

Automatic Generation of Lyrics in Bob Dylan's Style

Dongzhuo Li (lidongzh@stanford.edu), Chao Liang (chao2@stanford.edu), Tianze Liu (tianze@stanford.edu)

1. Introduction

Bob Dylan was recently awarded the Nobel Prize "for having created new poetic expressions within the great American song tradition". It is interesting to see if machine could learn his poetic style by looking at his lyrics. In this project, we use N-grams and Recurrent Neural Network (RNN) with Long Short Term Memory (LSTM) to model Dylan's lyrics, and eventually use the algorithms to generate samples of lyrics in Bob Dylan's style.

2. Data

The data we use includes lyrics of 465 songs downloaded from Bob Dylan's official website (728KB in total).

In order to make it easier for machines to process the data, we preprocessed the data by lower-casing all words and isolating punctuations from words.

3. N-grams

Training:

Find the distribution that maximizes:

$P("apple" | "John", "likes")$

Prediction:

Randomly sample from the distribution.



4. RNN with LSTM

LSTM: Memory that allows RNN to learn long-term dependencies.

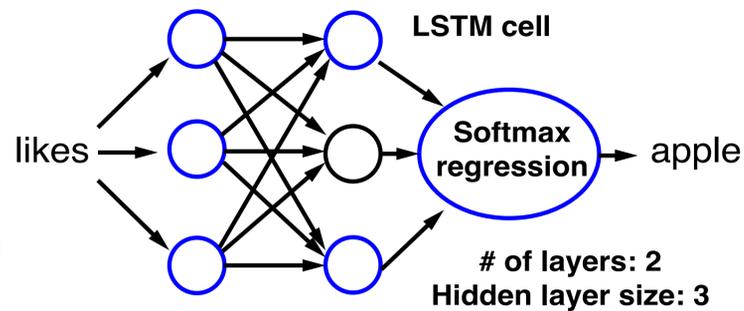
Example: I am from [China](#). I came to the US, but I still like [Chinese](#) food.

Training:

Update the weights and biases in each cell with back-propagation

Prediction:

Forward propagation in the neural network.



5. Word vs. Character level RNN

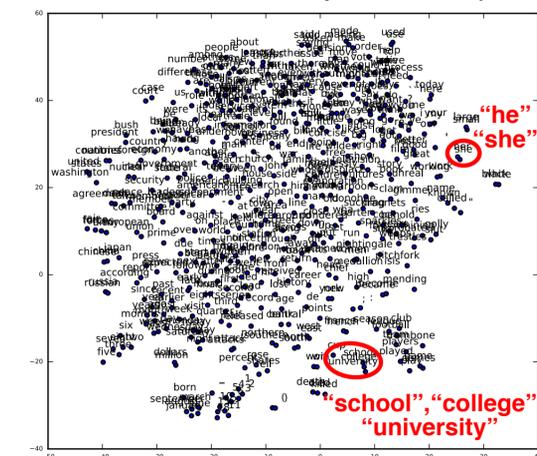
• In practice, RNNs work at either word level or character level.

• To make word level RNN more efficient, words are mapped into high dimensional vector in which semantically similar words are close to each (word embedding), before fed to RNN.

• In our experiments, character level RNNs performs significantly better than word level RNNs, likely due to the small number of words our data contains.

• In the following sections all results are generated by character level RNNs.

Word embedding from Global Vectors for Word Representation (GloVe)



6. Metrics for RNN Performance

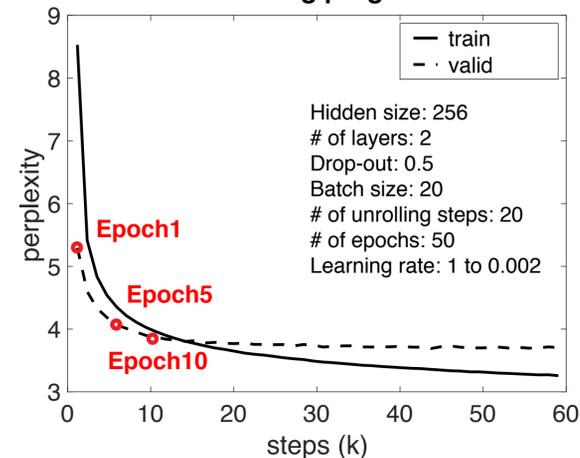
We use **perplexity** as the metrics for performance of RNNs, which is defined as:

$$perp = \exp\left(-\frac{1}{N} \sum_{i=1}^N \log p_{target_i}\right)$$

p_{target_i} is the probability of the i th word output by softmax regression following RNN.

7. Training RNN

Training progress



Epoch 1:

When, my prace, ana-peeviof
Plaevd Jos a beyf pays rome!
Thack Jpeps frays and her forbake
roses...

