THE UNIVERSITY OF RHODE ISLAND

Context-Sensitive Quote Generation: Leveraging Transformer-Based Models for Quote Generation

- Arup Mazumder
- Piriyankan Kirupaharan

THINK BIG WE DO™

Outline

- Introduction
- Related work
- Methodology and Experiment
  - Data processing
  - Model selection
- Model Fine-Tuning and Training
  - Hyperparameter search
  - Deployment and demonstration
- Evaluation Metrics and Results
- Conclusion and limitations

Introduction

- People with IDD are more likely to experience abuse than those without disabilities [1].
- Therapists who work with people with IDD use motivational quotes calm them.
- Showing same quotes is not effective. Also, It is challenging to create new quotes that are customized to specific themes and contexts.
- We use the pre-trained language model GPT-2 fine tune and to generate context-specific quotes based on tags

Related work

- A Neural Network Approach to Quote Recommendation in Writings (Jiwei Tan et al. 2016) using LSTM.
- Quote Recommendation in Dialogue using Deep Neural Network (Hanbit Lee et al. 2016) using RNN and CNN

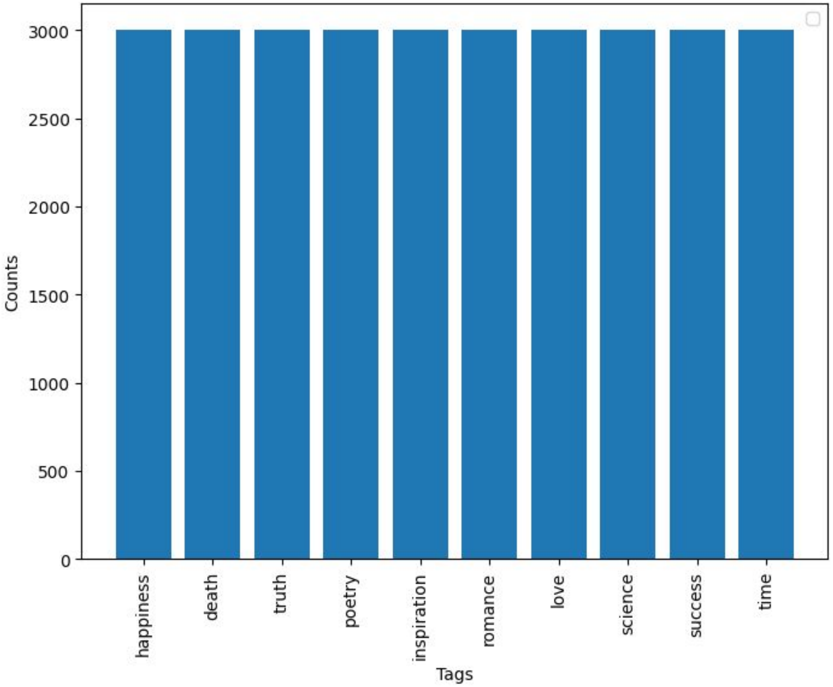
Data collection

- Dataset containing 30,000 quotes from 10 distinct categories, sourced from the popular website goodreads.com

- Dataset is publically available on Kaggle (https://www.kaggle.com/datasets/sanjeetsinghnaik/quotes-from-goodread)

- We also used another dataset which is similar to our training dataset from GitHub (https://github.com/ShivaliGoel/Quotes-500K) to prepare the test dataset.

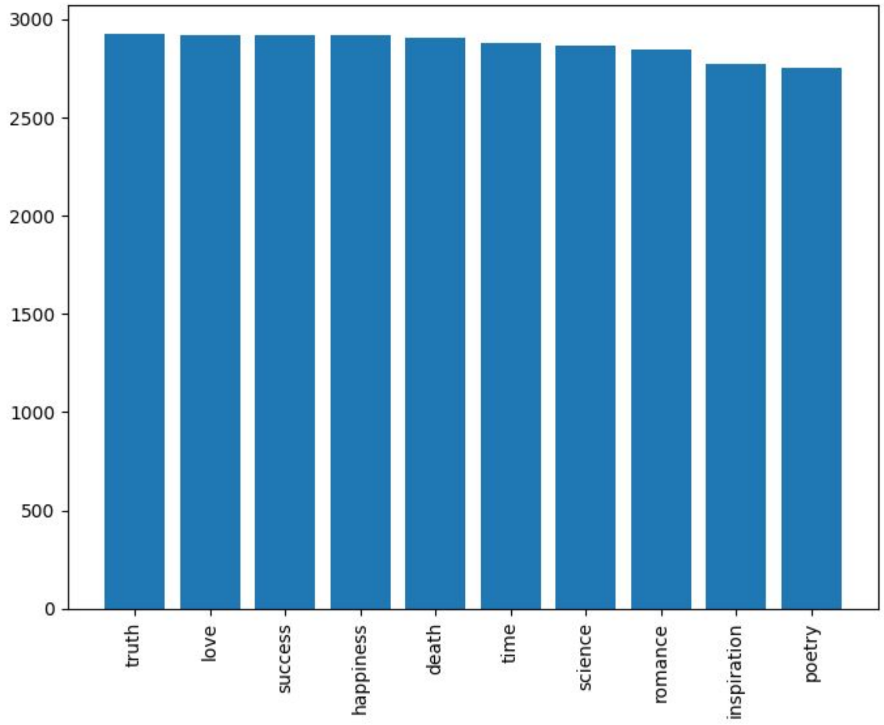
Data exploration - Training and Validation



- Quotes are equally distributed among the tags.
- Most of the quotes had less than 100 words.

