International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Chhattisgarh Tourism Review Generator using GPT-2

Shubham Giri[1], Saurabh Toppo[2]

[1]*B.E. Student, Dept. of Computer Sciecne Engineering, SSEC, Chhattisgarh, India*
[2]*B.E. Student, Dept. of Computer Science Engineering, SSEC, Chhattisgarh, India*

Abstract – The enormous growth and potential of online consumer reviews of tourism are playing an important role in consumer attitude and buying or renting behaviors. In fact, user reviews have become a very popular tool that strongly influence the buying decision of consumers. By having a lot of reviews online, people will be able to get a feel for your business. Many online travel businesses offer quantitative ratings, textual reviews or a combination of both. Our group sought to replicate different tourist's online reviews using the Open AI's NLP technique based GPT-2 model and transfer learning. We propose a domain specific language model that operates solely based on text input such as tourist place. We used GPT-2 to extrapolate from a given context, allowing to generate any type of reviews like positive, negative or neutral.

Key Words: **Natural language processing, Transfer learning, Online reviews, GPT-2, LSTM, Transformer, RAKE, GloVe**

1. INTRODUCTION

Machine learning systems now excel (in expectation) at tasks they are trained for by using a combination of large datasets, high-capacity models, and supervised learning. OpenAI's GPT-2 is a transformer-based language model trained with the goal of predicting the next word passed all previous words of some text as parameter[5]. Given some prompt, the model has been able to generate similar human-authored text in response to the prompt that makes sense both grammatically and semantically. We wanted to extrapolate this model to generate tourism reviews given a dataset of online tourism reviews to train on, rather than responding to a given prompt. Specifically, we wanted to extract a topic from a given review and train on generating reviews about said topic.

In this paper we analyze to use Open AI's NLP technique GPT-2 to recognize different style, contents of online tourism reviews and to generate similar reviews for different destinations and places. By utilizing transfer learning and building up GPT-2 text generation model, we tried to train a model for the domain specific task of generating reviews based on a given tourist place. We scraped about 6000 online reviews from the Trip Advisor website. We used RAKE to extract a list of keywords, and then for each keyword, we used GloVe to generate a set of similar words that were appended to a "history" for the reviews. Finally, based on the dataset of reviews with their corresponding histories, we were able to generate new reviews via GPT-2.

2. RELATED WORK

2.1 Recurrent Neural Networks and LSTM for Text Generation

Recurrent neural networks, mainly, long short-term memory networks, are profoundly used for text-generation tasks. Sequence to sequence encoder/decoder models, such as that provided in by Sutskever et. al. (2014) [1], utilize LSTM to map sequences of text to a newly generated sequence of text, which has been shown to be effective for translation and dialogue texts [2]. LSTMs have also been used in previous works on text generation; the very similar existing work to our project is that of MIT postdoc Bradley Hayes: @DeepDrumpf, a Twitter bot that posts Trump-like tweets. Hayes's model uses a recurrent neural network that generates tweets one character at a time. [3].

2.2 Generative Pretrained Transformer-2

The GPT-2 is a transformer model, which is an upgrade of GPT by Open AI, that alleviates the computationally expensive training problem by neither using recurrent nor convolutional neural networks and only using attention mechanisms, which has allowed for text-generation models that are much less computationally expensive [4]. OpenAI has trained a state-of-the-art transformer based language model on over 8 million online documents [5]. These 40 GBs of text, coined WebText, were curated by scraping the web with an emphasis on document quality. The model OpenAI created, GPT2, expanded on their original with slight layer normalization tweaks and an expanded vocabulary and context size. Without fine-tuning, it broke previous accuracy records for seven out of eight datasets. Indeed, it outperformed these previous records without ever actually training on the datasets themselves.

OpenAI's team conclude their paper with a recognition that their work has purely examined GPT-2's zero-shot performance. That is, how it has been able to generalize its unsupervised training to new problems. However, they speculate that GPT-2's potential far exceeds these metrics as it can easily be fine-tuned to tasks using transfer learning. OpenAI has released all four models GPT-2_117M, GPT-2_345M, GPT-2_774M and GPT-2_1.5B parameter version of their model.

3. Extract Features from Dataset

We took the help of online available web scrappers to create our dataset, we used web scrapper to run on trip advisor website to collect reviews of tourist visiting one of the states of India, Chhattisgarh. Web Scrapper allows you

to scrap as many reviews available for a particular tourist place in their review section of website. These reviews will include one word or emojis, which we had to filter out given that we wanted to capture the specific style of each tourist. This left us with around 1200 reviews for each place we looked at. We then processed each review, filtering out common emojis, images and links which left us with plain text for each entry. Often links and images are often an important part of a reviewer's persona, we chose to filter these out as the aim of this project was to capture the grammar and word usage of a reviewer using GPT-2, which was not built to deal with images and links. Along with the filtering, we also used keywords to generate context or a "history" for each review. Using RAKE (Rapid Automatic Keyword Extraction), we extracted unigram and bigram keywords from each review text. From there we used GloVe300D vectors to find the 10 most similar words to each key word, and used the aggregation of these similar type of words as the context for each review text.

