Generating STEM Related TED Talks using Deep Learning

By Tomas Escalante, Roberto Ventura, and Grant Gambetta

Abstract

For this project, we decided to investigate deep learning models for text generation. Specifically, we were interested in generating Science, Technology, Engineering, and Math (STEM) based TED talks. To achieve this goal, we developed two different deep learning models: a multilayer LSTM and fine-tuned GPT-2. The results of the two models were compared.

1 Introduction

In recent years, state-of-the-art techniques in deep learning have advanced very rapidly. In terms of Natural Language Processing (NLP), this has gone from RNN, LSTM, and GRU models to models such as BERT, GPT, and XLNet. All of these models have been proven to produce human-like results for various NLP tasks. In this paper, one NLP task, text generation, will be attempted using a multilayer LSTM baseline model and a state-of-the-art pre-trained GPT-2 model.

2 Goal

Our minimum goal for this project was to develop a deep learning model that would predict the next word of a sentence and repeat this process until the model generated multiple sentences. We were able to take this idea one step further and by the end of the project, our model was able to generate multiple sentences at once based on the prompt(s) provided by the user.

3 Project Overview

3.1 TED Talks Dataset

We were able to find a TED talks dataset on Kaggle ([1]) which consisted of roughly 4000 unique transcripts in twelve different languages. For the sake of our project, we used only the English transcripts. The transcripts in the dataset covered a wide range of topics, so we decided to pick only science and technology related transcripts. To do this, we used some of the functions from the "TED - Word Clouds" notebook written by Miguel Corral Jr. ([2]). After we filtered the data on our topics of interest, we had 3096 unique transcripts to train our models with.

3.2 Data Preprocessing

Preprocessing the data was very important for worthwhile results from the LSTM. To begin, the 3096 separate transcripts were combined to create one large transcript of over 4.5 million words. After this, the combined transcript was filtered for symbols such as colons, semicolons, dashes, slashes and numbers which were removed completely while symbols such as periods, question and exclamation marks and ellipses were converted to '<eos>'. Once this was done, the transcript was used to create a dictionary of unique words which totaled to about 50000. Of these 50000 words, the 10000 most frequently occurring words were kept and the rest were converted to '<unk>'. With our transcript the way we wanted it, we were able to change from text to indices and split into sequences, or arrays, of data and labels. These sequences were of length 32 where each data sequence was [w_n , ... , w_{n+31}] and each label sequence was [w_{n+1} , ... , w_{n+1+31}] which resulted in a dataset with a shape of about 4.5 million by 32 and labels of the same shape.

3.3 Model Training

To achieve our goal, we opted to develop two different deep learning models: a multilayer LSTM (baseline) and a fine-tuned pre-trained GPT-2. For the LSTM, we trained it on 90% of the preprocessed dataset and then validated and tested it on the remaining 10% of the preprocessed dataset (5% for validation and 5% for test). With regard to GPT-2, we were able to simply convert the transcripts from the raw dataset into a list of transcripts and then split the data into the training and testing sets without having to do much preprocessing. This was because the pre-trained GPT-2 model automatically created sequences of a certain length as the training began. For fine-tuning the pre-trained GPT-2, we trained it on 80% of the dataset and then tested it on the remaining 20% of the dataset.

4 Results

The purpose of this project was to generate STEM related text, similar to the format of a TED talk. For consistency, we fed the same initial prompt to the GPT-2 model as well as the fine-tuned GPT-2 model to view the differences in their outputs. In the vein of TED talks, “Hello everyone, today I'll be discussing...” was the prompt we used.

Using the GPT-2 model, an example of the text that was generated for the previously mentioned prompt was: “Hello everyone, today I'll be discussing the fact that I'm a vegetarian. My food choice was probably the easiest question and it didn't get much better than today. One of my coworkers called to ask if I wanted to eat chicken breasts. I gave him the answer – yes, I did – and he was not so unkind that he got angry that I was not eating them. When I asked him again, he said, I'll eat it. You'll be fine.” This generated text is not related to STEM and this is most likely because the prompt is very open ended and therefore the probability that the model would predict STEM related text was low.

