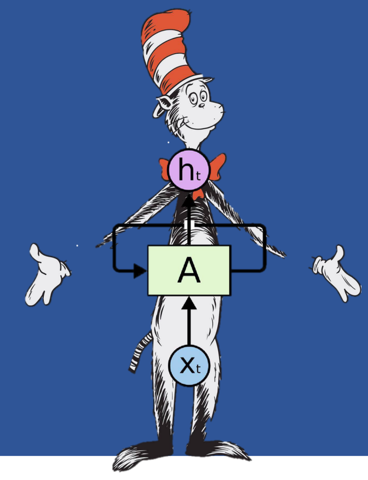
The RNN in the Hat: Generating a Dr. Seuss Picture Book



Allison Tielking[1], Michael Karr[1]

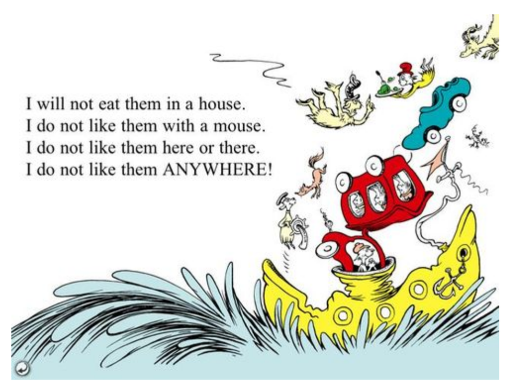
[1]Stanford University

Abstract

- Dr. Seuss is one of the most popular children's book authors of all time.
- Recent trends in artificial intelligence have attempted to generate works of art, scripts, and books given a particular art or writing style.
- Given the unique structure of a Dr. Seuss book, it is an interesting challenge to generate lines out of a Dr. Seuss book.
- **Goal: Implement a character-based Recurrent Neural Network that generates Seussian text.**

Introduction

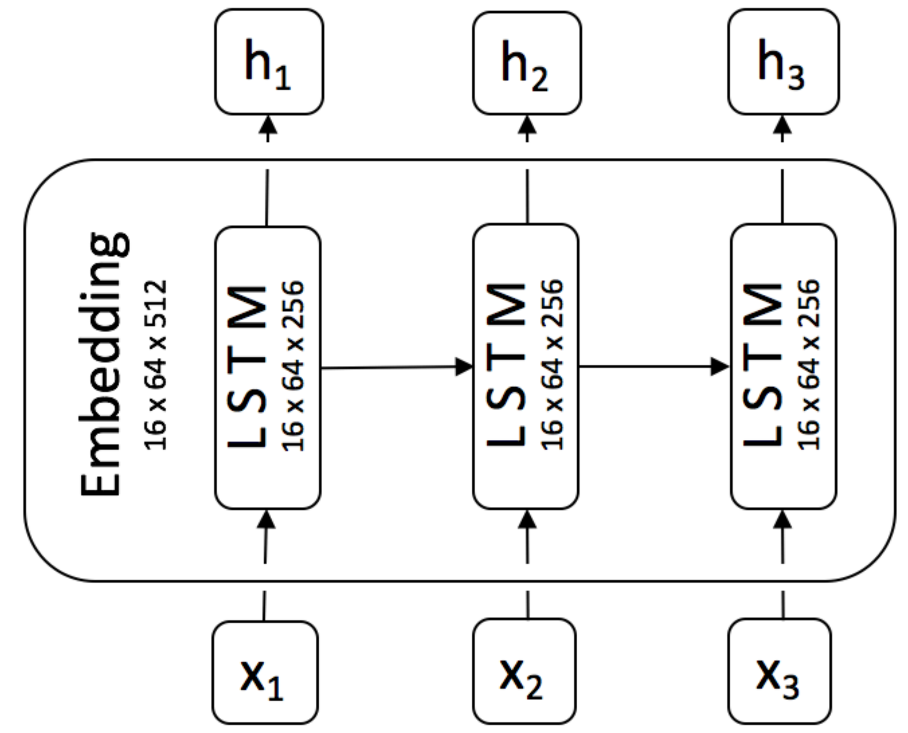
- Seussian rhyme schemes- rhyming pattern of last word in line
  - "Alternate rhyme": ABAB
  - "Couplet": AABBCCDD
  - "Seussian": XaXaXbXb



