Reddit Post Generator

Peter Hafner
Dept. of Computer Science
Western Washington University
Bellingham, WA, USA
hafnerp2@wwu.edu

[redacted]
Dept. of Computer Science
Western Washington University
[redacted]

[redacted]
Dept. of Computer Science
Western Washington University
[redacted]

Abstract

We focused on creating a title and body text generator that tries to replicate the way that posts from r/WallStreetBets subreddit sound like. Using a small Kaggle dataset of posts from r/WallStreetBets, we fine-tuned a pre-trained GPT-2 model on the body of the post. Using transformers, PyTorch, and CUDA library with GPT-2, we were able to create a fine-tuned model which generated body paragraphs based on the posts from the Kaggle dataset. Our model generates text from a keyword and summarizes it to produce a related title sentence. We combined fine-tuned content generation with text summarization techniques to achieve more "human like" text.

1 Introduction

This project focuses on the development of language models and explores the possibility of training a model to establish a usable internet presence. With the release of ChatGPT 3.5 and its versatile applications in everyday life, it's essential to consider how large language models can imitate real humans based on given training data. The paper addresses the risks of trusting generated text from language models. This research aims to achieve a usable chat bot capable of generating popular community posts with minimal mistakes. It highlights the harmful implications of the relationship between humans and large language models and the dangers of believing information that is being generated. This research heavily relies on the capabilities of our own machines and the assistance of Google Colab to fine-tune a pre-trained model of GPT-2. This research deepens our understanding of the dynamic between humans and language models to find the ethical implications in real world context.

2 Related Works

The evaluation of text generation models is a rather new topic with the recent advancements in the accuracy of text generated models. Looking at Zhang, Tianyi, et al there is a BERTScore test that can be conducted to automatically evaluate text generation models. In Zhang, Tianyi, et al they found that BERTScore was able to successfully evaluate a models ability to appear human. To summarize simply "BERTScore computes a similarity score for each token in a candidate sentence with each token in the reference sentence" (Zhang, Tianyi, et al, page 1). While other evaluation models look for exact matches, BERTScore compares the tokens using similiarity. This allows for the BERTScore model to better evaluate if a model mimics human judgement.

We also deep dived into the requirements of datasets to train models well. In Radford, Alec, et al. they dive deeper into how GPT-2 performs given "sufficiently large and diverse dataset" (Radford, Alec, et al. page 10). We are also using GPT-2 as our base pre-trained model. Radford, Alec, et al. mentioned that their best model that performs at the state-of-the-art was 1.5 billion parameters and was trained on millions of webpages called WebText. This provides a little more context to our results mentioned later. To summarize, our dataset consists of about 52,000 posts from r/WallStreetBets, but unfortunately about half of the data points have

empty bodies so we have to leave those out of the training dataset.

3 Experiments

3.1 Dataset

The dataset utilized for this research was sourced from Kaggle's Reddit WallStreetBets Posts (Kaggle dataset), consisting of scraped posts from the Reddit WallStreetBets community. Created by users who share insights on the current state of Wall Street and recommended stocks for investment, the dataset includes post titles, bodies, and their corresponding scores. During pre-processing, we filtered out rows containing non-numeric values, most likely to be image posts, which would not contribute to text generation. This left us with around 18000 rows left in our dataframe. Additionally, as a group, we decided to exclude posts based on their scores, considering factors such as upvotes, downvotes, shares, reposts, and comments. When implemented this pre-processing, the amount of rows left in our data would be largely based on the score given on each post, focusing on highly-rated posts. This decision was made to optimize data training within the allotted project timeframe, and also aimed to use better content for post generation.

3.2 Experimentation

With research, we gained an understanding of the most efficient ways to implement a chatbot using a large language model. Considering, the scope of the project, the time allotted, and our current knowledge of training models, we concluded on using a pre-trained model to generate text, which ended up being GPT-2. Before training the data, we performed normalization techniques on the body and title columns of the dataset. We converted all the text to lowercase and removed punctuation and stopwords. When creating our first model, we encountered issues where the model would generate long URL links in the posts. To address this, we included URL stripping in our normalization process. Using the GPT-2 library, we attempted but failed to input NLTK's form of tokenization into the model. Instead, we used GPT-2's tokenizer, which splits the text into tokens by whitespace and then converts the data into encoded IDs that GPT-2 can read. HuggingFace contains the pre-trained model of GPT-2, which we successfully loaded into our code. After running the tokenizer, we needed to define our training arguments.

