18.065 Matrix Methods in Data Analysis, Signal Processing, and Machine Learning
Project Report

Can we foresee the story ending?

Exploring deep-learning models to predict how stories will end

May 9, 2022

Aijia Yao
aijia@mit.edu

1 Abstract

This paper focuses on the text generation task for predicting story endings, using deep-learning models with Natural Language Processing(**NLP**) techniques. Firstly, sampling strategies are selected for designed models by analysing the relevance between the text tokens. Then, the training processes are demonstrated with **Seq2Seq LSTM** and transformer-based **GPT-2** models on different stories, including 2 book series and more than 10 books. Finally, the result shows that the predicted endings can vary between genre, series and training epoch numbers. Meanwhile, character-wise sampling is used as an ideal strategy to solve double descent problem in Seq2Seq model training.

2 Introduction

From the moment people created language to communicate, stories were born and told. In these stories, the authors share their own life experiences or sparkling imaginations offering a feast for the readers. The hints and clues woven into the plot and details that leads to the final ending often give readers a wonderful surprise. Or sometimes, the ending itself is a surprise somehow came out of the blue. So there is always a question in our mind before we turn to the last page of the book, 'How is it going to end?'.

Nowadays, these stories can not only be read by people, Natural Language Processing (NLP) has also made 'reading' text possible for machines. And this so-called 'reading' is actually to tokenize the text into data groups and then perform analysis to understand the relationship between tokens. [1] In this way, a text-generation task can be done smoothly with predictions made from the data. Utilizing data to simulate the way that we gradually unveil the next chapters following the author's guidance, guessing what is the end.

Nevertheless, although various kinds of text-generation methods have been proposed along with deep-learning models such as Seq2Seq model using LSTM[2] and more advanced transformers with attention mechanism[3], their best performance lies in the non-stop predictions instead of completing the story and reaching an end. Therefore, to realize a specific generation targeting better story endings, further analysis is required.

Similar works have been made to generate a story ending by designing the encoding with multi-source attention[4]. However, the complexity of these models can lead to lack of interoperability. Hence, in this paper, flexible sampling strategies and appropriate model selection are mainly focused, providing insights by answering the following questions:

- How to evaluate a good ending? And how to realize that through data analysis?
- What kind of tokens and models works the best for producing story ending?
- After taking text to data, how to optimize the training process and the result with appropriate model selection and design?
- How does the text structure and the sampling strategy influence the results?

3 Text and data

3.1 Good story endings need good decoders

Besides basic requirements for coherence and logic. A good story ending should complete the story or extend it for more possibilities. From the writer's perspective, the intentional design of suspense should be unravelled. From the reader's perspective, there could always be expecting some surprise in the final scene. To achieve both, a good ending should bring the subtext into text with creative or unexpected surprises. This leads to two important sampling techniques in decoding layers used in the model:

- **Temperature Sampling**

$$P(y_t = w_i|y_{<t}, x) = \frac{exp(z_{t,i}/T)}{\sum_{j=1}^{|V|} exp(z_{t,j}/T)} \tag{1}$$

As the temperature T increases from 0 to 1, the normalized probability distribution would be less uniform, which gives rise to generated text being random and somehow creative. In comparison, it can be seen as a modified version of the softmax layer in expression(2).

$$\sigma(z_{t,i}) = P(y_t = w_i|y_{<t}, x) = \frac{exp(z_{t,i})}{\sum_{j=1}^{|V|} exp(z_{t,j})} \tag{2}$$

With the same logits $\mathbf{z_t}$ (see in expression(4)) and vocabulary cardinality $|V|$, temperature sampling provides a more flexible activation function than the softmax.

- **Top-k Sampling**

$$P'(y_t = w_i|y_{t-1}, x) = \frac{P(y_t = w_i|y_{t-1}, x)}{\sum_{x \in V^{(k)}} P(y_t = w_i|y_{t-1}, x))} \tag{3}$$

In equation(3), $V^{(k)} \subset V$ is the top-k vocabulary set when its size k maximize the denominator $\sum_{x \in V^{(k)}} P(y_t = w_i|y_{t-}, x))$. So every time step, it would only take the k-most possible tokens into the sample and re-scale the distribution with the latest tokens. This can both serve as a noise filter and logic enhancement for the model training. Selecting an appropriate k is then the degree of freedom to tune for an ideal story ending text.

