CS 671A: Natural Language Processing

Lyric Generator

Group 30

Rahul Gupta
grahul@iitk.ac.in

Sharad Roy
rsharad@iitk.ac.in

Sai Harsha Nalluru
harshan@iitk.ac.in

Navneet Singh
snavneet@iitk.ac.in

Vikulp Bansal
vikulp@iitk.ac.in

Puneet Kumar Verma
puneet@iitk.ac.in

April 4, 2018

Problem Statement

The aim of natural language processing is to be able to generate or verify language or written content as good as human beings. A major part of the content is available to us in the form of songs and poems. Therefore it is imperative that artificial intelligence is able to generate poems or lyrics by learning them. We aim to move ahead in this direction and create an application which can learn content based on style, tempo, rhyme scheme e.t.c., belonging to different artists and generate acceptable lyrics.

Dataset & Libraries

- CMU Pronouncing Dictionary : For a given word, gives rhyming words present in its database. [4]
- NLTK : Natural Language Processing Toolkit
- WordNet : For a given word, it gives synonyms and hypernyms.
- Gensim: A python library that provides feature vector implementations like GloVe: Global Vectors for Word Representation [8].

Related Work

This problem has been a hot topic for projects in various universities and has been been researched for quite a while now. A previous work [1] used linear-interpolated Quadgram Model. Recently, a lot of work has been done in field of neural networks and several deep learning techniques are now available that are capable of learning context and writing style to generate new text.

A more recent work [7] used LSTM and FSA (Finite State Acceptor) for generating a poem on a given topic. Given a topic, it finds the related words and pair of rhyming words to end the lines. It forms a sequence of words using 10 syllables and uses CMU Dictionary [4] to compute stress patterns in a word. FSA is used to implement various constraints in sonnets and stress patterns. It also checks for ending lines with rhyme words and using punctuations. It does so by recording line number and syllable count. It uses LSTM to locate fluent path of FSA having highest score.

A novel algorithm [2] to generate poems from buzz words by providing the Rhyme Scheme and the syllables as inputs was proposed. Knowing the total number of syllables in each line, we can generate all combinations of compositions from the available syllables and one such composition constitutes the base. The ultimate and the penultimate syllables of each line are taken to search for the rhyming words and consequently fill the poem. We can extend this algorithm and attempt making syntactically sound sentences by restricting the successor words based on the rules of English Grammar.

Approach

Data Collection Data would be collected from multiple sources to experiment around with different styles of writing. Online lyrics of songs will be scraped from A-Z lyrics and SongMeanings. Other literary works like Shakespeare's work would be obtained from relevant online sources.

Training Given a series of words from our dataset, our model will try to predict the next word. This will be implemented using RNN based LSTM networks which will take the context of 6-7 words into account. If we would have enough time, we might also compare its performance with skipgram based models. We will then recompute our model using the error with respect to the ground truth. This will then be extended to generate arbitrarily long poems/lyrics.

Generating We will use the Beam search approach for machine translation described in [11] to generate multiple candidate lines and greedily select the best fitting line at each step to eventually generate a graph with a best-first structure.

Evaluation Following [10], the rhyme feature that we would be examining in this work is rhyme density. Rhyme density is defined as the total number of rhymed syllables divided by the total number of syllables. For evaluating the closeness of generated lyrics to an artist's work, we plan to use tf-idf score of a verse with respect to each of artist's verses as previously done in [9]. We also propose a novel style of evaluation in this domain as given in following What's New section.

What's new Build the generator from the existing approaches and restricting the output by checking POS score to ensure the outputs are meaningful. We are doing this to ensure that lyrics make syntactical sense.

References

[1] Hieu Nguyen, Brian Sa. *Rap Lyric Generator* Project Report, CS224, Autumn 2009, Stanford University.

[2] Dmitry Alimov *Buzzword Poem Generator*

[3] Kevin Lenzo *CMU Pronouncing Dictionary*

[4] *pronouncing 0.1.5* *Python Library for CMU pronouncing dictionary*

[5] Poetry database - Poetry Foundation,*Modern and Renaissance poetry for classification exercises*

[6] Count syllables - *Printing the number of syllables in a word*

[7] Marjan Ghazvininejad and Xing Shi and Yejin Choi and Kevin Knight, EMNLP, 2016 *Generating Topical Poetry*

[8] Penington, J., Socher, R., and Manning, C. (2014). *GloVe: Global Vectors for Word Representation.*

[9] Peter Potash, Alexey Romanov, Anna Rumshisky *GhostWriter: Using an LSTM for Automatic Rap Lyric Generation*

[10] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis *Dopelearning: A computational approach to rap lyrics generation*

[11] Alexander M., Rush Yin-Wen Chang C., Michael Collins *Optimal Beam Search for Machine Translation.*