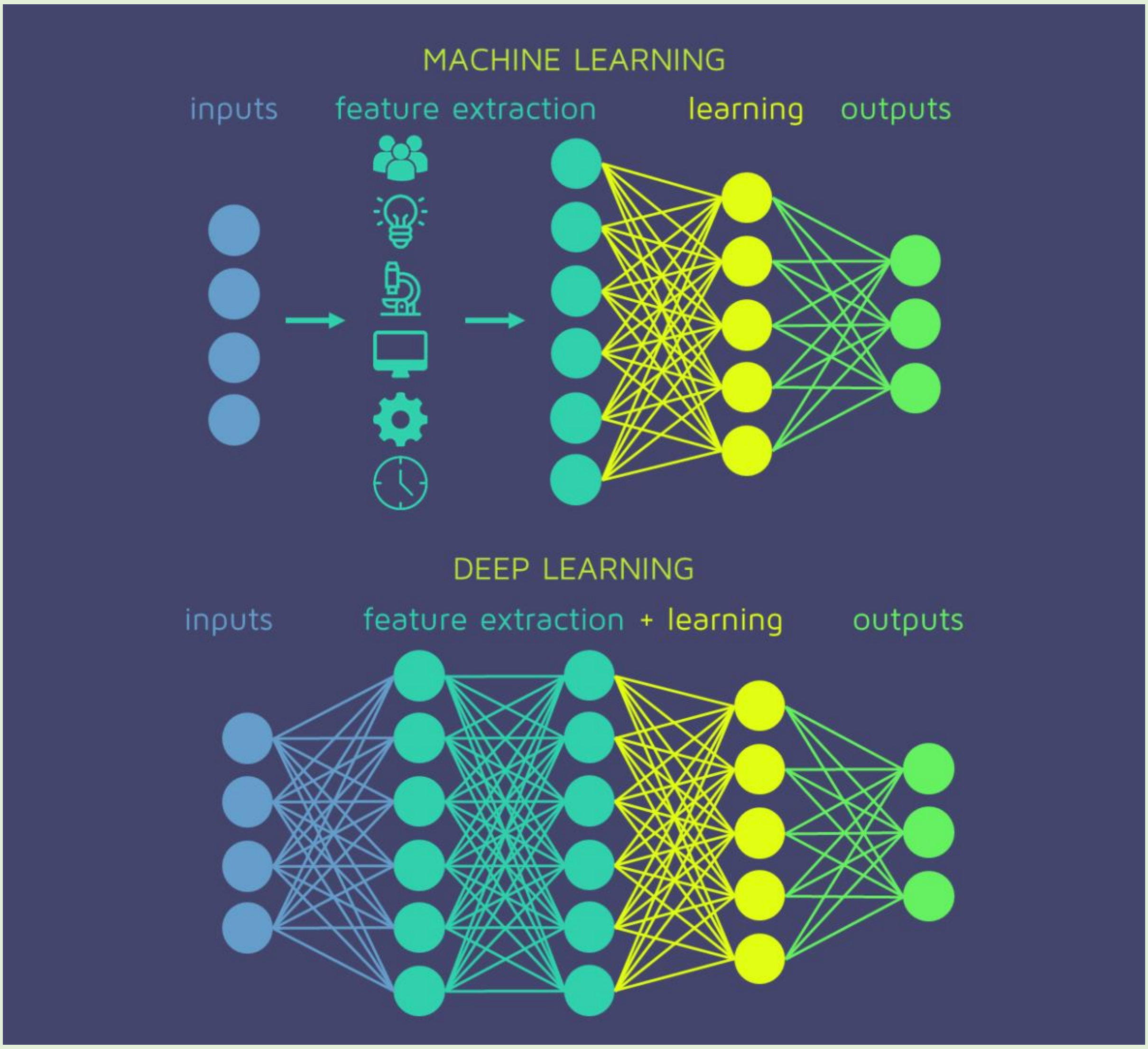
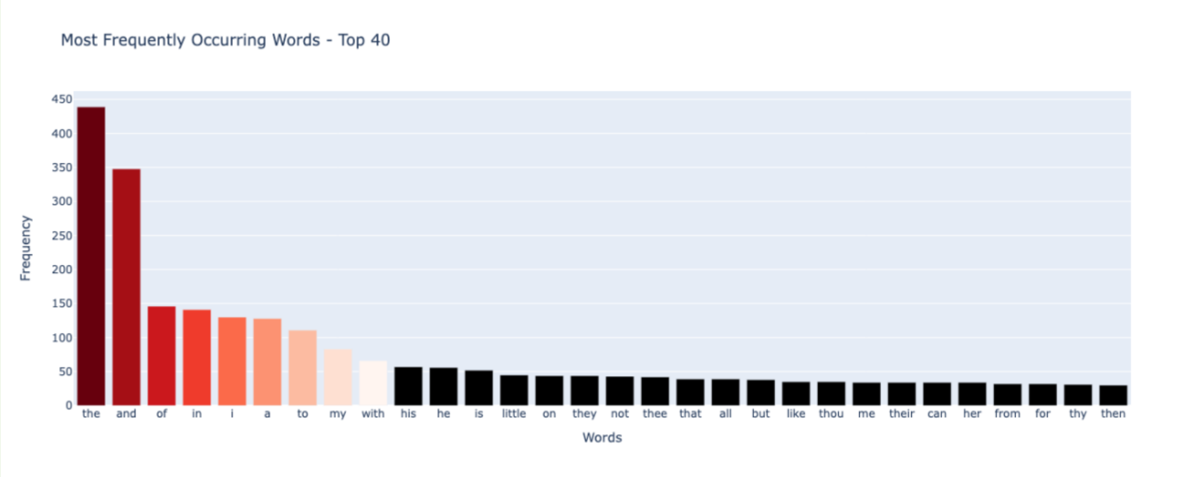
Generating Poems in the Style of William Blake using RNN-based Language Models and Evaluating their Quality with BLEU Score