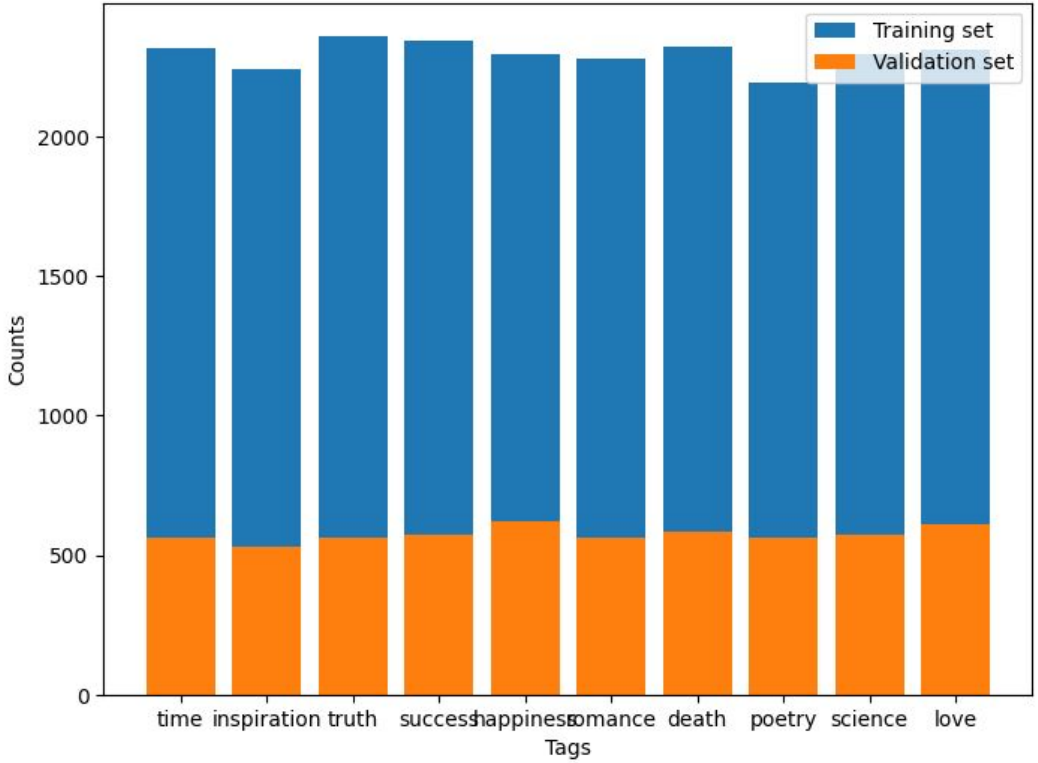
Sentences with 0-100 words: 27432
Sentences with 100-200 words: 1088
Sentences with 200-300 words: 142
Sentences with 300-400 words: 23
Sentences with 400-500 words: 9
Sentences with 500-600 words: 3
Sentences with 600+ words: 2

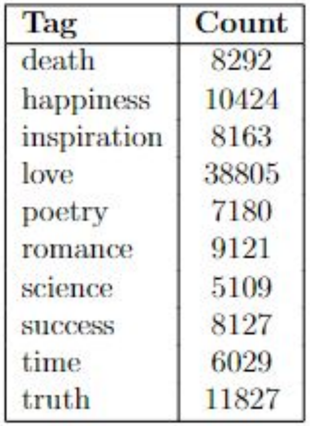
Data cleansing and transformation



- Cleansing techniques applied:
  - Remove quotes that are on-English.
  - Remove special characters like (“, @) .
  - Merged the quotes and tags, adopting the format <tag>: <quote>.
- The cleaned dataset contains 28699 quotes.