Due to the size of our model and the limited computing power of our personal machines, we had to set our epochs and batch sizes to relatively low values. Despite having relatively powerful GPUs set up with CUDA, our batch size had to be limited to only 4, as anything higher than that would exceed our GPUs memory. The model was fine-tuned using the Kaggle dataset and was able to generate text when given a prompt or a starting word. The current model generates text for the body, and instead of training another large model, we used a BART model to summarize the text to serve as our title. Research showed that BART models are particularly good at summarizing, so we used a pre-trained BART model and got reasonably believable results from this.

3.3 Evaluation

Due to the unsupervised nature of our model fine-tuning, it is difficult to achieve a numerical value for evaluating our model. We propose that the best way to evaluate the model would be through qualitative human evaluation. This could be done in a number of ways, such as a survey with human judges or potentially a blind comparison. A survey could be conducted where human judges (ideally people familiar with the WallStreetBets community) are given a list of generated posts and titles, then asked to grade the posts on a Likert scale based on whether or not the posts were believably written by a human. Our posts could also be compared to random, genuine posts from WallStreetBets, then a blind test could be conducted asking someone unfamiliar with the project which post was generated by a human and which was generated by AI. Due to time constraints neither of these options was feasible in the scope of this project, but this could be a future continuation of the project.

While the previously discussed evaluation method could lead great insights, it does not truly quantitatively evaluate the generative model. BERTScore would be a better way to get a quantitative evaluation of the created model. We were looking into implementing BERTScore into our code, but due to time constraints with the model run time and external factors implementation of BERTScore did not happen. We were looking into using code from Neulab's github which is the official implementation from the "BERTScore: Evaluating Text Generation with Bert" paper.

3.4 Replication

In order to replicate what we did to achieve our results, find the attached zip file and unzip it. We created a virtual environment for this project due to the use of PyTorch and CUDA. Once in that environment, the models are saved and can be loaded using the notebook or new models can be generated using the csv file. The implementation of the text generation and the summarizing is located near the bottom of the notebook. The generation of the body and the title are generated by a randomly selected token from the text, or can be generated from a prompt of choice (i.e. "Today in the stock market").

4 Results and Discussion

Below in Table 1, you can see a handful of generated titles and bodies from our model. The bodies have been cut short in this example, but they generate fairly lengthy posts. You can notice some interesting trends here, specifically the link created in the first body listed. This is not a real link and leads to nowhere, but it certainly looks like a legitimate link at first glance. Also, the titles appear to be accurately tied to the body they were generated from, though they are a little lengthy.

As discussed in the Evaluation section, we were unable to generate any hard data for the effectiveness of our model, however at first glance the results look good.

5 Conclusion/Future Work

To summarize our findings, our model's outputs strongly reflect the characteristics of the input data. The results highlight the potential harm large language models can cause when trained on unfiltered internet data, as they can influence a large audience or use language that targets specific demographics. Additionally, we want to emphasize the dangers of the model generating links within the text, which can make the content appear more trustworthy and credible than it may be.

For future work on this project, there are multiple ways we could develop our models further. We could implement different types of generative models, as well as running models for extended periods of time to train on larger data sets. Evaluation could be improved in the future with the suggestions made in the Evaluation section.

Token	Title	Body
Web	BlackBerry is the largest mobile phone company in the world. BlackBerry has a history of innovation and innovation in software and hardware.	web.com/news/articles/2021/02/06/the-blackberry-gamestop-is-a-game-of-thrones-and-the/ BlackBerry is a company that has been around for ...
for	The SEC has not yet commented on this. If you're wondering why the Feds are not investigating Citadel, you should read the SEC's Notice of Investigation.	for the past few days, I've been seeing a lot of posts about how the SEC is trying to cover up for Citadel. I've seen a few posts saying that the Securities ...
equals	If you are a billionaire and you own a company, you can buy a share of the company at a discount. If you hold a lot of shares, it is a good idea to buy shares at the discount, because the stock price	equals the amount of money you have to invest in a stock. If you are a billionaire and you own a company, you can buy a share of the company at a discount. If you hold a lot of shares, ...

Table 1: Generated Posts and Titles

References

Zhang, Tianyi, et al. "BERTScore: Evaluating Text Generation with Bert." arXiv.Org, 24 Feb. 2020, arxiv.org/abs/1904.09675.

Neulab. "Neulab/Code-Bert-Score: Codebertscore: An Automatic Metric for Code Generation, Based on Bertscore." GitHub, 2023, github.com/neulab/code-bert-score.

Radford, Alec, et al. "[PDF] Language Models Are Unsupervised Multitask Learners | Semantic Scholar." Semantic Scholar, 2019, www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe.

Preda, Gabriel. "Reddit Wallstreetbets Posts." Kaggle, 16 Aug. 2021, www.kaggle.com/datasets/gpreda/reddit-wallstreetsbets-posts.