Towards "Leveled" Story Generation: Project Final Report

Yevgeni Chuvyrov
chuvyrov@stanford.edu

June 2021

1 Introduction

Per Dr. Catalin Voss' proposal for AI class project, "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're often boring, leaving older kids who fell behind unmotivated. The goal of this project would be to build a language model that can re-write a children's book story for *any* reading level."

2 Literature Review

I focused my research on four different areas: (1) understanding current non-AI based approaches to writing stories targeted at specific reading level, (2) methods for assessing the readability level of a given text passage, (3) public data available for training and modeling, as well as efforts of working with that data, and, finally, (4) modern methods for generating text.

The (human) process for writing stories targeted at specific reading level is well understood and documented. In the "Children's Writers Word Book," the authors provide extensive lists of words appropriate for different reading levels, from Kindergarten through the sixth grade. For example, the kindergarten student's vocabulary centers around one-syllable words under six letters. Examples of Kindergarten-specific words that start with the letter "a" are "ant, any, are, arm, art." In that book, there is also a large number of tips for human writers writing kids stories. One of the suggestions, for instance, centers around character's ages. "The protagonist(s) of your story should be the same age as or a little older than the reader. Establish the age of the protagonist as soon as possible, and make sure your characters act appropriately for their ages." Finally, the book also provides a list of thesaurus, all appropriately marked with the reading levels, for a given word. These word lists proved to be valuable for both establishing the baseline for this project.

The research on estimating readability of text is ample. There's even a Kaggle competition with a $60,000 prize to identify the appropriate reading level of a passage of text (https://www.kaggle.com/c/commonlitreadabilityprize/overview/timeline). While the traditional feature formulas, like the Flesch formula for assessing the readability of text, relied on linear models, today's state of the art systems rely on deep learning methods incorporating attention mechanisms, per "Linguistic Features for Readability Assessment" by Deutsch et al.

Per Microsoft Research paper "MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text," Microsoft has crowd sourced the creation of 500 short children stories and released it for use by the research community (https://github.com/mcobzarenco/mctest). According to that paper, all stories should be accessible to the seventh grade students. Researchers describe some of the earliest text understanding efforts, namely DARPA introducing the Airline Travel Information System (ATIS) in the early 90's: the task was to slot-fill flight-related information by modeling the intent of spoken language. Facebook Research group used a similar approach of using multiple choice questions to predict missing words and test text understanding in "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representation." As part of that work, Facebook researchers created Children's Book Test Dataset (CBT), which contains published kids stories, for which copyright has exprired. These stories (and many more) are now also available for free on gutenberg.org.

Finally, text generation is a very active and fast moving area of research. Traditional, joint probability-based Language Models (Bengio et al) gave way to Recurrent Neural Networks, which are in turn being replaced by more robust Transformer (Giacaglia) and Generative Adversarial Networks-based methods. Models based on transformers and trained on large text corpus, such as GPT-2 (Radford et al) now achieve state of the art results in a zero-shot setting. Additionally, another state-of-the-art framework, Generative Adversarial Networks (GANs), is also an active area of text generation research (Iqbal and Qureshi). Most recent efforts have also focused on combining Transformer architectures with GANs (Jiang et al).

3 Dataset

Children's Book Test Dataset from the Goldilocks study by Facebook Research was used for this project. This dataset contains such classics as Louis Carrol's "Alice's Adventures in Wonderland" and Rudyard Kipling's "The Jungle Book." In total, it's made up of 108 full-text books, represented as one sentence per line. The dataset has been pre-processed with all the images and header info removed and the sentences available in plain text ready for processing. Here's an example title and a first sentence from "Alice in Wonderland:"

_BOOK_TITLE_ : Lewis_Carroll___Alice's_Adventures_in_Wonderland.txt.out
CHAPTER I. Down the Rabbit-Hole Alice was beginning to get very tired of

sitting by her sister on the bank , and of having nothing to do : once or twice she had peeped into the book her sister was reading , but it had no pictures or conversations in it , ‘ and what is the use of a book , ’ thought Alice ‘ without pictures or conversation ? ’

-

Another option could have been to use the MCTest dataset with its 500 short stories for kids. However, the target reading level for all stories is the same (7th grade) and all of the stories in that dataset are brand new and not generally familiar.

