

AI-based Poetry (Sonnet) Generation

Team Size: 2

Team Members :

Poorva Bhalerao



Harshada Yesane



Table of Content

Team Members

Overview

Problem Statement

Technologies/API used

Solution Architecture

Image Captioning

Sonnet Generation from Text



Overview

- The hackathon is to develop solution to generate sonnet given an image as input.
- Sonnet is a fourteen-line poem with a fixed rhyme scheme.
- Characteristics of Sonnet:
 - First three stanza contains 4 lines each
 - Last stanza contains 2 lines
 - Iambic Pentameter



Technologies/API used

Framework:

Python libraries and modules:

1. Tensorflow
2. Keras
3. Python (notebooks)

Solution Architecture

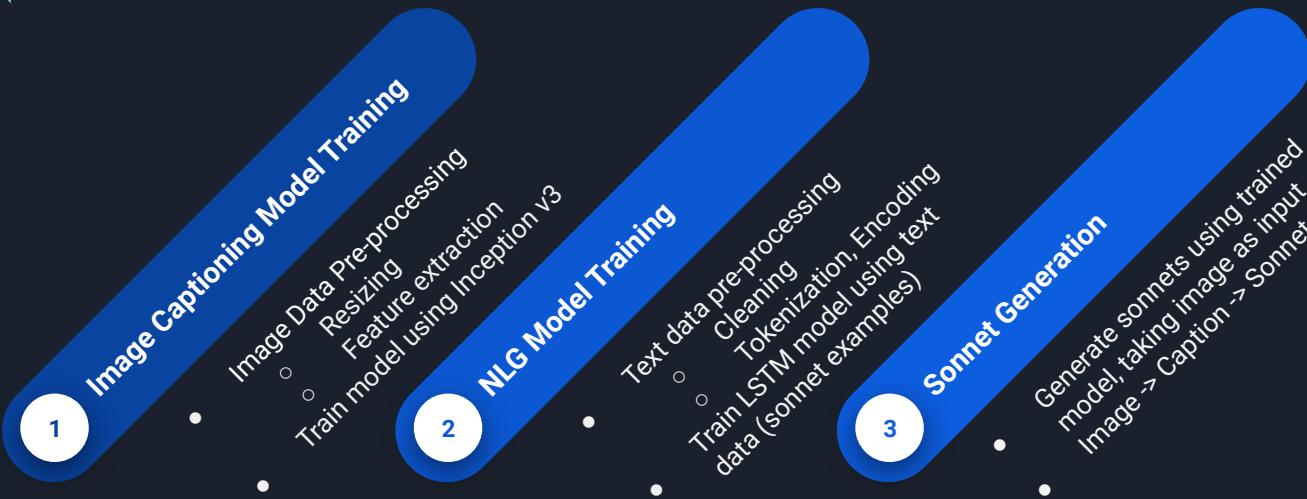




Image Captioning

- 1) Given an image like the example , our goal is to generate a caption.
- 2) Downloaded caption annotation files and image files
- 3) Grouping all the captions together having same image id.
- 4) Resizing all images into standard format as per encoding network.
- 5) Feature extraction is done by using InceptionV3 and then loading data stored for prediction.
- 6) Implementing those saved weights and generating a caption from image
- 7) Feed this caption as input to NLG_From_Text model

Accuracy of image captioning model is 64 % and during training



NLG from Text

1. The dataset is first broken into individual sentences and stored in a list
2. This list is then tokenized to get word index + a zero token (which means null)
3. For each line (sentence) in the input list-> create a sequence for every word such that it has the token of itself + previous words token in that sentence so for a sentence containing n words there will be a nxn matrix for each sentence containing 1 word to n words -

[0 0 0 0 token_first_word]
[0 0 0 token_first_words token_second_word] ...



Sonnet Generation from Text (contd)

4. This is done in order to train the neural network so that it predicts the next likely word based on the input sequence of previous word -
5. So the last value can be treated as the output and the rest of the list as input. In this manner success words are generated based on previous outcome.
6. Perform one hot encoding on the last column (value to be predicted)

Building the model

Sequential + embedding (dimension = 240 due to lot of variations as input is a language) + bidirectional LSTM + dense layer with softmax activation function + categorical loss function

Now train the model epochs = 100 (due to presence of unstructured data)



Thank you!

