CS 229 Machine Learning Final Report: Rewriting Children's Stories for Different Reading Levels

Project Category: Natural Language

Project Team: Feiyang Liu (SUNetID: liuf9); Yiqun Tian (SUNetID: tianyq); Bochen Zhang (SUNetID: zhang772);

Introduction

Most children learn to read by – well – reading books. However, the task of "leveling" books appropriately for the right reader is quite difficult for teachers. "Fun" high-interest stories often span across levels and are inaccessible to readers of lower levels. Specifically leveled stories exist, but they are often boring, leaving older kids who fell behind unmotivated. This project aims to use and modify existing language models leveraging machine learning techniques that can re-write a children's book story for "any" reading level.

At the core of this project is language modeling. We found there have been many successful language models out there for general natural language processing such as the self-attention architectures like the Transformer [3], the OpenAI's GPT-2 [2] and encoder-decoder RNN architecture[5]. OpenAI's GPT-2 model is a great example of a language model based on the transformer concept that predicts words using given tests. The model was pre-trained with large amounts of words and has shown potential in a broad set of capabilities, including the ability to generate conditional synthetic texts of unprecedented quality [2].

The project team's objective is to re-generate stories based on existing storybooks picked by the education provider for any reading level defined. A dataset of 156 transcripts from children's books was the point of departure to generate new stories along with their levels and word lists to train a language model for generating new stories that fits corresponding levels. State-of-art natural language processing models were used to not only recognize the leveling of the stories but also generate appropriate content for children in different levels. The generated stories were evaluated using both qualitative and quantitative metrics to define the performance of the model selected.

Datasets

The project sponsor provided a dataset of 156 unpublished books in the format of text files with a total of around 52000 words and 4200 lines. The stories were not coming with a specific level from the publisher and we can hardly find a universal leveling criteria to group the stories. Therefore we define leveling using the nature of the stories: the number of words in the story, the standard deviation (std) of the number of words in each sentence, the number of distinct words in each story normalized by the number of sentences (mean and std), the number of repeated words in each story normalized by the number of sentences (mean and std), and the standard deviation of the word position (i.e., indices) of the repeated words among sentences as the potential features for leveling. Then we run the PCA model on the generated data and based on the review of data point locations on the first two principal components and the corresponding stories, we discover that the data points scattered on the outer region tend to have a complex story structure and the data points concentrated in a cluster tend to be simple. So after selecting through different functions $f: R^2 \rightarrow R$, we project each data point on a line, and finally split the dataset into 3 levels, each with 52 examples, according to the ranking given by the values of data points on this line:

$$V = a \cdot PC[1] + b \cdot PC[2]$$

By adjusting a and b, we retrieve a fairly reasonable leveling and the resulting examples for each level are attached in table 1.

Table 1. Examples for the Three Levels

Level	Reference Score	Example
Level 0	70 ~ 1500	It sounds like a poltergeist army is holding a midnight parade, but it's only the alley cat chorus in a brassy backyard serenade.
Level 1	-230 ~ 70	I am walking down the street when I hear someone crying. It's a bear! He looks lost and afraid.
Level 2	-370 ~ -230	Here is a jet. The jet is wet. Here is the pet. I met the pet. The pet is wet!

Methods

1). RNN Style Transfer Model:

This model aims to build a system which can transfer a positive text style into negative text style. And the main content will still remain. The work is non-trivial since we don't have enough data to use. Therefore, the model uses encoder and decoder to disentangle style attributes and content given only unaligned sentences[1]. Figure 1 shows the model architecture:

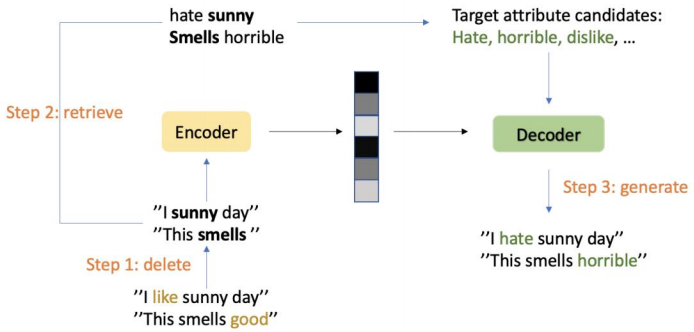

Figure 1. RNN Style Transfer Model Architecture

We then generate word dictionaries that are ordered by the frequency of the words in the leveling datasets so that we can project our target dictionary for training purposes.