Screenshots

1. Most Common Words and Wordcloud



2. Model Summary

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 300)          452100

 lstm (LSTM)                 (None, 100, 800)          3523200

 lstm_1 (LSTM)               (None, 100, 800)          5123200

 lstm_2 (LSTM)               (None, 800)               5123200

 dense (Dense)               (None, 1507)              1207107

=================================================================
Total params: 15,428,807
Trainable params: 15,428,807
Non-trainable params: 0
_________________________________________________________________
```

3. Poem 1:

```
Generated Poem:
the dimpling forge
voiceless blind southern
shouted shadows pensive
him partner tenfold us
went breathing lily tiger
ah glistening ready doubt
pity included unrest mole
BLEU Score: 0.9591894571091382
```

4. Poem 2:

```
Generated Poem:
my are false receive
stand leaped pity wight
clear sleep louder was
never cheer mouth slept
punish breathed wailing
taught wildered conveyed
clothes forgot selfish
BLEU Score: 0.9591894571091382
```

5. Poem 3:

```
Generated Poem:
play bar dreams meekin
dale iv wailing root
shaved birth plants
nothing shewd play
slender hands summer lost
dick smelling virgin man
duty fighting calling
BLEU Score: 0.9591894571091382
```

Methodology

The poem preprocessing involves removing all single characters, except for "a" and "I," as they are crucial for understanding the poem. Additionally, non-alphabetic characters and consecutive spacing are eliminated, and the text is converted to lowercase. A sequential model with an embedding layer, four LSTM layers of 120 neurons each, and a dense layer is constructed. The model is compiled using categorical cross-entropy as the loss function and the Adam optimizer. Finally, the model is trained for ten epochs with a batch size of 64 using the input and output sequences.

Critique

The BLEU score for the three poems remained the same despite the differences in each poem, and varied hyperparameters. The modification of the output function from Softmax to Sigmoid and Relu resulted in elevated loss rates. Despite experimenting with diverse input sequence lengths, the BLEU score did not improve. Consequently, the minimum length was adopted to optimize computational resources. The embedding size was set at 300 to enable a compact vector representation of each word while keeping the model's complexity at bay. This decision was based on the observation that an increase in embedding dimensions did not contribute to the improvement of the BLEU score. Furthermore, the trial of different batch sizes ranging from 8 to 128 did not result in any improvement in the BLEU score. To capture long-term dependencies between words in the input sequence, LSTM layers were employed and composed of 800 neurons to effectively learn such dependencies.

Conclusion

The hyperparameter tuning attempts made on the model architecture were found to be ineffective as the same outcomes were produced despite varying the hyperparameters. To address this, a more sensitive evaluation technique that can account for the intricacies of sentence construction could be considered in future research.