

# CS7643 Spring '24: Assignment 4

Tyler Burki  
tburki3@gatech.edu

April 2, 2024

## 1 Theory Problem Set

### 1.1 RNN Backprop

First let us define the structure of the network:

$$a_t = Ux_t + Wh_{t-1} + b \quad (1)$$

$$h_t = \tanh(a_t) \quad (2)$$

$$o_t = Vh_t + c \quad (3)$$

$$\hat{y}_t = \text{Softmax}(o_t) \quad (4)$$

$$L_t = CE(\hat{y}_t, y_t) \quad (5)$$

To find  $\frac{\partial L_2}{\partial b}$  let us define the problem via chain rule:

$$\frac{\partial L_2}{\partial b} = \frac{\partial L_2}{\partial o_2} \cdot \frac{\partial o_2}{\partial h_2} \cdot \frac{\partial h_2}{\partial a_2} \cdot \frac{\partial a_2}{\partial b} \quad (6)$$

Then we can find each partial from (6):

$$\frac{\partial L_2}{\partial o_2} = \hat{y}_2 - y_2 = \text{Softmax}(o_2) - y_2 \quad (7)$$

$$\frac{\partial o_2}{\partial h_2} = V \quad (8)$$

$$\frac{\partial h_2}{\partial a_2} = 1 - \tanh^2(a_2) \quad (9)$$

$$\frac{\partial a_2}{\partial b} = 1 \quad (10)$$

Substituting equations (7) - (10) into (6) we get:

$$\frac{\partial L_2}{\partial b} = (\text{Softmax}(o_2) - y_2)(V)(1 - \tanh^2(a_2))(1) \quad (11)$$

To define the equation in terms of the inputs we first find the functions in terms of the inputs:

$$a_1 = Ux_1 + b \quad (12)$$

$$h_1 = \tanh(a_1) = \tanh(Ux_1 + b) \quad (13)$$

$$a_2 = Ux_2 + Wh_1 + b = Ux_2 + W(\tanh(Ux_1 + b)) + b \quad (14)$$

$$h_2 = \tanh(a_2) = \tanh(Ux_2 + W(\tanh(Ux_1 + b)) + b) \quad (15)$$

$$o_2 = Vh_2 + c = V(\tanh(Ux_2 + W(\tanh(Ux_1 + b)) + b)) + c \quad (16)$$

Then, substituting (12) - (16) into (11) we get the partial with respect to the inputs:

$$\frac{\partial L_2}{\partial b} = (\text{Softmax}(V(\tanh(Ux_2 + W(\tanh(Ux_1 + b)) + b)) + c) - y_2)(V)(1 - \tanh^2(Ux_2 + W(\tanh(Ux_1 + b)) + b)) \quad (17)$$

The unfolded computation graph for the network is shown in *Figure 1*.