Train, Validation, Testing Data Splitting

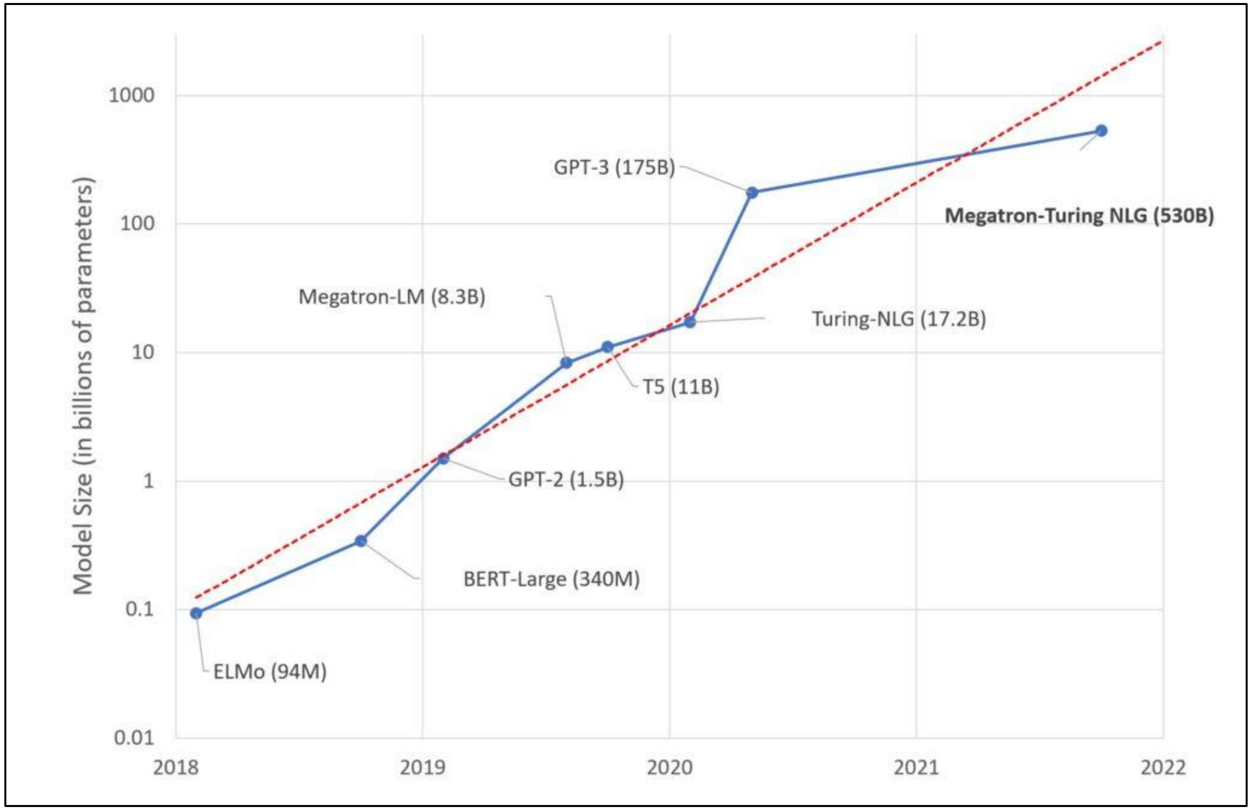




Tag	Count
death	8292
happiness	10424
inspiration	8163
love	38805
poetry	7180
romance	9121
science	5109
success	8127
time	6029
truth	11827

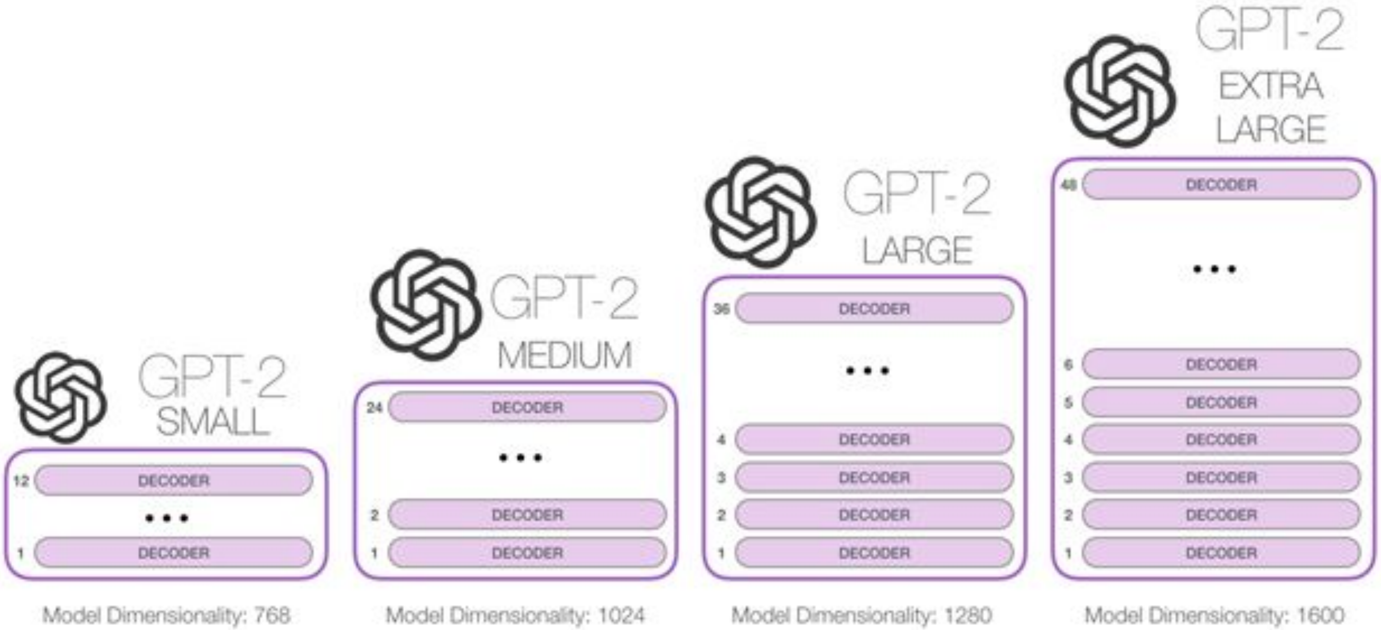
Table 1: Count of quotes per tag for test dataset

Model selection



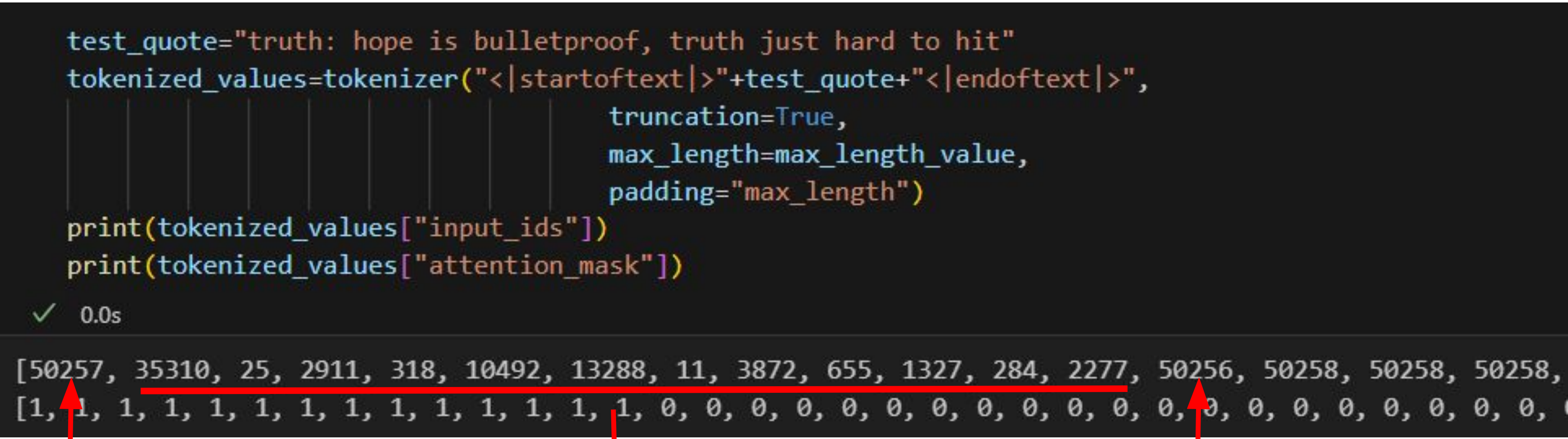
Source:https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/