4 Baseline

The baseline has been implemented by simply removing age-inappropriate words from sentences in the dataset, or, more accurately, by keeping only age-appropriate words in sentences. A set of words provided in "Children's Writers Word Book" appropriate for a selected reading level has been used to determine "age-appropriate set" of words. All named entities (identified by capitalization) are left intact. All sentences are stemmed and lemmatized (using Python nltk package) before checking against the list of "appropriate words."

5 Main approach

The general approach to solving the problem consisted of three distinct steps: (1) using the readability metric, get the training data appropriate for the desired reading level/grade, (2) select and train a language model on data obtained in (1), and, finally, (3) generate text using model created in (2). Each of these steps is described in detail below.

To get reading-level appropriate data, I used Flesch-Kincaid reading grade metric to estimate reading level of each sentence in the CBT dataset and kept only the sentences that met the required reading level criteria. To better fit the language modeling approach I selected (described below), I limited training data to a single book that I wanted to "re-write" using age-appropriate language. Surprisingly, some books, such as Alice in Wonderland, contained very few sentences that matched the Kindergarten level readability score (one of the levels I used for my evaluation); however, there were enough books with sufficient level-appropriate data for my training.

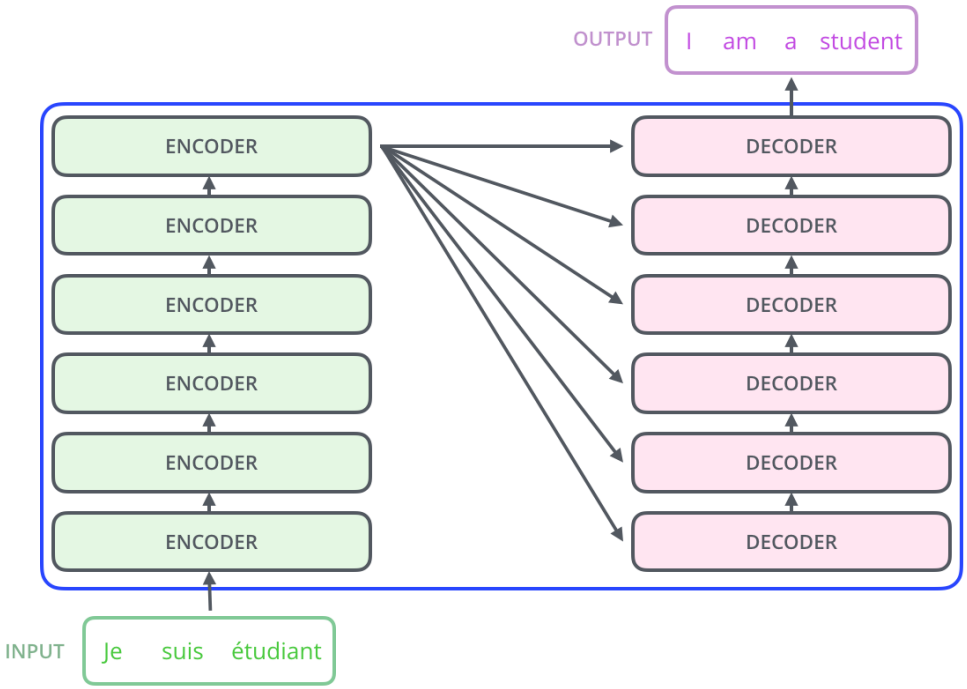
I considered three approaches to training a language model. The first one was a statistical model of language that could be represented by the conditional probability of the next word given the previous, as summarized by the equation below (Bengio et al):

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t|w_{t-1})$$

While this could still be a valid approach, the literature consulted during my research pointed to limitations of this approach: it can only make inference on things it has been trained on. This approach has also been shown inferior to a deep neural networks-based approaches (LSTM, RNN, CNN) and hence was not evaluated as part of this project.

The second approach considered was to use Generative Adversarial Networks (GANs) for text generation. However, while this state-of-the-art framework excels at image generation, it finds text generation much more challenging. This is due to the non-differentiable nature of discrete symbols, per Iqbal and S. Qureshi in "The survey: Text generation models in deep learning." Text generation process is discrete, which makes output error hard to back-propagate to the generator in GANs. The way GANs can be used for better text generative models is an active area of research and a number of approaches for working through the sparsity of this binary guided signal in GANs has been suggested. One such approach, LeakGAN, combines feature matching and hierarchical reinforcement learning, and had the best documented text generation performance out of a number of approaches. From the "Long Text Generation via Adversarial Training with Leaked Information," the setup "leaks a goal embedding vector during the generation process to guide the generator on how to get improved." I have tried to deploy a LeakGAN-based model, but was throttled by the complexity of setup, as well as a number of open GitHub issues on the LeakGAN repo, stating that the results were not reproducible, or at least not easily reproducible. Nevertheless, GANs and, in particular, LeakGAN-based implementations for "leveling" kids books could still prove to be very effective.