- Meter- the basic rhythm structure of a verse
  - Iambic trimeter
  - Iambic tetrameter
  - Anapestic tetrameter
  - Trochaic pentameter
- Stanza- grouped set of lines within a poem
  - Quatrains- 4 lines
  - Sonnet- 14 lines
- Sight words- high-frequency words
- Made-up words

Method

1. Dataset
   - Used a corpus of 7 most popular Dr. Seuss books
   - Created a corpus of works of other popular poets
2. Requirements
   - Each generated sample tested on classifier
   - Same dimension vocabulary size vector
3. Training Process
   - Implemented in Keras with TensorFlow backend
   - Trained for 100 epochs with batch size of 16 and sequence length of 64
   - Cross-entropy loss using Adam optimization
   - Built custom classifier for generated text using predefined "Seussian" criteria



Results

N-Grams

> … you did not know what to say. our mother like this? we don't know. and you may. try them and you may …

The N-grams model represented our attempt at a baseline, using N = 3 to generate words solely off of the pattern in the text provided.

Vanilla RNN

> … in took." I hook same, at is you fas. Saws fithet wise at thing OB.? At, to up they wild thin they way? Note fen, And Thow" …

Our Vanilla RNN was a bare-bones RNN in which we manipulated the weights and gradients manually, leading to suboptimal performance.

Keras RNN (Seq. Length = 64)

> … "Why do you like to go away. On the Grinch pump light on your shown to be went. And the magical things you can do with that ball …

We decided to convert our model to Keras to give us more freedom in analysis, also switching the optimization from Adagrad to Adam.

Keras RNN (Seq. Length = 16)

> … Mr. Knox. Now come to chew, sir. You're off to the fir. On their is heart or the small, Was singing! Without any putn. So shake …

The results too closely mirrored the input text, so we decremented the Sequence Length to get less accurate, but more creative results.

Keras RNN (**Multi-Corpus Training**)

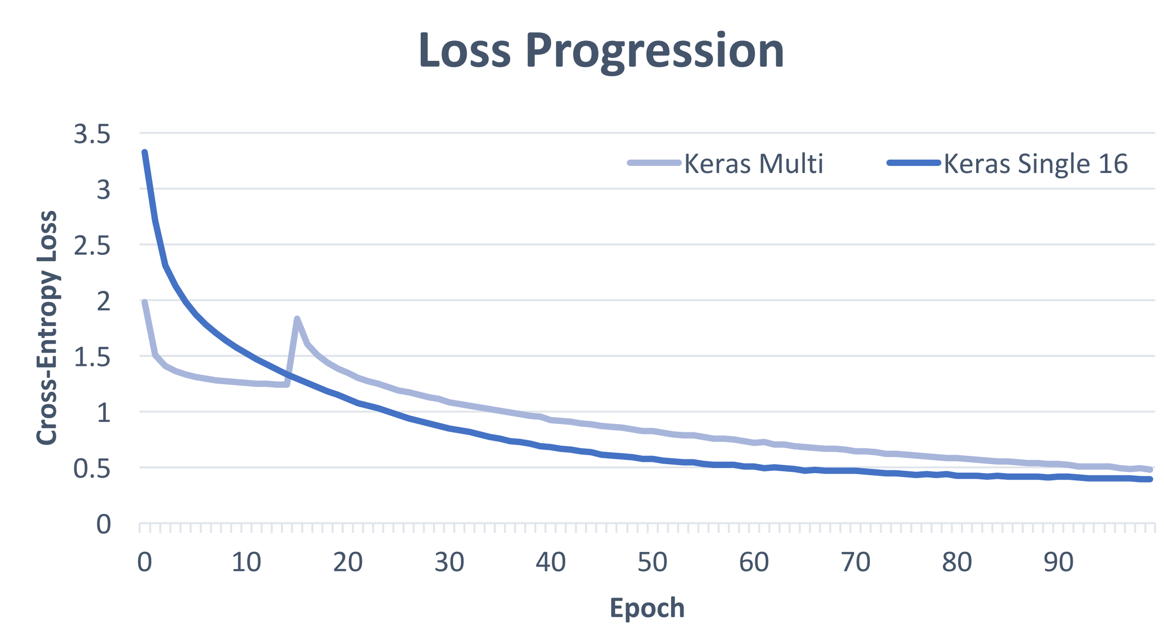
> **… "Then I say, "I must stop this whole tible, Around the father bags in a bottle, While these Things had everywhere. …**

To provide the RNN with a larger language context, we trained on first a poetry corpus and later on the Seuss corpus to emphasize Seuss's writing style.