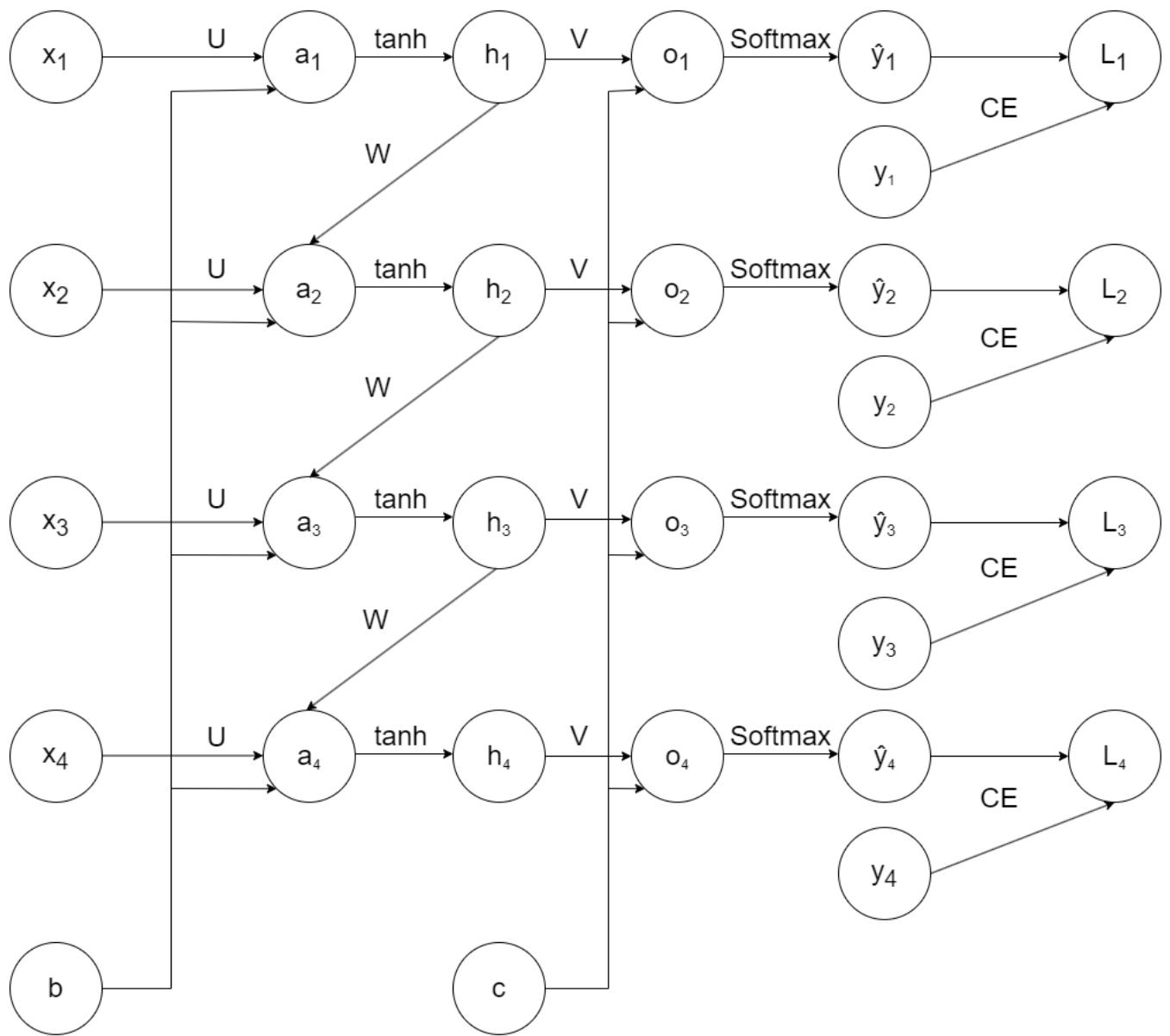


Figure 1: Unfolded computation graph of RNN specified in 1.1

## 2 Paper Review

This section reviews and analyzes the 2020 paper *Language Models are Few-Shot Learners* by Brown et al. [1]

### 2.1 Review

In *Language Models are Few-Shot Learners*, Brown et al. present their large language model, GPT-3, and discuss its performance on a variety of common NLP tasks under zero-shot, one-shot and few-shot transfer learning conditions. The authors begin by showing that as natural language models grow in size, so does their ability to generalize well to many tasks with few to no task-specific training examples. They demonstrate the power of the 175 billion-parameter GPT-3 over its 1.5 billion-parameter ancestor GPT-2 by scoring it against well-known NLP benchmarks. For some tasks, such as closed-book question answering, they have found that GPT-3 approaches state of the art fine-tuned models in the zero-shot case, and can eclipse them in the few-shot case. For other tasks like natural language inference the authors note their model lags state of the art, and in the zero and one-shot cases amounts to only slightly better performance than random guessing.

The authors dedicate a large section of the paper to discussing data contamination and ethical concerns, which help lend credibility to the research in that the results cannot simply be taken at face value. Errors in removing contaminated data are documented and plans for future releases are suggested even when testing shows contamination may not have more than a trivial impact on results. Social biases are clearly outlined and explained through examples, and they explain why such biases can be harmful to certain groups.

Personally, I was most in awe of the sheer size of the model and the lengths necessary to obtain and clean the pre-training data. I found it interesting that data contamination was found post-training and that the cost of re-training was prohibitive enough to simply note the fact and move on. This alone puts the scope of the model into perspective for me, when a company with the resources of OpenAI finds it infeasible to retrain. I was also surprised at how little such models are used for malicious activity, but agree with the authors that as large language models progress the return for using them in such a way will increase dramatically.