Then, using the fine-tuned GPT-2 model, the following text was generated for the same prompt: “Hello everyone, today I'll be discussing these two topics. First, we're going to talk about one of the great innovations in artificial intelligence, called natural selection. Our intuition says that if a computer can think about certain possible actions, it can do these actions in a certain way. We see this effect with our social systems, such as social networks. We can think about what social groups look like and what people look like. Then, our intuition, which says that a computer can think about certain things about those things, will say that it can actually interpret human actions.” As we can clearly see, the generated text from the fine-tuned model was much more STEM related than the generated text from the non-fine-tuned model.

To evaluate our models other than by the content of the generated text, we opted to use loss and perplexity. Below is a table of our average results for perplexity and loss for the LSTM, GPT-2, and fine-tuned GPT-2.

Model	Perplexity	Loss
Multilayer LSTM	217.892	5.3840
GPT-2	32.9132	3.4939
Fine-tuned GPT-2	28.8572	3.3624

As one would expect, the perplexity and loss changed every time we trained the models due to the randomness inherent in the models, as well as the modification of hyperparameters. The perplexity and loss of the models were always in this general range and allowed us to clearly understand how the models were performing compared to one another.

5 Discussion

The performance of GPT-2 was clearly much stronger than the performance of the multilayer LSTM based on the perplexity and loss. This is to be expected as GPT-2 is a pretrained, state-of-the-art model which we simply fine-tuned to our data. As shown, when passing the prompt through GPT-2, the output was a very general discussion while it became STEM-related after being passed through the fine-tuned GPT-2. In order for the LSTM to be as effective as GPT-2, we would most likely need to train it with more STEM related data.

The LSTM results were not mentioned because the text generated by this model did not flow well. This occurred in the different attempts in making this model. One model predicted that an end of sentence would occur after the introduction and did not make any further progress. Another tuned LSTM model did have more variety in word choice, but the choice of words became the same cycle after a few words. Overall, this baseline model was not successful.

6 Challenges

One of the biggest challenges we faced was dealing with the raw data; it took a lot of time and effort to figure out how to format and clean the text. Initially, our results were nowhere near satisfactory with losses above 6.5 (perplexity > 665) which was due to errors in our preprocessing. After this, the most challenging part was working out the kinks in our LSTM code to produce better results. Another major setback for our results came from the size of our data which was very costly while training our LSTM. Because of this, we constantly reached usage limits after hours of training which required us to restart and lose our progress. With regard to GPT-2, the main challenge we faced was a GPU memory issue which occurred when we tried to increase the sequence length. This issue prevented us from using larger sequences when training the model, and therefore we had to keep the sequence length at 128 or less in order to avoid this error. Being able to use larger sequences most likely would have improved the training time of the model.

7 Conclusion & Future Steps

From the initial goals that were set, the project was successful in generating text similar to a STEM based TED talk. To accomplish this goal, two models were created, a LSTM model and a GPT-2 model. The LSTM model did not accomplish the initial goal because it did not do well in generating text or choosing a topic with a given introduction. As expected, the GPT-2 model outperformed the LSTM baseline because the model had a smaller perplexity and generated text that flowed well. With an introduction fed as the input, the created GPT-2 model was successful in generating text that would start with a STEM related topic, then continue to discuss this topic until it reached the word limit set by the function.

With regard to future work, improvements can be made on the current LSTM and GPT-2 models to improve the results gathered. One improvement that could be implemented to get better results on the LSTM model is to replace the embedding layer with a pre-trained embedding such as BERT. Another potential task to improve both the GPT-2 model and LSTM model is to pick a single topic instead of STEM related topics used in the project. This would potentially make it easier for the model to choose a topic to discuss and know more information overall about the topic which could improve the diction of the generated text.

Acknowledgments

We would like to acknowledge the following Python libraries that we used for this project: PyTorch, simpletransformers, NumPy, pandas, and Matplotlib. Additionally, we would like to acknowledge the preprocessing code which Miguel Corral Jr. wrote for the TED Talks dataset. These functions made it much easier for us to select the topics we needed.

References

[1] Jr, Miguel Corral. “TED – Ultimate Dataset.” *Kaggle*, 5 May 2020, www.kaggle.com/miguelcorraljr/ted-ultimate-dataset.

[2] Jr, Miguel Corral. “TED - Word Clouds.” *Kaggle*, 5 May 2020, https://www.kaggle.com/miguelcorraljr/ted-word-clouds