CSCE 5290
Joshua Higginbotham, Sarah Beaver, John Rutledge
3/27/2022
https://github.com/Sarah-Beaver/NLPClassProject

Parody Generation with Neural Networks

We want to use NLP techniques for humor and entertainment. In our divided, politicized world, with its wars and pandemics and whatnot, sometimes it helps to take a more lighthearted approach. Especially towards people who take themselves too seriously. Like the people at InfoWars.

Our project is a simple one: train a language model on a corpus of featured articles scraped from infowars.com, and laugh and cringe at what it says.

This project builds directly on a previous class project (specifically for AI Software Development) in which one of our team members took a leading development role. You can see the final results in their Google Drive folder.

The previous project used a simple GRU network with sparse categorical cross-entropy loss and no smoothing to demonstrate the effects of different hyperparameters on model performance, such as number of training epochs, whether it was per-word or per-character, and the sequence length of training inputs.

However, much could be done to improve the code and outputs. The previous project had to spend a great deal of time figuring out how to preprocess the corpus, how to get a single codebase to handle character and/or word based text generation depending on the configuration, and finally had to spend the time to train a model for each combination of hyperparameters.

For this project, we plan to make several improvements. We hope that they will increase the humor value of the model's output and make the code base easier for future teams to expand upon.

Increment 1

- Change GRU to LSTM
- Update model to use Tensorflow's object-oriented interface rather than function interface
- Strip out the character-based code since word-based was better anyway
- Change to read data from Gist rather than Google Drive
- Migrate from Google Colab to modular Python scripts on GitHub, including:
    - Change hard-coded hyperparameters to command-line arguments
    - Split training and generation into separate scripts

Increment 2

- Update preprocessing to leverage NLTK and/or Spacy
- Implement GAN loss function
- Implement smoothing so that vocab doesn't have to be restricted and unknown words can be handled
- Hyperparameter tuning

References

1. Previous project Google Drive: https://drive.google.com/drive/folders/1rA2yyJYlS2sHo-gfeSOrDG3JL9ZHRx4G?usp=sharing
2. Text Generation with Tensorflow: https://www.tensorflow.org/text/tutorials/text_generation
3. InfoWars: https://www.infowars.com/