The third approach, and the one that I adopted for training my language model, is based on the Transformers architecture. Transformers is a popular, deep learning-based framework that excels at sequence transduction, or transforming input sequences into output sequences. The basic components of a transformer are shown in the diagram below, courtesy of Giacaglia. Internally, the Transformer consists of six encoders and six decoders, as shown in the diagram.



The attractiveness of Transformers-based approach is the vast ecosystem built on top of the framework (BERT, GPT), as well as impressive results delivered by the projects that adopt Transformers-based architecture. For this project, I considered two transformer architectures for auto-regressive language modeling: (1) training a Transformer-XL model from scratch on my corpus, and (2) fine-tuning an existing GPT-2 checkpoint on a reading level-specific training data. Since Peric et al obtained better results with fine-tuning GPT-2 model with custom data for legal text generation, I decided to adopt their approach and leveraged a Transformer-based GPT-2 model from OpenAI, fine-tuned with my custom data.

Finally, the easy part was generating text after training the fine-tuned GPT-2 model. There are multiple methods available for tuning the most relevant text sequences (and there's still plenty of room for experimentation in that area). I chose the greedy way of picking what the model thought was the most relevant result out of multiple suggestions.

6 Evaluation Metric

Quantitative Metrics: Readability and Grammar

"Leveled" generated text is evaluated for readability using the same approach as was used for collecting training data. Namely, Flesch–Kincaid Grade Readability score (provided as part of Python textstat package) will be used to quantitatively determine the score of each generated sentence. That score is averaged

over the full generated passage to give an average readability score.

It would be also desirable to have a similar qualitative metric for the grammar of each generated sentence. Since grammar can get quite complex, however, it doesn't appear that such metric exists, although it could likely be created by hand by adopting a (potentially large) set of predefined grammar rules.

Qualitative Metrics: Sequitur/Non Sequitur Flow
The hardest part of generating what is essentially a new story is preserving the theme and overall intent of the original story. Do sentences follow each other in a logical manner? Do they make sense together? Does the overall theme hold? Those are the questions that at the moment can only be answered qualitatively by a human.

7 Results and Analysis

To deploy and test the Transformer-based solution described in the Main Approach, I have provisioned a Data Science Virtual Machine in Microsoft Azure with NVIDIA Tesla K80 GPU attached. (Side note: I tried using Google Cloud Deep Learning VM first, but did not have luck with it - it deployed without the libraries needed and without CUDA drivers installed). I then created a Jupyter notebook for preparing training data, training and generating text (all code for that notebook is at the GitHub link in the code section).
After uploading CBTest dataset, I created two training datasets: (1) for Reading Level 3 and (2) for Reading Level 13, both for a single book by Lucy Maud Montgomery "Short Stories, 1909 to 1922." The reason for this specific book was because it contained a large number of sentences with reading level 3. The reason for only a single book is due to training large amounts of data taking significant time. For example, this single book trained with sentences meeting reading level 13 and below took 4.5 hours on the Azure Deep Learning machine I created (this could be partially resolved by a more powerful machine and more powerful GPU, both of which are expensive). After augmenting a GPT-2 Transformer with my custom training dataset(s), I was able to get the following results for the first 3 sentences of the corpus.

Original Text
A Golden Wedding The land dropped abruptly down from the gate , and a thick , shrubby growth of young apple orchard almost hid the little weather-grey house from the road . This was why the young man who opened the sagging gate could not see that it was boarded up , and did not cease his cheerful whistling until he had pressed through the crowding trees and found himself almost on the sunken stone doorstep over which in olden days honeysuckle had been wont to arch . Now only a few straggling , uncared-for vines clung forlornly to the shingles , and the windows were , as has been said , all boarded up .