3.2 Open endings

For stories in series or having a sequel, the ending is perhaps open-ended or even serves as the beginning of the next one. In this sense, the prediction of an 'ending' text should not only driven by the possible 'trend' of the original text, but also introduce random possibilities. However, this is tricky and greatly relies on the author's own style of narration.

To investigate the generation potential for both kinds of the endings, a list of books written in English were chosen as in the following:

Book List			
Title	Author	Genre	Stories
Harry Potter Series	J.K. Rolling	Fiction/Fantasy	7
Sherlock Holmes Series	Arthur Conan Doyle	Fiction/Detective	12
The Hound of the Baskervilles	Arthur Conan Doyle	Fiction/Detective	1
Grimms' Fairy Tales	Grimms brother	Fairy tales	62+
The Happy Prince	Oscar Wild	Fairy tales	5
Alice in Wonderland	Lewis Carroll	Fantasy	1
The Wonderful Wizard of Oz	L. Frank Baum	Fantasy	1
The Great Gatsby	F. Scott Fitzgerald	Fiction	1
Total story count:			90+

Since the training on each book is very time-consuming with neural network, this selection is made based on some trail trainings which took the computer CPU for an average of 2 hours per book. Accordingly, a minimal number of books were chosen to permit comparison between endings in different genres and story length and types.

3.3 Data preparation

The source of the books is the latest edition on Project Gutenberg(`https://www.gutenberg.org/`). Except for the *Grimms' Fairy Tales*, which happens to have two versions with different stories, making it a natural split for a training set and a testing set. The ending text and tests for two editions of *Grimms' Fairy Tales* are organized into one spreadsheet which can be found in my github repository here.

4 Methodology

4.1 Training Models

4.1.1 Seq2Seq LSTM

The Long Short-Term Memory (LSTM) network itself has a wide application in making predictions from a series or a sequence of data[5], with each unit generating logits $\mathbf{z_t}$ and states from the input x_t according to the equation(4), where $\sigma_g(\cdot)$ is a sigmoid function.

Based on this approach, a Seq2Seq learning model can be constructed with multilayered LSTM, which is originally applied in language translation. Taking the natural paragraphs of the text as sequences, the development of the story can be viewed as time-series data. Feeding the text before the ending scene into the network, a Seq2Seq model can be trained to predict the ending derived from the 'historical' content. As a straight-forward prediction, this Seq2Seq generation model is implemented in Matlab as the baseline model.

$$
\begin{aligned}
z_t &= \sigma_g(W_z \cdot [h_{t-1}, x_t]) \\
f_t &= \sigma_g(W_r \cdot [h_{t-1}, x_t]) \\
\tilde{h_t} &= \tanh(W \cdot [r * h_{t-1}, x_t]) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h_t}
\end{aligned} \tag{4}
$$

It is also worth mentioning that the input $\mathbf{x} = (x_0, x_1, \cdots, x_t \cdots)$ in the expression(4) is generated by the key unit called the wordEmbeddingLayer inside the network, which takes text sequence to vectors. This text vectorization is done simply by creating dictionary indices for each unique vocabulary and map the indices for each sequence to form a vector. Which is different in the next GPT-2 model with character-wise encoding (each word can have a specific tensor). This is one reason why the coherence of the predicted text can be improved.

4.1.2 GPT-2 with prompt

As shown below in Figure 1, to achieve a better generated story ending, a transformer-based GPT-2 model is implemented. This model not only includes the sampling strategies discussed in the early section, but also offers an extra prompt input that can be used to control the endings generated.

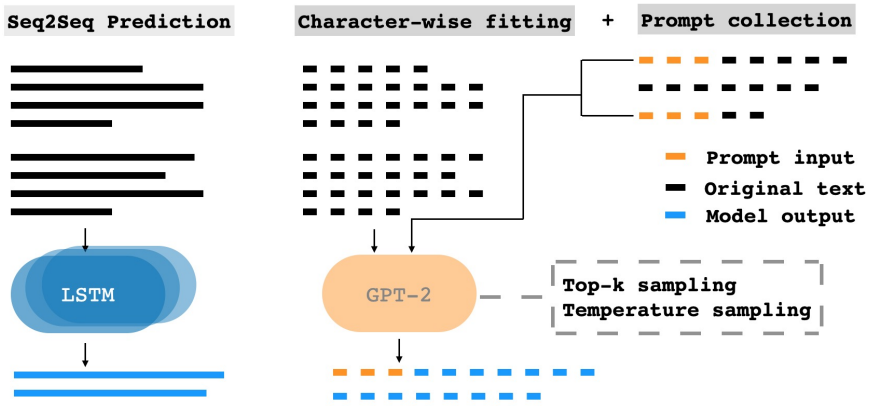

Figure 1: Left:The Seq2Seq LSTM model shown is used as a baseline;
Right: A prompt-driven ending prediction model implemented in this project.