Epoch 5:

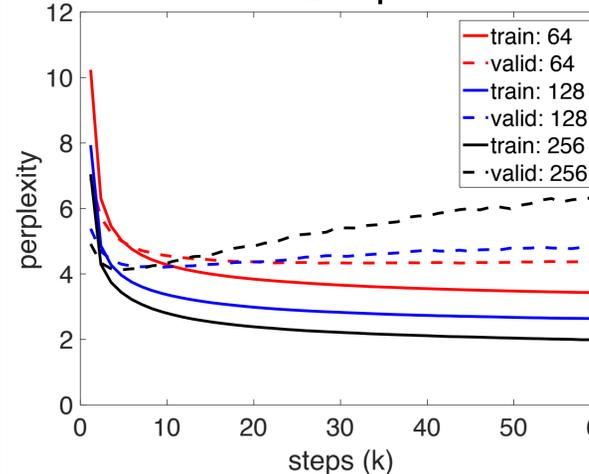
When, you play up to the whole laughed
Everything is she's crazy growid in me
If you one time for anythe frien...

Epoch 10:

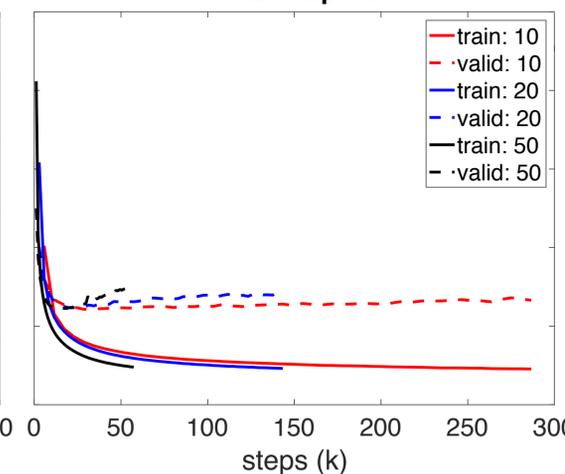
When, I say, 'That say I poor you,
Well, I got anythin' for ev'ryone
You think and I could fine...

8. Effects of Batch Size and Hidden Layer Size

hidden size experiment



batch size experiment



• Without drop-out, larger hidden layer size reduces training perplexity but increases validation perplexity, which indicates overfitting.

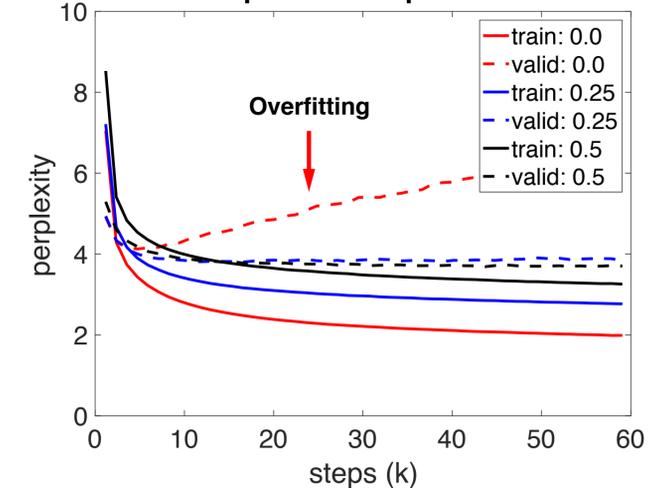
• Larger batch size has faster training speed and leads to earlier overfitting.

9. Regularization by Drop-out

• Drop-out means randomly disable a certain fraction of neurons in each iteration, in order to avoid over-fitting.

• In our experiments, a non-zero drop-out rate is very effective in preventing over-fitting.

dropout rate experiment



10. Sample lyrics

N-Grams

N=1: You weary, shine. road. Hell's dead fate me stand That a. winds With name?

N=3: One by one, until there were none. Two by two, they stepped into the night. Drinkin' white rum in a pie. Let the bird sing, let the bird fly.

CharRNN

Well, I want to hear my money from the river. Yes, and the next from the ground. With the wind blown of the day

I am going one . I'm in the moon. Something is the wind . I can't be seen

11. Discussion

N-Grams:

1. Low training perplexity (N=2: 32.3; N=3: 2), high test perplexity (~589902). This may be due to the irrelevance between lyrics of different songs and the small size of dataset.

2. The generated lyrics make more sense as N increases, but with higher probability of duplication.

CharRNN:

1. Larger hidden size has risk of causing overfitting, which can be regularized using dropout.

2. RNN seems good at capturing the grammar (syntax) of the sentences.

3. Need a larger dataset to train a better performing character level RNN.

12. References

- Andrej Karpathy. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>, last visited, 10/20/2016.
- Andrej Karpathy. <https://github.com/crazydonkey200/TensorFlow-char-rnn>. last visited, 10/20/2016.
- Bob Dylan, <http://bobdylan.com>, last visited, 10/20/2016.
- Goodfellow I, Bengio Y., and Courville A. 2016, Deep Learning, Book in preparation for MIT Press, <http://www.deeplearningbook.org>.
- Michael A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.
- Jurafsky D. and Martin J., Speech and Language Processing, Prentice Hall, 2008.
- Chen Liang. <https://github.com/crazydonkey200/tensorflow-char-rnn>, last visited, 12/12/2016.