What problem did you select and why did you choose it?
We want to work on the problem of automatic text generation. We want to build two different models by training on sarcastic / non-sarcastic text to build different text outputs from the same input text.

What database/dataset will you use? Is it large enough to train a deep network?
We will be using a news headlines dataset from Kaggle given at this link. It is a labeled dataset taken from two sites: Huffington Post and The Onion and classifies the headlines based on whether it is a sarcastic news story.

What deep network will you use?
We will be trying different generative models such as Recurrent Neural Networks and Transformers for text generation.

Model comparison(delete after reading)
The LMs mentioned above have their advantages and disadvantages.
In a very short and simple comparison:
Transformers are novel models but they require much more data to be trained with.
RNNs can not create coherent long sequences
Encoder-Decoder models enhanced with Attention Mechanism could perform better than RNNs but worse than Transformers
GANs can not be easily trained or converge.
https://medium.com/deep-learning-with-keras/fundamentals-of-text-generation-745d66238a1f

Will it be a standard form of the network, or will you have to customize it?

What framework will you use to implement the network? Why?
We will be using PyTorch as it gives us more freedom to build custom modules and have better control over the training process.

What reference materials will you use to obtain sufficient background on applying the chosen network to the specific problem that you selected?
Few of the reference materials are mentioned below
https://www.tensorflow.org/text/tutorials/text_generation
https://paperswithcode.com/task/text-generation

How will you judge the performance of the network? What metrics will you use?
We will be using different variations of BLEU and Rouge metrics
https://towardsdatascience.com/how-to-evaluate-text-generation-models-metrics-for-automatic-evaluation-of-nlp-models-e1c251b04ec1

Provide a rough schedule for completing the project.

11/8/22: Finish proposal, create GitHub
11/9/22-11/16/22: Download the code and make sure that we can duplicate the results of the paper
11/17/22-11/23/22: Hyperparameter tune the model in the paper and see if we can
11/24/22-11/26/22: Understand SHAP and LIME, work on the interpretability of paper results
11/27/22-11/30/22: Time permitting, test on another dataset–perhaps on more sensationalist news, like Buzzfeed
12/01/22: Start report
12/13/2022: Submit the project