Thus, we see that for the review:-

"Ambuja City mall is the biggest, fun loving and shopping mall in the Raipur, the capital city of Chhattisgarh"
A simplified context could look like 'Biggest fun loving shopping mall Raipur Chhattisgarh'. We used these contexts to help our model develop an understanding of what reviews on similar topics look like. This also allowed our model to generate reviews on topics that were tangential to the subjects of the reviews in our dataset, instead of only being able to generate reviews directly from the content matter of the dataset.

3. MODEL

3.1 Training

Given the RAKE then GloVe extrapolated context for each tourist place's review, our training algorithm then generated training and validation examples. Following the single-input method described in Golovanov, Sergey, et al.[2019], our model was trained using a reviews's context concatenated with the actual review.

To specify which place a review is for, we have added the location's name to the starting of the input sequence. In addition, we use a sentence segment embedding layer to denote which parts of the input were the location's name, context for the review, and review itself. Finally, since GPT-2 is not a recurrent neural network, a positional embedding layer is added to give the model a concrete sense of position for each word.

3.2 Multi Task Loss

We tried to fine tune GPT-2 to generate reviews in a similar style to a given review of a location or tourist place. We used a multi-task loss defined as a linear combination of language model loss and Multiple Choice Loss. This combination of loss functions was found in Wolf, Thomas, et al. [2019] to drastically increase dialogue readability and adherence to context. Language model loss is defined as the cross-entropy loss applied to a softmax of the transformer decoder output with the example review words as the labels. Given the word sequence S = {w 1 , ..., w |s| }, language model loss is defined as described in Wolf, Thomas, et al. [2019], Multiple Choice Loss involves adding "distractor" examples for each true example in the dataset. As previously mentioned, these examples have the context of a previous review, but the training review itself is randomly selected from other location's review. Multiple Choice Loss involves calculating a final hidden layer after the sentence has ended. This final hiden layer, h l , is used as the input to a linear classifier layer to correctly classify the true example among distractors. During training, this classifier is trained jointly with the transformer model using the loss function:

$$l_{mc}(S) = -((y \log(\sigma (h_l * W_h)) + (1-y)\log(1- (h_l * W_h))$$

Where W h is learned during training, y is defined as 1 if the example is the true example and 0 otherwise, and σ is the sigmoid function.

At this point that it is worth mentioning that both of these losses, and the multiple choice classifier are automatically generated in our code during training using the Huggingface transformers library for Pytorch. It is open-source and dramatically reduces the barrier to entry into NLP using transformers.

3.3 Model Specifics

Our final model was trained on approximately 6000 reviews. It used an equal weighting of language model and multiple choice loss for 7 epochs. It trained with a learning rate of 4.175e −7 and batch size of 3. Using google colaboratory and GPU, the model only took around 25 minutes to train, since the training set was small.

3.4 Review Generation

To generate reviews, the output of the decoder must be extrapolated into words. GPT-2 takes an input and returns an array of probabilities for each word in its vocabulary corresponding to the probability of that each word follows the input. To generate text, we simply take one of these words and append it to the input and rerun the model. However, choosing which word to append given the probability distribution is difficult. The baseline is to use Greedy Decoding, which takes the highest probability word. Interestingly, this has been shown to not generate lifelike sentences. In our model, top k filtering was used to generate reviews. This involves always sampling from the top k number of words from all word probabilities. This addition of randomness among the top probabilities was found by Fan, Angela, Mike Lewis, and Yann Dauphin. (2018) to lead to more variance in sentence structure as well as longer sentences. In this case, k = 4 was used.

4. CONCLUSIONS

Use of transfer learning in converting GPT-2 to our domain specific task proved to be more successful than any other approach like using an LSTM. Using contexts from one location's review as the input for a model trained on othe location's reviews, we were able to generate every kind of positive, negative and neutral reviews. Even though, we did not spend a significant amount of time fine-tuning our model; however, for our goal of generating location specific reviews, transfer learning on top of a transformer model proved more worthwhile. Tourism review is not so aspect based domain, it will highly require its own metrics. Thus, additional tweaking of current evaluation techniques and more research needs to be done in order to more robustly analyze our reviews.

ACKNOWLEDGMENT

Many people helped us in this project but specially we would like to acknowledge the efforts of these two professors who helped us making this project successful and guided us with their knowledge:-

1. R.S. Panda - Ex-HOD Computer Science Engineering, SSEC, Chhattisgarh, India

REFERENCES

[1] Sutskever, I., Vinyals, O., Le, Q. "Sequence to Sequence Learning with Neural Networks." 2014.

[2] Fan, Angela, Mike Lewis, and Yann Dauphin. "Hierarchical neural story generation." arXiv preprintarXiv: 1805.04833 (2018).

[3] Hayes, Bradley. "DeepDrumpf." Twitter account.

[4] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 2017.

[5] Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI Blog 1.8 2019.