2). Character-based text generation RNN model:

This model will generate text by predicting the next character with different times. With a given sequence of characters from the story, the model will train to predict the following character at each time step. For training, we need (input, label) pairs where input is the current character and label is the next character. The model contains three layers: 1) the input layer: a trainable lookup table that will map each character-ID to a vector with embedding dimensions; 2) a type of RNN with size units; 3) the output layer, with vocabulary size outputs. It outputs one logit for each character in the vocabulary. These are the log-likelihood of each character according to the model. Figure 2 shows for each character the model looks up the embedding, runs the GRU one timestep with the embedding as input, and applies the dense layer to generate logits predicting the log-likelihood of the next character.

3). GPT - 2 Model

Open AI's GPT-2 is a large transformer-based language model with 1.5 billion parameters, trained on a dataset of 8 million web pages. With literatures released by the Open AI team, the GPT-2 model has achieved unprecedented performance without ever actually training on the datasets themselves.[2] In this project, we intend to use a smaller version (117M) of GPT-2 models for less computational power is required to fine tune the model.

Considering the limited datasets we obtained (156 stories divided into 3 levels), the fully trained GPT-2 would provide advantage for obtaining the features that the model obtained in the previous training. Without training on specific domains, GPT-2 displays potential in a broad set of areas, including the ability to generate conditional synthetic text samples of unprecedented quality. The Open AI team also discovered that the generated results adapt to the style and content of the conditioning text [2] which links directly to the purpose of this project: to generate stories for each level. Furthermore, in transformer models, a sentence's topic is never "forgotten" by the model when generating the next word compared to language models using recurrent neural networks.

Given pre-leveled story samples, we could fine tune the model using the stories we obtained from our sponsor. We first split our data into training samples, validation samples and test samples for each level. With computational power limitations, we set batch size to 1 in order to run the project on CPU. Then we encode our example for each level with a tokenizer and fine tune the GPT-2 model with temperature of 0.8 (float value controlling randomness) and 20-40 words considered for each step (token). As a result, we obtained fine tuned GPT-2 models for each level of stories.

All models are uploaded to the following link: https://github.com/liuf9/CS229_Project

Results

Although there are many quantitative scoring systems such as Bleu and Rouge for evaluating the result quality (especially meanings), we find they are not quite applicable for our situation. Therefore, we followed recommendations from Danial et al [4] to use human judgement to evaluate the result quality. We trained our models and generated samples every certain step/epoch. We stopped running the model when the model is generating results that approach the style of the trained stories. In Table 3, we summarized the final loss score at the time model stopped.

Table 2. Final Loss of the Three Models

	Level 0	Level 1	Level 2
GPT-2	1.40	1.22	1.04
RNN style transfer	0.55	0.47	0.45
RNN text generation	0.51	0.32	0.22

We then use the trained model to generate unconditioned texts for different levels. A sample of one of the stories used to train is shown on the first row.

Table 3. Examples of the unconditioned output from three models

	Level 0	Level 2
Sample original stories	In the evening, Ben sits on the fire escape and listens to the music from the Zig Zag Jazz Club. He joins in, playing his trumpet. Sometimes he plays until very late and falls asleep in the hot night air. Every day on the way home from school, Ben stops by the Zig Zag Jazz Club. He watches the musicians practice.	I like my red car. I like my blue car. I like my yellow car. I like my pink car. I like my purple car. I like my green car. I like my green car too.
GPT-2	The Witch Doctor lives in England. He and his family drink milk and talk to cows. It doesn't seem to bother them that witches live in the woods. They talk to bears, wolves, pigs,	I like my lawn mower. I like my faucet. I like my garbage can.

	sheep, and even bats. The animals are happy. One of the cows tells her story to The Doctor who lives in England. He and his family live in England. But the forest is a lot worse than that.	I like my car seat. I like my garage. I like my tree. But I LOVED YOU.
RNN style transfer	It was so Knuckles so so so fade , and then I 'll turn with which they had can't see. I did not cool trick with all , and they won't be able to eat their pirate or wings on.	Worm, worm, wiggle, squirm. So Max fish a dance. So Max fish and dance. Worm, worm, wiggle, squirm. So Max a dance. Worm, go home.
RNN text generation	When you lift the handkerchief , he was heartbroken. Charlie and Mouse saw Tess and Lottie followed Jack and the apple 's plate.	Here is an ear. Here is an eye. I see a cat. I see a dog. A bug in a rug. A bug in the grass.

We also did another experiment to generate conditional context by GPT-2 models fine tuned for each level from the same prompt story.