Model from HuggingFace



Source:https://www.modeldifferently.com/en/2021/12/generaci%C3%B3n-de-fake-news-con-gpt-2/

Tokenizer



```
test_quote="truth: hope is bulletproof, truth just hard to hit"
tokenized_values=tokenizer("<|startoftext|>"+test_quote+"<|endoftext|>",
                           truncation=True,
                           max_length=max_length_value,
                           padding="max_length")
print(tokenized_values["input_ids"])
print(tokenized_values["attention_mask"])
```

✓ 0.0s

```
[50257, 35310, 25, 2911, 318, 10492, 13288, 11, 3872, 655, 1327, 284, 2277, 50256, 50258, 50258, 50258,
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Token for the start of the text

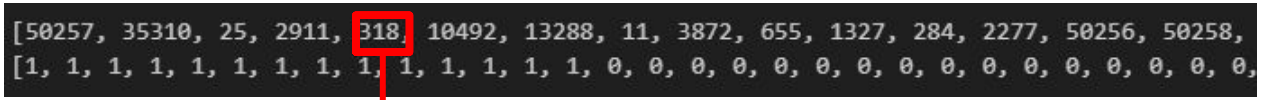
Token for the end of the text

Tokens for the given quote

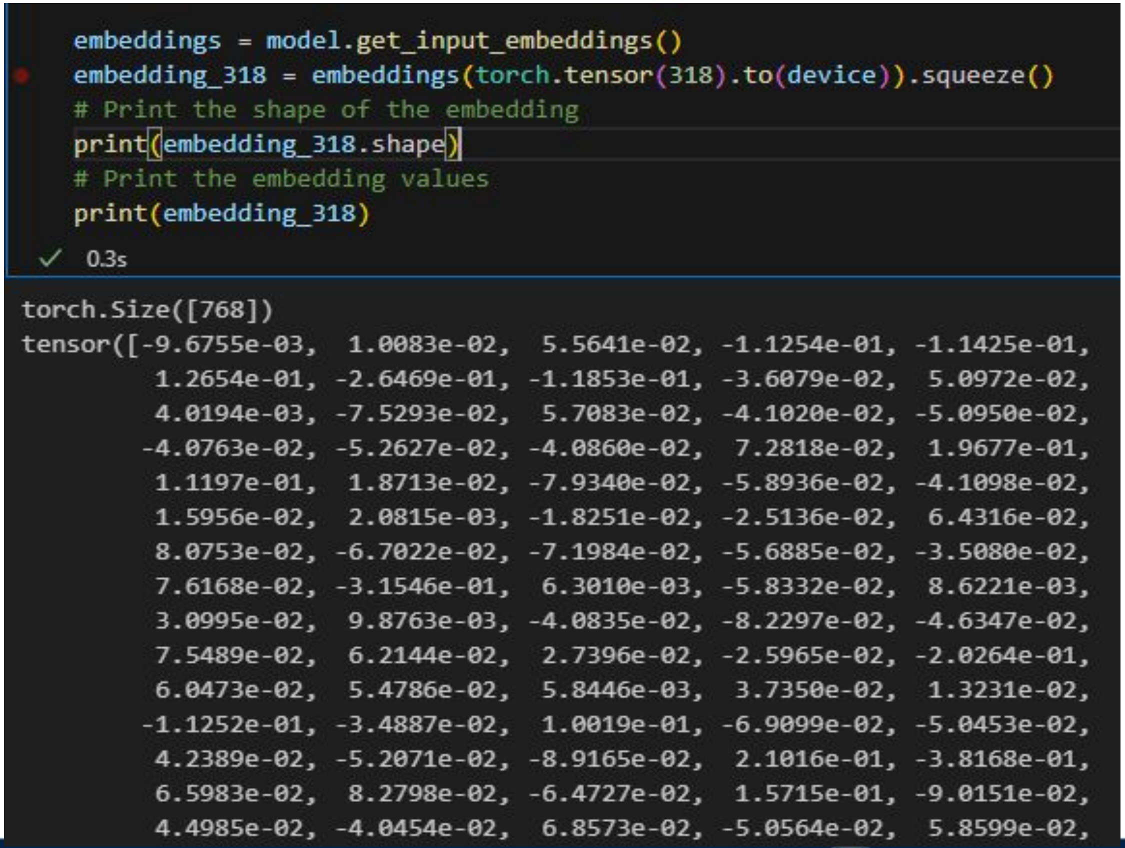
THINK BIG WE DO℠

THE UNIVERSITY OF RHODE ISLAND

Tokenizer embeddings



