The Text Generator About Nothing: Creating Arbitrary Seinfeld Episodes: Project Update

Harrison McCabe

Abstract

For my SI 630 course project, I am working on creating a text generator that when trained on scripts of Seinfeld, can create randomly generated scripts of the show. I hope to build four types of models, two as a baseline, and two as advanced. The data (i.e. captured scripts) will be collected from a web scraper and stored locally. Related work shows that such a project will be challenging, but still possible. The models created will be evaluated using perplexity and BERTScore. So far, I have completed a line extractor for a baseline and somewhat of a gold standard for complex models. Finally, I propose a work plan that will have me working on the project throughout the rest of the semester.

1 Introduction

Seinfeld was a popular American TV sitcom that ran throughout the 1990s. Although there were 180 episodes, many fans were disappointed that the show only lasted nine seasons. This begs the question of what would happen if Seinfeld continued for another season? What impact would each character make? How 'Seinfeldy' would the episodes be?

As a fan of the show, I am curious to see what makes Seinfeld unique. To do so, I want to see what a randomly generated episode of Seinfeld would look like. The show still has tens of millions of fans who watch syndicated episodes on Netflix, and it's likely that many of them would be interested in seeing new episodes.

Additionally, dialogue generation is a topic of interest in the field of Natural Language Processing. Building a model using a relatively small specialized domain, such as the episodes of a single TV, can provide valuable insight to researchers. Even though this specific example is a novel domain, the methods used could be extended to other contexts.

2 Data

The website https://www.seinfeldscripts.com/ contains scripts of Seinfeld episodes that are captured and reverse engineered by fans of the show. Each episode has its own page on the site.

In order to get the scripts, I built a web scraper using Beautiful Soup to automate the process by taking advantage of a somewhat consistent HTML format among each episode's page. In total, I was able to scrape 120 episodes worth of text, which is 2/3 of all episodes written.

3 Related Work

This project dives into the domain of text generation, which is an area that is well studied by NLP researchers.

One part of my project is that I need to create meaningful word embeddings, and that process can be optimized with the right model and hyperparameters (Levy et al., 2015). The parameters that they use will be a good reference point for the ones that I choose.

A challenge to my project is that I am only working off a corpus of 180 TV episodes, containing a few thousand scenes, whereas many state of the art ones are trained on millions of pieces outlets. This means that I may need to consider what types of non-conversational data I should include in my model. (Su et al., 2020) suggests that dialogue generation models can be improved with this information.

Another challenge of my project is that the text generation I am creating is more large scale. It's much more than just a sentence or two, i.e. what most prior work has focused on. However, research has been done by (Guo et al., 2018) as how to leverage GANs to create longer strings of text.

One way to think about this project is that it is generating dialogue, as if a human was having a conversation with a chatbot. Many high performing deep models accomplish this do so by focusing on the following properties of human conversation: informativity, coherence, and ease of answering. (Li et al., 2016) shows that these models generate diverse, long strings of text.

4 Methodology

The NLP task at hand is threefold. One, tag the parts of speech in the script. Two, use a model like word2vec to embed the context of the words. Three, build a model to randomly generate Seinfeld script using the following four approaches.

1) Random line selection. This baseline technique generates random script by a user selecting how many lines they want, and then that many random lines are extracted from the corpus. The one caveat is that the same character cannot have consecutive lines, just like in real scripts.

2) Word distribution weighted by part of speech. This is a baseline technique that selects random words while trying to make some sort of logical sense with the ordering of parts of speech. I anticipate that it will end up looking like a page of MadLibs.

3) Long Short Term Memory (LSTM). This Recurrent Neural Network is designed to predict the next word in a sequence, given what has already been generated.

4) Diversity-Promoting GAN (DP-GAN). First proposed by (Xu et al., 2018), DP-GAN is a model that encourages diverse text generation, favoring words and contexts that have not already appeared. The benefit of using this for this project is that it accounts for the wide variety of topics that appear in a typical Seinfeld episode.

5 Evaluation and Results

When the models are created. I will need unique ways to evaluate the text generated from it. It's difficult because even though it's created through semi-supervised techniques, there is no true training data with which to compare the output. There is also no ground truth as to what makes a series of dialogue 'Seinfeldy'. Fortunately, there are ways to evaluate randomly generated text. Primarily I will be looking at perplexity which answers the question of given some distribution of text, how likely is a given series likely to occur (normalized to length.

6 Work Plan

I propose the following work plan for myself.

1) By Mar 16th: Create and fully evaluate second baseline model

2) By March 30th: Create and fully evaluate LSTM model

3) By April 10th: Create and fully evaluate DP-GAN Model

4) End of the semester: Finish writing research paper and publish findings.

References

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *CoRR*, abs/1606.01541.

Hui Su, Xiaoyu Shen, Sanqiang Zhao, Xiao Zhou, Pengwei Hu, Randy Zhong, Cheng Niu, and Jie Zhou. 2020. Diversifying dialogue generation with non-conversational text. *CoRR*, abs/2005.04346.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949, Brussels, Belgium. Association for Computational Linguistics.