In particular, the prompt words such as 'In the end', 'Then', and 'So' are collected from the training set of *Grimms' Fairy Tales*. Forced by these prompts, the generated text can 'converge' to certain predictable endings without deviating from the context to a completely new story or meaningless open endings.

4.2 Why GPT-2

There are two crucial implemented mechanisms in GPT-2 and other transformer-based network that especially fit this generation task.

The first one is the word tensor instead of the vector from the sequence. Although they are both data vectors from the text in essence, the sequence vector is fixed in length and be fed into the network group-by-group while the tensor is formed from character-wise sampling, which has a greater flexibility in its length and is easy to implement prompt inputs. Also, character-wise sampling can also provide potentially new words generated that is not entirely identical with the original context, going beyond the vocabulary!

And more importantly, transformers like GPT-2 also come with attention mechanism with the multi-head attention layer. The layer performs an equation each time as below.

$$Attention(Q, K, V) = \sigma(\frac{QK^T}{\sqrt{d_k}})V = SV \tag{5}$$

The matrices Q, K, V each stands for query, key and value embedding inputs calculated by a weighted dot product of the input word tensor, and the weight for query, key and value are trained by the network. So, instead of copying the original text word-by-word, attention mechanism can assign different weights to each part of the text input, leading to less repetitive and potentially more creative story endings.

4.3 Training Parameters

Before throwing the data into the models, there are still some important parameters need to be explained and formulated.

- Learning rate decay function

  $$lr_t = lr_0 \frac{1}{1 + lr_{t-1} * iteration * decay_t} \tag{6}$$

  The initial learning rate $lr_0 = 6e - 4$ would be set to a decayed lr_t for each iteration step. In neural network with large iteration numbers, this is a useful method to improve training results[6]. This function is set so that not all the context before the ending is equivalently weighted, which is followed from an assumption that the ending should not be highly dependent on the beginning or other far-reached context.

- Adam optimizer

  To achieve more ideal results with an efficient training process, Adaptive Moment Estimation(Adam) [7], is chosen to perform objective function minimization.

  Although it is generally a better optimizer for text generation tasks, for better outcomes, a dense data pattern should be guaranteed. There is a simple explanation about this in the discussion section when tackling the double descent problem.

5 Testing and Experiments

Seq2Seq LSTM training in Matlab	GPT-2 training on Google colab
CPU: 2.7 GHz Intel Core i5	GPU: 1190 MHz Tesla P100-PCIE
1.5 hours/100 epochs	1 hour/100 epochs

With Matlab's deep-learning tool box, it can be easily tuned to run the baseline test for different stories and books. This provides a general prediction made by the context before the ending. Then the results of the Seq2Seq LSTM models would be used as an reference for better improvements or worse results in the prompt input driven GPT-2 text generation.

Ideally, the *Sherlock Holmes* and *Harry Potter* series would be compared to decide whether the genre of series will effect the ending generation. At the same time, the *Sherlock Holmes* series is also compared with the book written by the same author to investigate whether the model can tell series from a solo story.

And finally, since the fairy tales aiming for young readers have much simpler endings, the prompt words can be collected and used to see how the endings may vary with the prompts. For making the comparison between the best outputs, the sampling strategies will be implemented and tuned in the GPT-2 model for the best possible result for each prompt.

6 Results

6.1 Fantasy VS Reality

6.1.1 Series

In baseline tests, the Harry Potter series only have predictions that are irregular and less readable while double descent made the Sherlock Holmes series also in poor outcomes. But the transformer didn't show clear distinction on the different genres readability.

	HarryPotter	SherlockHolmes
Baseline	And the seconds wizards of studying gray and the same burning before they weren\'t grinned any of the way .	He was my word, " if you were some more purposes, sir, " said I.
Pompt	Then a dragon was sweeping about in his usual bad temper,	Then he sprang upon the tea table, and coffee and wine were in front of them.
Less epochs	So I think I've in troduced myself? Sir Nicholas de Mimsy-Porpington at your service.	So the baronet could probably swallow on the trembling hazy deged.