```
[50257, 35310, 25, 2911, 318, 10492, 13288, 11, 3872, 655, 1327, 284, 2277, 50256, 50258,
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```



```
embeddings = model.get_input_embeddings()
embedding_318 = embeddings(torch.tensor(318).to(device)).squeeze()
# Print the shape of the embedding
print(embedding_318.shape)
# Print the embedding values
print(embedding_318)
```

```
torch.Size([768])
tensor([-9.6755e-03,  1.0083e-02,  5.5641e-02, -1.1254e-01, -1.1425e-01,
         1.2654e-01, -2.6469e-01, -1.1853e-01, -3.6079e-02,  5.0972e-02,
         4.0194e-03, -7.5293e-02,  5.7083e-02, -4.1020e-02, -5.0950e-02,
        -4.0763e-02, -5.2627e-02, -4.0860e-02,  7.2818e-02,  1.9677e-01,
         1.1197e-01,  1.8713e-02, -7.9340e-02, -5.8936e-02, -4.1098e-02,
         1.5956e-02,  2.0815e-03, -1.8251e-02, -2.5136e-02,  6.4316e-02,
         8.0753e-02, -6.7022e-02, -7.1984e-02, -5.6885e-02, -3.5080e-02,
         7.6168e-02, -3.1546e-01,  6.3010e-03, -5.8332e-02,  8.6221e-03,
         3.0995e-02,  9.8763e-03, -4.0835e-02, -8.2297e-02, -4.6347e-02,
         7.5489e-02,  6.2144e-02,  2.7396e-02, -2.5965e-02, -2.0264e-01,
         6.0473e-02,  5.4786e-02,  5.8446e-03,  3.7350e-02,  1.3231e-02,
        -1.1252e-01, -3.4887e-02,  1.0019e-01, -6.9099e-02, -5.0453e-02,
         4.2389e-02, -5.2071e-02, -8.9165e-02,  2.1016e-01, -3.8168e-01,
         6.5983e-02,  8.2798e-02, -6.4727e-02,  1.5715e-01, -9.0151e-02,
         4.4985e-02, -4.0454e-02,  6.8573e-02, -5.0564e-02,  5.8599e-02,
```

- Each token will have a word embedding of size 768 which is the maximum embedding for GPT-2 small model.

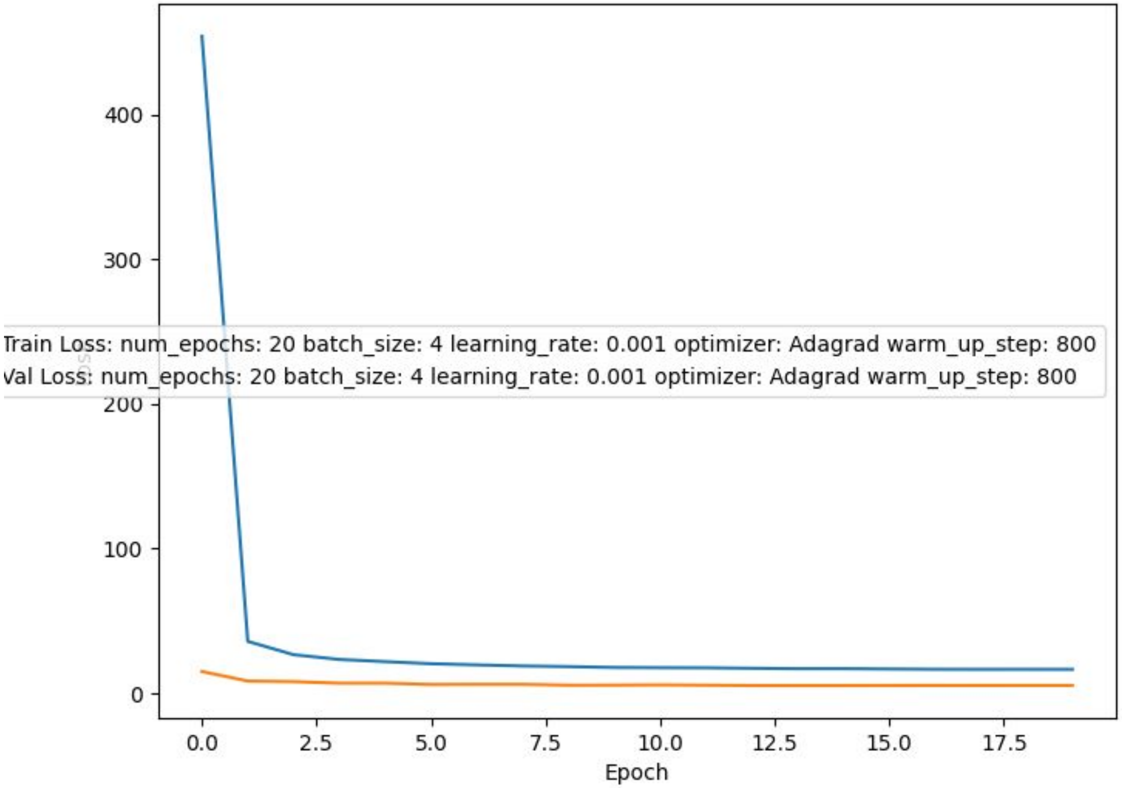
- Model was trained on RTX 3080, where the entire dataset took around ~10 hours to train.