Table 1. Accuracy, Loss, and Classification for Various Models

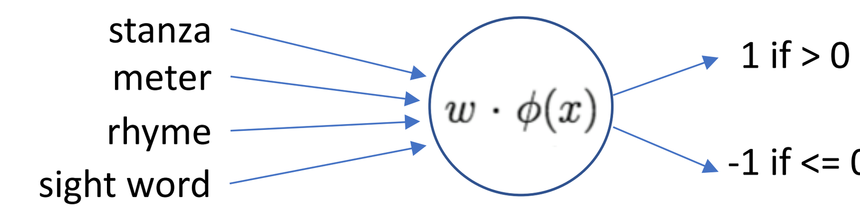
	Accuracy	Loss	Classification
N-Grams	N/A	N/A	**1** (1e-16)
Vanilla RNN	N/A	33.14	**-1** (-0.3)
Keras RNN (64)	0.96	0.13	**1** (1e-16)
Keras RNN (16)	0.88	0.39	**1** (0.5)
Keras RNN (Multi)	0.84	0.48	**1** (1e-16)

Loss Progression



Classification

- Use Poetry Tools Python library to estimate the stanza, rhyme scheme, and meter features of a given .txt file (classifies 1 if is of Seussian style).
  - Uses Levenshtein Distance to find closest estimates
- Calculate proportion of .txt file that are sight words (1 if is Seussian proportion)
- Initialize weights to [0] vector, then optimize using stochastic gradient descent
- Linear classifier: 1 (Seussian), -1 (Non-Seussian)



Conclusions

- Keras RNN with a sequence length of 16 yielded the strongest results according to the classifier
- Samples from Keras RNN classified as Seussian, whereas Vanilla RNN classified as non-Seussian
- Keras RNN samples lack a clear, consistent rhyme pattern, but some rhymes are present
- Achieve a more confident classification when sequence length is 16 rather than 64, only training on Seuss
- Decrementing the sequence length gave the samples less syntactically correct, but more "creative" results
  - This is due to a higher character volatility leading to less consistent word outputs
  - Mimics higher temperature results

Future Directions

- We would have liked to find the optimal training ratio between the standard poetry and Seuss texts in order to yield the most Seuss-like results
- Furthermore, we would like to add more robust poetry feature classification and generation mechanisms

Contact Information

Michael Karr
Stanford University
mkarr{at}stanford{dot}edu
Class of 2020

Allison Tielking
Stanford University
atielkin{at}stanford{dot}edu
Class of 2020

References

1. J. Foster and C. Mackie, Lexical Analysis of the Dr. Seuss Corpus, Concordia Working Papers in Applied Linguistics, 2013.
2. A. Graves, Generating Sequences With Recurrent Neural Networks, University of Toronto, 2014.
3. J. Brownlee, Text Generation With LSTM Recurrent Neural Networks in Python with Keras, Machine Learning Mastery, 2016.
4. I. Sutskever, J. Martens, G. Hinton, Generating Text with Recurrent Neural Networks, University of Toronto, 2011.
5. A. Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, karpathy.github.io, 2015.
6. M. Michaels, The Basics of Seussian Verse, mickmichaels.com, 2012.
7. J. Beck, The Secret to Dr. Seusss Made-Up Words, The Atlantic, 2015.
8. S. Ozair, Char RNN Tensorflow, Github Repository, https://github.com/sherjilozair/char-rnn-tensorflow.
9. S. Ruder, An Overview of Gradient Descent Optimization Algorithms, ruder.io/optimizing-gradient-descent/index.html/adagrad, 19 Jan 2016
10. D. Kingma, J. Ba,Adam: A Method for Stochastic Optimization, ICLR, San Diego, 2015.

Acknowledgements

We would like to acknowledge Percy Liang and Stefano Ermon for helping provide us with the tools to tackle our project objectives.
We would also like to thank Orion Despo for his guidance and assistance throughout the project life cycle.

CS 221 Final Project Poster Session
December 5, 2017