Baseline Text (all non-age appropriate words removed), Reading Level Grade 0 (Kindergarten)
A Golden Wedding The land dropped down from the and a of almost the little house from the road. This was why the man who opened the could not see that it was up and not his he had the trees and found almost on the in days had been to. Now only a few to the and the were as has been said all up

Generated Text, Reading Level Grade 0 (Kindergarten)
A Golden Wedding dress was on her mind. This was why I kept coming back. Now only a lazy bone in the body.

Generated Text, Reading Level Grade 13 (High School)
A Golden Wedding The land dropped abruptly down from the gate, and a thick, shrubby growth of young apple orchard almost hid the little weather-grey house from the road. This was why I had no correspond with you. Now only a little; now a good for a mother!

Reading Level Assessment: Original and Generated Text

Text	Reading Level (Flesch Kincaid Grade)
Original	9.74
Baseline	3.52
Generated, Level 0	2.86
Generated, Level 13	5.58

Quantitatively, based on reading level results alone, things appear that they work, i.e. we see that the reading level desired is the one we get with the generated text. And, when read individually, generated sentences are a lot more sound grammatically than the baseline (which is gibberish-like at the K level with many words removed). It is also noteworthy that our model chose to keep the first sentence in its entirety when asked to generate text for the high school reading level, but not when asked to generate text for kindergartners. The sentences generated for the K level are short, simple, and use words that kindergarten-level kids should be familiar with.
However, when taken altogether, the sentences do not follow each other logically, nor do they appear to weave a story intended in the original text.

8 Error Analysis

The GPT-2 based Transformer model was able to generate grammatically correct, reading level appropriate sentences that did not follow original story line nor did the sentences follow each other logically (non sequitur). This is likely by (incomplete) design: the generation is currently one sentence at a time, and the generated sentence is seeded with a first few words of the original sentence, then tokenized, transformed, and completed by the model. Switching to paragraph-by-paragraph based generation is likely to yield better results.

However, it is not obvious how to determine the paragraph structure in the CBTest dataset. The text file includes sentences, one line at a time, without paragraph separators.
Additionally, the model lacks the notion of the overall "theme" of both the complete book, as well as individual paragraphs/chapters. Without this knowledge, it is likely impossible to end up with a reading-level appropriate text that follows the original story line (however loosely). I discuss several possible way to overcome this in the Future Work section.

9 Future Work

The first step that should improve results is to get better data. There are several issues with CBTest dataset, including the fact that the books are a hundred years old and many language constructs, as well as the world around us, have evolved significantly. Additionally, lots of special characters and separators still appear in the text, important constructs (paragraphs) are not preserved and many sentences are split along several lines, making these parts appear to our model as separate sentences.

Another important step could be experimentation with injecting a list of topics, perhaps on a paragraph-by-paragraph basis, into the generation process. One potential way to approach this could be via beam search of sentences generated by the transformer and evaluating how much each sequence contributes towards the main topics of the paragraph.

Finally, it's worth looking at other ways of generating text besides Transformers. Basic probabilistic language models, trained on age-appropriate text corpus, could yield fast, reliable results. Additionally, evaluating GANs, by themselves or in conjunction with Transformers via something like TransGAN (Jiang et al).

10 References

1. Alijandra Magliner and Toyopa Magliner, "Children's Writers Word Book"
2. Matthew Richardson, Christopher J.C. Burges, Erin Renshaw "MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text."
3. Felix Hill, Antoine Bordes, Sumit Chopra, Jason Weston "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations"
4. T. Iqbal and S. Qureshi, The survey: Text generation models in deep learning, Journal of King Saud University – Computer andInformation Sciences,https://doi.org/10.1016/j.jksuci.2020.04.001
5. Tovly Deutsch, Masoud Jasbi, Stuart Shieber "Linguistic Features for Readability Assessment"
6. Lazar PERICa, Stefan MIJICa, Dominik STAMMBACH and Elliott ASH "Legal Language Modeling with Transformers" http://ceur-ws.org/Vol-2764/paper2.pdf
7. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya

Sutskever "Language Models are Unsupervised Multitask Learners" https://d4mucfpksywv.cloudfront.net/bette
language-models/language-models.pdf
8. Giuloano Giacaglia "How Transformers Work: The Neural Network used by
Open AI and DeepMind" https://towardsdatascience.com/transformers-141e32e69591
9. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin "A Neu-
ral Probabilistic Language Model" https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf
10. Yifan Jiang, Shiyu Chang, Zhangyang Wang "TransGAN: Two Transform-
ers Can Make One Strong GAN" https://arxiv.org/abs/2102.07074