Hyperparameter Search - First Round

- Batch Size : 4,
- LR : 1e-1,1e-3,
- Optimizer : Adagrad, AdamW,
- WarmUp Steps : 100, 800,
- Num Epochs : 20,

Best Combination: Batch Size : 4, LR : 1e-3, Optimizer : AdamW, WarmUp Steps : 800, Num Epochs : 20

Results - First Round Hyperparam Search



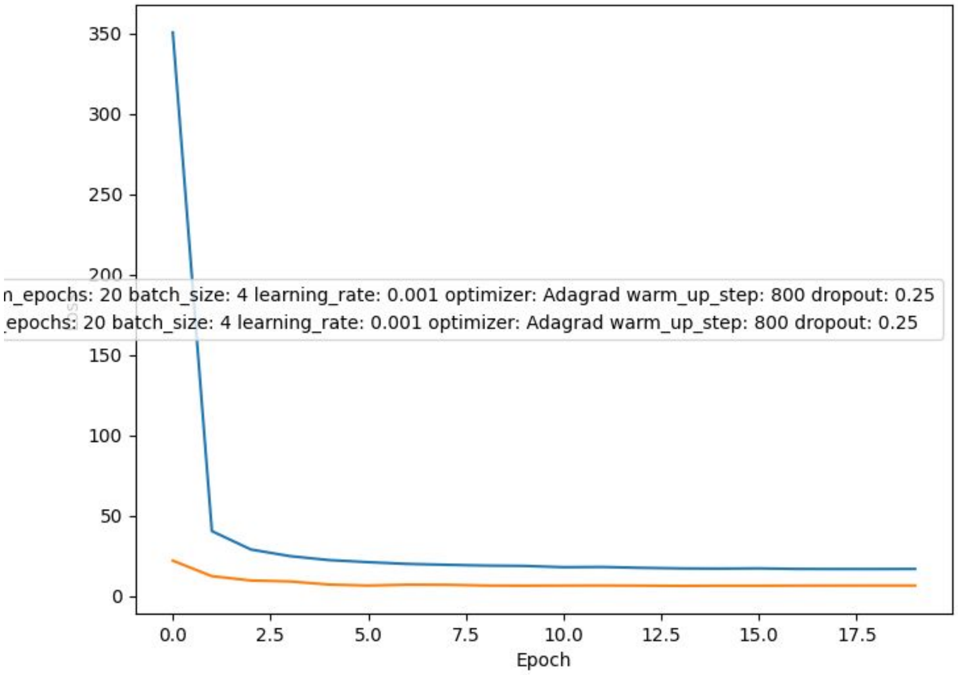
Hyperparameter Search - Second Round

Best hyperparams from First Round

- Dropout : 0.0, .25, 0.5

Best Combination:
Best from First Round + Dropout : 0.25

Results - Second Round Hyperparam Search



Hyperparameter Search - Final Round

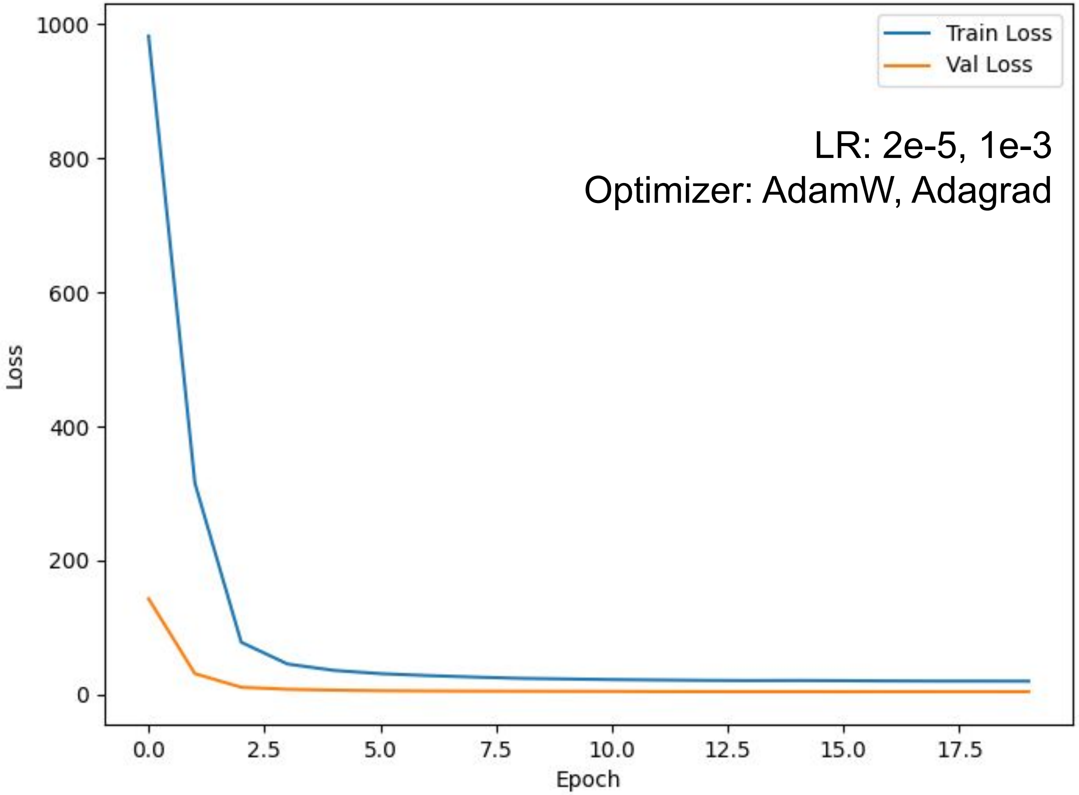
Best hyperparams from Second Round

- LR : 2e-5, 1e-3,
- Optimizer : Adagrad, AdamW,

Best Combination:

