Learning Horror Language Models: Provoking Emotional Responses

Thomas Higginson, Kailash Jayaram, Kelvin Lui, Melissa Hui

ABSTRACT

Considering that the purpose of machine learning is to be able to create programs or systems that learn how to deal with certain data sets that are presented to them, the opportunity to explore particular neural network abilities becomes increasingly vast and limitless—especially with sufficient training and efficient algorithms. In our work, we touch upon the idea of imitating the horror genre, a distinctly unique literary genre. With this in mind, we gathered a handful of short horror stories to use for studying the efficacy of training deep neural networks to capture the notable and prominent language structures represented in the genre, and subsequently using a trained model to replicate the genre. By analyzing the coherence, resemblance, and plausibility of our generated sentences, we will obtain a deeper look deeper look into the value of our deep learning networks and insight on the efficacy of using such models to emulate specific subgenres of literature.

1. Task

The ability to automatically generate literature that is simultaneously horror-like and also emotion-provoking is a challenging task. One challenge is the level of subjectivity that comes with analyzing the resulting text. Questions that may arise, for example, include:

1) Can the generated text exhibit strong enough persuasive imagery or intense language that is indicative of the horror genre?
2) What characterizes a sentence, for example, as falling under the category of horror? Is it the punctuation, word length, or terminology? Is it all of them?
3) Would we be able to specifically distinguish the generated results from other literary genres and subgenres?

In order to tackle these initial questions, our group has decided to take an objective approach, where we intend to implement and compare different preprocessing and model techniques to accomplish our goal, and then empirically analyze the generated samples from our trained models. With respect to subjectivity, our proposed strategy is focused on incorporating crucial terms to rank the generated text. However, we encounter numerous implementation issues and ultimately did an empirical measurement of results. Also, with regards to the relatively limited published research on training deep learning neural networks to replicate the semantics and diction of a literary genre, our group attempts to discover and hopefully coagulate the possible procedures and model techniques that we believe will be essential to producing conclusive and pragmatic results.

2. Data

Our model was trained on a precleaned dataset provided by the popular data science website Kaggle (https://www.kaggle.com/c/spooky-author-identification/data). The data is a set of sentences created via "chunking larger texts into sentences using CoreNLP's MaxEnt sentence tokenizer," and contains only text written by the following spooky authors: Edgar Allen Poe, H.P. Lovecraft, and Mary Wollstonecraft Shelley. Moreover, the dataset has 27,971 unique sample sentences, 28,727 total unique words, 744,802 total words, and an average sentence length of 27 words. Our group decided to not remove punctuation from the text; and split commas, periods, exclamation marks, and question marks into their own respective tokens.

3. Model

Our initial goal was to implement both a Long Short Term Memory Cell (LSTM) and a vanilla Recurrent Neural Network, testing both with pre-trained word embeddings. We encountered difficulty doing so because implementation and computational troubles. That being said, our group successfully implemented a LSTM using pyTorch. Note, we had to reduce our dictionary of unique words to the top 5000 most frequent to reduce our space and time complexity to be reasonable. Our model input is a vector of 10 words, and each word is mapped to a 50 dimension word embedding—we decided to train the word embedding alongside the LSTM. After embedding the words, we use an instance of the *nn.LSTM* class with the specified hyper parameters:

- *learning rate* = 0.01
- *num_layers* = 3
- *hidden_size* = 50
- *dropout_rate* = 0.20

With the specified model parameters, the LSTM implements a forward propagation algorithm by looping through the list of word embeddings to retrieve all the hidden states in the last layer and the hidden states after the last timestep. Following, the model stores the tensor containing the output features from the last layer of the LSTM, for each t, and then applied a log softmax function to predict the next word. Specifically, the output is a vector of word probabilities of the next word given the initial input. Additionally, the hidden layer is saved such that it can be used in the next timestep.

4. Results

After training, we sampled results by specifying 10 words that represent the beginning of a sentence to our model, and sampling 10 next words. We repeated and recorded the end result for 100 iterations. The procedure to generate samples is as follows:

1. Declare 10 starting words in an array as input word vector.
2. For 100 epochs:
    a. For 10 more words:
        i. Use input word vector as input for LSTM model.
        ii. Sample a next word from softmax function output.
        iii. Append sampled next word to end of input word vector.
    b. Append full 20 word sentence to results array.
    c. Re-initialize 10 starting words.

So, from our output of next word probabilities, we take a single sample, append the chosen sample next word to the initial to increase our generalizability from training on the dataset. After multiple tests on numerous different starting words, we believe our model performed decently. A few of the top results of the model can be found in the table below. From the results of the model we can see it picked up on some syntactic elements of language such as adding both starting and ending quotes in the second sentence in the table. It also learned many simple relations such as periods following abbreviated titles (e.g. Mr. or Mrs.) and adjectives preceding nouns in many cases. In our case, adjectives properly preceding nouns is an accomplishment because it represents the intensive descriptive nature of the spooky genre—creating spooky imagery for the story. In our evaluation, we considered the use of provocative adjectives and adverbs in more results than less, an underlying trend that the model learned to use a wide vocabulary. That being said, there was no full, coherent sentences in any of our samples. We concluded that the possible reasons for that are because our sampling method is randomized, our training corpus is relatively small, our dictionary is relatively small, or our word embedding method.

However despite not forming complete sentences, the words seem to take on some of the horror aspects of the training data. The outputs like “we were dead.” and “the pain of my superior” as well as “the lurking baron” seem to be phrases that would come right out of a spooky novel. In our research of other similar projects we found few models that were able to fully structure complete sentences using training data of a similar size to ours. So despite the fact that none of the text generated from the model really read like a book, we feel that model performed very well. Not only did it manage to pick up on some of the structure of the English language but it also did a decent job of achieving our goal of capturing the writing style of the horror genre in the phrases that it learned to generated. Overall the LSTM model with an embedding layer as part of the network was by far the best model we tested. It outperformed many of our expectations of the difference a model trained on only horror novels might have compared to one trained on a variety of novels. It clearly fit to the horror genre, despite the small dataset and hardware constraints that forced us to use smaller hyperparameters for the model.

The output with text seed: “The details were examined and it was found that”

dwelling of the balloon had a certain hide rang to
waves were desperate. “we were dead.”
success of centuries itself might be.” “it is
lurking baron should become the, and had been,
consequences of evening. “god, the reality of autumn
silent imagination was no longer five alone.” a
pain of my superior.” continued the and a