From the table, it shows that the the results of the prompt(highlighted in orange) GPT-2 would be pure replication from the original text when the epochs of the training is small, (the epochs number for the baseline is always 200) even with high temperature(0.9). Still, the common prompts such as 'Then' generate more side stories in Harry Potter series than the Sherlock Holmes. So the series books with more side stories and branches would make it harder to predict an ending for the main parts of the story. This is not so obvious in the Holmes's case, which also indicates that the generator doesn't tell the endings of a series or an independent story for reality fictions.

6.1.2 Independent stories

The fantasies are again facing the problem of unpredictable imaginary scenes. Comparing to the them, *the Great Gatsby* stands out as a reality fiction.

	The Great Gatsby's ending
Original	And one fine morning— So we beat on, boats against the current, borne back ceaselessly into the past.
Baseline	He stood the doorway, sounded well enough but shade his face there green .
100 Epochs	**Finally**, and a weather in love with her face bruised and breathing along in his hands.
200 Epochs	**Finally**, with reluctance. "I love Daisy's furious because the drug to put her under a story of white dress as I asked where I am.
300 Epochs	**Eventually,** he died in the vicinity. **In the end,** sparkled by a wire, and disappeared into the living room

The quality of the generated ending is generally stable with 200 trained epochs or above. The amazing parts that have new usage or new meanings of words are underlined as in the table above. Considering that there was no phrase as 'in the end' in the whole book, this may be a supporting evidence that new prompts outside of the story can inspire an unexpected new thing for the ending text.

6.2 Fairy Tales

The *Grimms' Fairy Tales*, as a collection of fairy tales from the same author, provides an ideal database of prompts and endings. The statistics are collected in the spread sheet in my github repository:`https://github.com/aijia01/18.065-Final-Project`. And in the following table is the test of the top four frequently used ending prompts in the Fairy Tales on the story called *The Star-Money*, which is not in the training set.

In this case, the 300 epochs trained has led to longer ending sentences while 200 epochs training has generated endings more concise and fairy-tale-alike. As these stories are made

to be told and convey simple messages, they are reasonable to keep the story not being too complicated to understand.

200 epochs	300 epochs
Then she gathered together the money into this, and was rich all the days of her life.	Then he was told that the king had never seen one of the room, but they were alone in the wood together
Then she grieved sorely at her misfortune, and said she would give him the princess;	Then she was sorry as the morning when he saw how his daughter behaved, and how she ill-treated all his guesses.
So they were married, and lived together very happy.	So the stairs was made to shee, and said to herself, 'The gold will soon be mine.'
And she was made to set the fresh of the sausages.	And thus she went roving on through the wide world, and looked about and flew into the fields

7 Conclusion

In general, the fantasy fictions and especially the fantasy series are not within the reach of this model to produce an ideal ending. But for the reality fictions and the simple fairy tales, the results can be readable and even surprising with appropriate prompts feed and suitable number of training epochs.

8 Discussion

In this project, the genre types investigated are limited, and there could be more variety if more books are trained. So it definitely leaves more space to explore other genres such as science fictions and other books to further improve the model.

Additionally,a double descent curve of the mean loss has occurred in the training of *Sherlock Holmes* series, which is not seen in other books or stories. To fix this, the text data were checked for each book. It turns out that the *Sherlock* series has on average 50 words per paragraph more than the others. Considering the great dependence of the tokens on the paragraph length and sentence length, a character-wise LSTM is used which leads to stable loss minimization. Still, the mechanism of double descent can be further studied.

Finally, the fact that this prompt-driven GPT-2 model can actually generating new words out of the original vocabulary base is surprising. The next stage is perhaps to study how to put more control over the new things generated. This is also an AI's pursuit of the writing creativity. But so far, it is evident that deep-learning is still one of the best ways to explore all the hidden truth. Story ends, but the learning will never stop!

References

[1] H. S. Christopher D. Manning, Prabhakar Raghavan, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] J. Guan, Y. Wang, and M. Huang, "Story ending generation with incremental encoding and commonsense knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6473–6480.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] K. You, M. Long, J. Wang, and M. I. Jordan, "How does learning rate decay help modern neural networks?" *arXiv preprint arXiv:1908.01878*, 2019.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] J. M. Nikhil Ketkar, *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, 2021.

[9] F. Chollet, *Deep Learning with Python, Second Edition*, 2021.