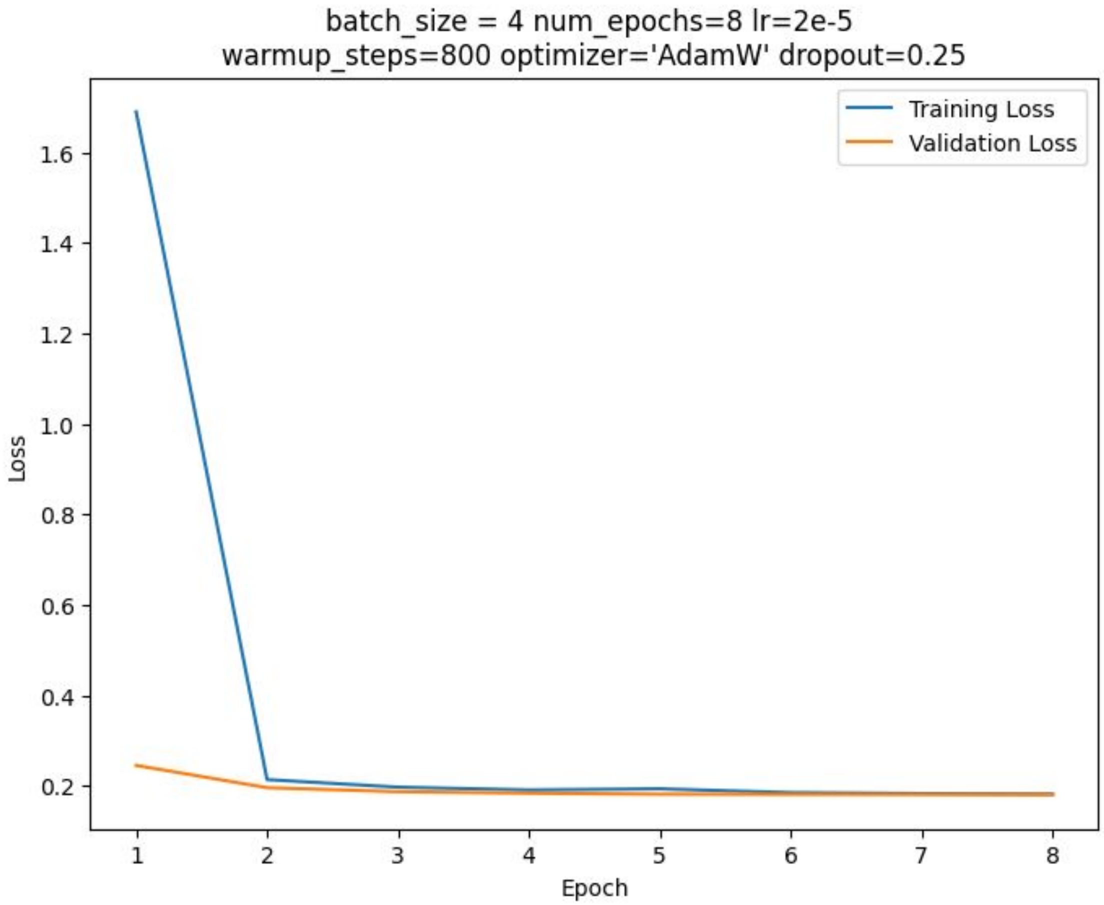
Best from Second Round + LR : 2e-5

Results - Final Round Hyperparam Search



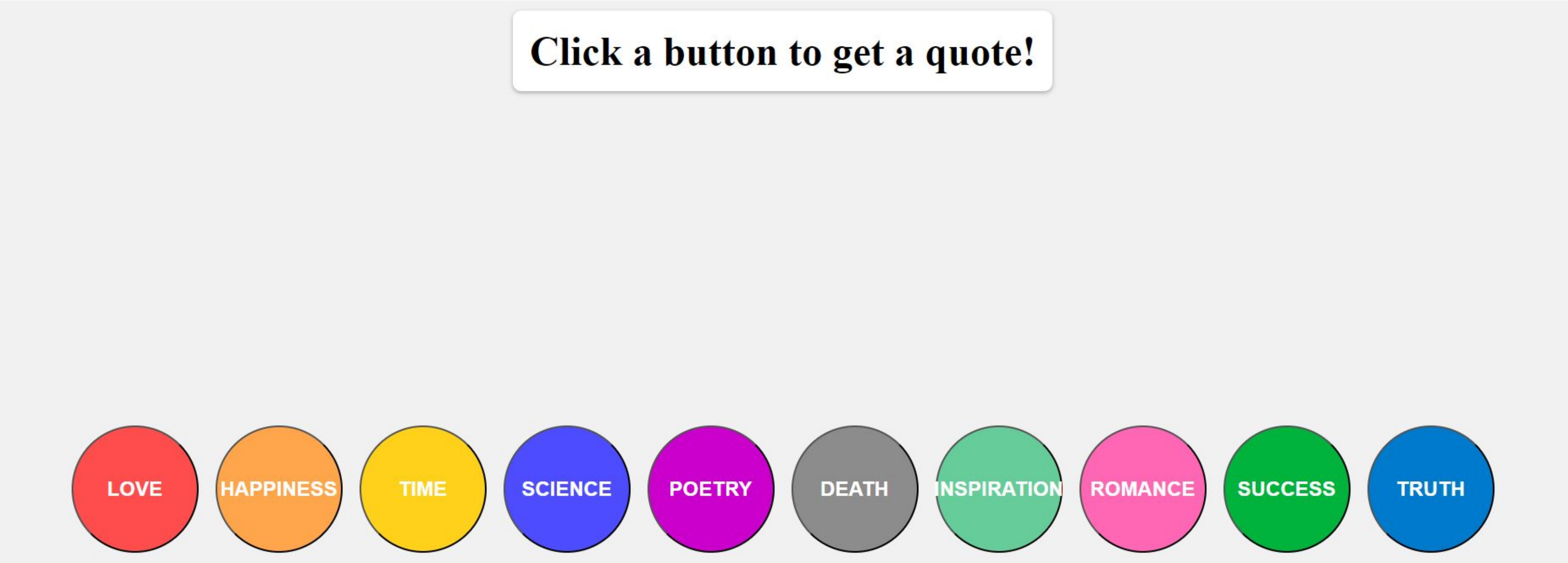
LR: 2e-5, 1e-3
Optimizer: AdamW, Adagrad

