pdf,

2015