Table 4. Rewrite stories by fine tuned GPT-2 model for each story level

Story used as prompt	I like my red car. I like my blue car. I like my yellow car. I like my pink car. I like my purple car. I like my green car. I like my green car too. I like my black car. I like my white car. I like my car.
Model generated story: Level 0	It's time for my father's car. It takes a lot of money. But he always comes by my house. It takes a lot of people buying cars. And buying cars very, very fast. And doing it in little and little bursts of speed. It takes some crazy people. But there are a lot of crazy people. And I mean crazy people. I like motorcycles. I like cars. I like clothes. But I also like clothes. I like cars. And motorcycles. And clothes. It's not all bad. I like bugs. I like fish. I like flowers. I like bugs. But I also like bugs. Lots of bugs! A lot of bugs! And bugs are heavy. Really heavy. For a bicycle. For a car. For a big car. For a whole car. Oh my goodness. It's heavy. Really heavy. For a car. It's almost as heavy as the ground I'm in. Yes, I know the way around the house. The way the birds call it. High above. High above. Up above. Up above. Oh my goodness.
Model generated story: Level 1	I think I like my red from front to back. I think I like my blue from front to back. I think I like my pink from front to back. I think I like my purple from my old car. I think I like my purple from my new car. I think I like my red from my bike. I think I like my blue from my bike. I think I like my pink from my old bike. I think I like my pink from my ice car. I think I like my pink from my bike.
Model generated story: Level 2	I like my dark green earrings. I like my lavender earrings. I like my blue earrings. I like my pink earrings. I like my blue earrings and I think I like them pink.

For the results in each level, we generated about 10 samples and selected the one that we could observe similar writing patterns that compare to the original story. The generated results have shown obvious features that fit their corresponding levels utilizing the leveling criteria we defined. However, we observed obvious grammar issues that would hinder comprehension for lower levels (complex sentence structure).

Discussion

We observed the RNN style transfer model did not fully transfer the positive style into the negative style. Fortunately, most of the transfer stories are readable and the output stories have more negative words than before. The reason for why this model didn't truly transfer style may be that the source and target vocabulary we used in the model is not enough. Second, our training and development data are too small to train the model well. Third, the input stories are not 100% neutral and positive style. Therefore, this may lead the model to be less accurate. Besides, the training time in this model needs about 12 hours per level, which is the most expensive one among the three models.

RNN text generation model is expected to predict the following characters with the right level. However, since this is a character-based model, the predicted word sometimes is not a meaningful word. Compared with level 0 to level 2 output, this model performs better at a more easier level which is level 2 in our project. The reason can be that the character-based model performs better at short words. The training time for this model is less than an hour per level. Consider both the accuracy and the efficiency, this model is a better choice than the style transfer one.

The GPT-2 model has shown better performance for simpler story structures. However, for more complicated story structures, we observed the model tends to generate stories that are hard to comprehend. We identify this issue partially coming from the lack of training samples and also we need to spend more time to fine tune the model to obtain a more comprehensive model. A more robust leveling framework regarding paragraph structure, word selection and sentence structure need to be developed to better describe the features for each level.

Conclusion

With fine tune of the three models, they all successfully rewrite or generate the story within its original level. But the meaning of the stories and their accuracy are not performed well. In the future, we can use the RNN translation model to translate datasets into different languages and then translate back to have data augmentation. Second, after we have more datasets, we can generate more suitable dictionaries for the RNN style transfer model. Third, we can add more layers to the RNN text generation model to let the model learn better especially for the hard and long words. Furthermore, obtaining more data for fine tuning GPT-2 models is critical for the model to generate higher quality results. Last but not least, a more robust and quantitative result evaluation framework is needed to define what is a high quality result.

Contributions & Acknowledgements

Feiyang Liu preprocesses the data and splits the dataset into 3 levels and further splits and readjusts the format of 3 datasets with potential auxiliaries such as word dictionaries for training and testing.
Bochen Zhang fine-tunes the GPT-2 model and modifies the model to fit leveling.
Yiqun Tian finds the usable RNN models and fine-tunes them to get reasonable results. Three of us discussed the framework of the project and wrote the report together.
We would like to thank Catalin Voss who generously provided us with the story dataset. However, as the books are unpublished, the provided files that largely reveal the story contents are removed from the git repository.
The github repositories of the models we use are listed below:

1. text_generation model is from: https://www.tensorflow.org/text/tutorials/text_generation#build_the_model
2. text_style_transfer model is from: https://github.com/wyu-du/text-style-transfer
3. The original GPT-2 model is from: https://github.com/nshepperd/gpt-2

Reference

[1] Li, Juncen, et al. "Delete, retrieve, generate: A simple approach to sentiment and style transfer." arXiv preprint arXiv:1804.06437 (2018).
[2] Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.
[3] Vaswani, Ashish, et al. "Attention is all you need." arXiv preprint arXiv:1706.03762 (2017).
[4] Ziegler, Daniel M., et al. "Fine-tuning language models from human preferences." arXiv preprint arXiv:1909.08593 (2019).
[5]Jin, D., Jin, Z., Hu, Z., Vechtomova, O., & Mihalcea, R. (2020). Deep Learning for Text Style Transfer: A Survey. arXiv preprint arXiv:2011.00416.