Results - Main Model



THINK BIG WE DO℠

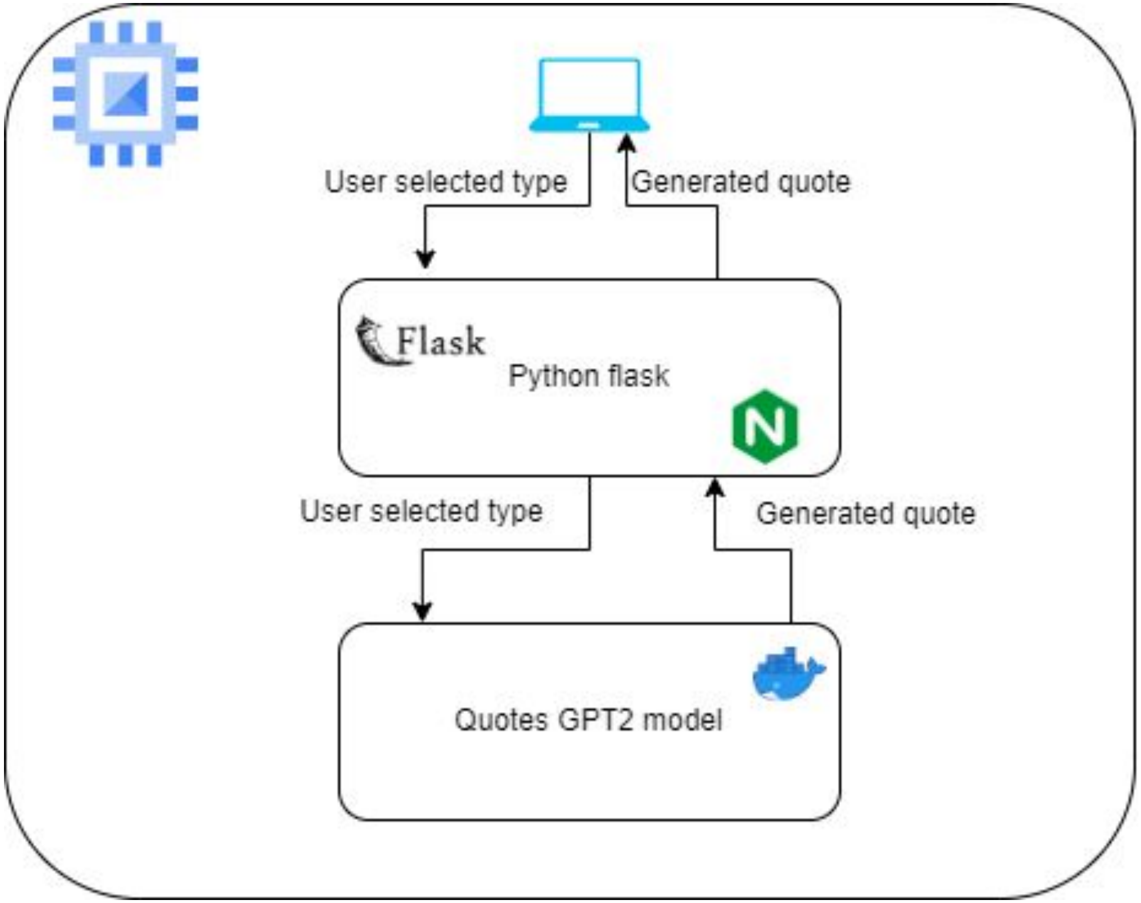
Web implementation: Demo



Click a button to get a quote!

Live URL: http://34.27.37.224/

Web implementation stack



Web implementation: Predictions by Model

Good Predictions:

Success - Our greatest success lies in yourself, and if you do not strive to achieve greatness, then you won't be able to do.

Happiness - Happiness comes to those who are blessed with the power to break free from their own limitations.

Bad Predictions:

Success - Success to live in the past, it is important to understand the purpose of the present.

Happiness - Happiness is never over. We need to live with it because it doesn't exist anymore. Happiness doesn't exist at all. It is the only.

Limitation

The performance of models can be limited by dataset noise, inconsistencies, or lack of diversity.

Current metrics like perplexity may not fully reflect the creativity and contextual relevance of the generated quotes.

Personal computers may lack the computational power to train large datasets efficiently, leading to long training times and potential errors.

Future Works

Alternative Models, Architectures: GPT-2-Large, GPT-3, or T5.

Dataset improvements: Increasing the number of quotes and categories can contribute to better fine-tuning and, ultimately, improved quote generation.

Tokenization and preprocessing: Investigating different tokenization methods and advanced preprocessing techniques.

Evaluation Metrics: Developing more sophisticated evaluation metrics (e.g. BLEU) that capture the creativity, novelty, and contextual relevance.

Conclusion

- Demonstrated the potential of fine-tuning pre-trained GPT-2 on a dataset of 30,000 quotes for context-aware quote generation.
- Emphasized the role of data preprocessing and tokenization in effective model training and fine-tuning.
- Used perplexity metric for performance evaluation, revealing satisfactory results and model efficacy.