CIS530-Project-MS1

Akriti Gupta, Bhumika Singhal, Namita Shukla, Smruti Chourasia

29 November 2022

1 Dataset

Haikus are generally tough to write, hence tough to find. However, for a generative task we need abundant amount of data for the model to be able to learn the constrained structure of haiku. Hence we resorted to combining datasets from different sources - after applying appropriate pre processing to each. Dataset is built using 5 sources:

Haiku dataset from Tempes Libres [4]
This unique database contains the haikus (favourites) on this site since January 2000. There are over 140K Haikus in the dataset. The vast majority, over 110K, comes from Twitter associated with the "twaiku" hashtag.
Sam Ballas' PoetRNN corpus [5]
There are about 8000 haikus that the author collected by scraping the following websites: daily-haiku.org
Herval Freire's Haikuzao corpus [6]
This is a collection of about 6000 haikus
The three-line poems from the "Beyond Narrative Description: Generating Poetry from Images by Multi-Adversarial Training" corpus.
Haiku Dataset — Kaggle [7]
This contains haiku scraped and cleaned from `reddit.com/r/haiku`

We combine the different datasets into a final Pandas dataframe; the first 3 columns refer to the three sentences of the haiku, we also store the source of the dataset. Our next feature is count of syllables in each line since our main moto here is to ensure that the model learns the 5-7-5 structure of haiku.
Since we combine Haikus from different sources we need to pre-process to ensure quality of data is good. The main steps taken are

- To improve the structure of the poems and have uniformity we use FastPunct to add punctuation and capitalization to text
- Since the poems are majorly extracted from the web, it is imperative to ensure that the poem is coherent. Hence, we use our Evalutation metric (GRUEN) to assess the quality of the poems and tune the threshold to have enough poems for training.
- Feature Generation: Our first feature is Syllables which is important for model to understand the 5-7-5 structure. According to literature using plain text the models may not be able to learn the syllable structure hence we also plan on splitting the text into phonemes using Phonmizer. Apart from that we also plan on incorporating topics of text extracted from KeyBERT

We have a total of 154406 haikus for training purposes. Since our aim here is to train a generative model we plan to use all of our poems for training the model, we use the 80-20 split for Training and Development set. The poems generated by our model will then be analysed using the Evaluation metric to access the quality of model. The final row counts of our Train and Development set are 123524 haikus (train.csv) and 30882 haikus (dev.csv) respectively.

2 Literature Review

In [1], the authors inspired by neural language models and predicate argument relation explored various techniques for generating Haiku to share the culture of Haiku among young people using a emotional chatbot named Rinna. Firstly, they experimented using minimal vanilla recurrent neural network for learning character level neural language models. They further explored multi-layer RNN using LSTM and GRU for character-level language models. Following this they used Character Level Recurrent Convolutional Neural Network (RCNN) and Sequence Generative Adversarial Networks. They scraped various Haiku websites and also used Rinna(chatbot) query log for training the above mentioned models. The author's training set comprised on 100,000 Haikus, validation set and test set had 5000 Haiku's respectively.The models were hypertuned parameters like epoch and embedding size. The conclusion of these experiments were that RNN-LSTM performed better than the other models for web scraped Haikus whereas RCNN performed the best for Rinna generated Haikus.They were able to successfully launch the Haiku model in Rinna and obtain more than 50 million accesses in a few months.

In [2], the authors - Jack Hopkins and Doewe Kiela produced several models for generating poetry. They majorly focused on sonnets, which not only have a constraint on the number of syllables per line, but must follow a specific rhyme scheme. They proposed two novel methodologies for automatically generating rhythmic poetry in various forms. The first approach used a neural language model which was trained on phonetic encoding catered to learn an implicit representation of both form and content of English Poetry. This model could learn basic poetic features like rhythm, alliteration and rhyme. Here, they used an LSTM model which was trained using stochastic gradient descent to predict the next phoneme given a sequence of phonemes and fine tuned it.
The second approach, rather tricky, was a character-level model which could produce more coherent text, but did not have any constraints on the form. They constrained the text at sample time by using a discriminator that would reject text that didn't conform to the desired meter (syllable format of 5-7-5). For this, they used a simple character-level model (Sutskever et al., 2011) with some tweaks which was focused solely on content. Their results were considerable good. An indistinguishability test, where participants were asked to classify a randomly selected set of human "nonsense verse" and machine-generated poetry, showed generated poetry to be indistinguishable from that written by humans. In addition, the poems that were deemed most 'human-like', most aesthetic and most emotive, respectively, were all machine-generated.

The paper [3] introduces a model "Haikoo" which is a transformer based model that outperforms previous SOTA neural network based haiku poetry generators. This model contains a GPT-2 model, which was fine-tuned on haiku poems and is able to successfully generalize the qualities of haiku poetry while retaining flexibility to compose poems on entities never seen on the training data, and a Plug and Play Language Models, which was used to control the generated results further than the classic prompt approach towards a particular topic / context. Hence, Haikoo as a model is able to generate haikus which satisfy poetic constraints and range from topics which are lyrical to hilarious. Vanilla LeakGAN model was used a baseline. TextGAN and GPT-2 (+ PPLM [generation with bag-of-words approach for highly focused control over the output.]) were the models fine tuned with task specific data and tested against the baseline. The metrics used to evaluate the generated haikus were Perplexity and "Sensicality" (how much sense it made), "Wisdom" (stupidity of a poem) and "Overall Quality" assessment by AMT workers. Using Bigram Perplexity scores on a sample output of both models it was found that GPT-2 performs better than GAN approach. On each AMT metric as well, GPT-2 outperforms the baseline with a particularly high margin in the metric of "Sensicality".

3 Bibliography

1. Haiku Generation Using Deep Neural Networks, Xianchao Wu, Momo Klyen, Kazushige Ito, Zhan Chen, Microsoft Development Co., Ltd ,
2. Automatically Generating Rhythmic Verse with Neural Networks, Jack Hopkins et al, Facebook AI Research.
3. Haiku Generation : A Transformer Based Approach With Lots Of Control, Giacomo Miceli.
4. Tempes Libres Haiku: `http://www.tempslibres.org/tl/en/dbhk00.html`
5. Poet RNN: `https://github.com/sballas8/PoetRNN/blob/master/data/haikus.csv`
6. Haikuzo: `https://github.com/herval/creative_machines/blob/master/haikuzao/src/main/resources/haiku.txt`
7. Haiku Dataset From Kaggle: `https://www.kaggle.com/datasets/bfbarry/haiku-dataset`