### 2.2 Questions

**How might we approach the technical limitations (e.g., changes in architecture, other context/data, optimization, etc.) mentioned in sections 5/6?**

A number of limitations of the model are described in section 5, including lack of bi-directional architecture for simplicity (e.g. denoising) (p.33), limits to the pre-training objective function (i.e. too broad of scope) (p.34), poor sample efficiency (i.e. training on more data than humans will ever see while still lagging their performance) (p.34) and expense of training/inference (p.34,39).

The first step to dealing with these limitations is to identify and describe them. The authors provide an extensive list in section 5 and even promise to release uncurated samples that underline the weaknesses of the model so that they may be better understood by the at-large community

(p.33). They then proceed to offer potential solutions to the limitations. Model distillation—the process of scaling-down a large, pre-trained model for a specific task—is provided as a potential solution for limiting the expense of training and inference (p.34, 39). Introducing additional modalities such as video and real-world physical interaction could expand the model’s domain and help it better identify important subjects (p.34). There is even mention of using other ML-based methods such as reinforcement learning to help fine-tune the model for specific tasks (p.34).

Overall, the approach for handling limitations appears rooted in simplifying the model by training on higher quality data while simultaneously focusing resources on the most important features of that data. This includes finding ways to preserve the generalization ability of the current model while introducing additional techniques to hone in on and fine tune for specific tasks—particularly those tasks that are currently difficult for the model to perform, such as “comparison” tasks (p.33). In a sense I am left wondering if this goal is in line with some of the proposed approaches such as adding additional modalities to the training set, but agree that the additional context is likely necessary to improve future outcomes.

**What are the social implications of deploying these models for various uses (e.g., to generate image captions, answer questions as chatbots, etc.)?**

The authors outline a number of fairness, bias and representational concerns in section 6. These concerns could lead to unintended and negative outcomes for a range of socially-based tasks. The authors note that “internet-trained models have internet-scale biases” (p.36). This means that because the model received a large amount of training data from the open internet, any and all biases present there likely found their way into the model. One test to demonstrate this was an association of gender to occupation, in which men were unilaterally over-represented (p.36). With regard to race, the model disproportionately outputs negative sentiment regarding black subjects and positive sentiment for Asian subjects (p.37,38). Similar results are obtained for topics of religion, where Islam is often associated with terrorists/terrorism while Christianity’s “worst” traits may be much more reasonable like “ignorant” and “judgmental” (p.38).

It is not difficult to project how such biases could play out during deployment. For example, an image of muslims praying could be detrimentally labeled as “terriosts plotting” or even how a question about the black variant of a product could return a more negative sentiment than its peers. These sorts of outcomes go beyond a correct/incorrect evaluation and can have far-reaching, negative impacts on already marginalized groups.

Thankfully, the authors bring these issues to the forefront willingly, and discuss the inherent difficulties in removing bias from large models as well as mitigation attempts via Model Cards and Model Reporting (p.39). Paths for further research are discussed, and a holistic approach suggested to avoid “blindspots” in metric driven bias removal to ensure that these potentially devastating features are removed as quickly and thoroughly as possible (p.39).

### 3 Coding

#### 3.1 Seq2Seq Results: Default Configuration

Model	Training Loss	Training Perplexity	Validation Loss	Validation Perplexity
RNN	4.2826	72.4300	4.3441	77.0231
LSTM	3.0273	20.6421	3.1893	24.2720
RNN w/ Attention	3.4277	30.8066	3.4464	31.3886
LSTM w/ Attention	2.9733	19.5557	3.1453	23.2259

Table 1: Results of Seq2Seq training on default hyperparameters

#### 3.2 Seq2Seq Explanation: RNN vs. LSTM

The LSTM saw loss reduced by 25% and perplexity reduced by 66% over the simple RNN. This is the result of the LSTM’s ability to leverage its gating architecture and memory cell to learn long-range dependencies. Because no action is taken toward vanishing gradient mitigation with the RNN (e.g. gradient clipping), the LSTM is better able to preserve associations between terms by retaining gradients for important terms across the sequence. Without these mechanisms, the RNN is unable to traverse the loss function to the same depths as the LSTM, as seen by the difference in loss. The effect of the long-range dependencies modeled by the LSTM can be seen in the significant decrease in perplexity—indicating that the LSTM is making higher quality (more probable) predictions at each step than the RNN.

#### 3.3 Seq2Seq Explanation: RNN vs. RNN w/ Attention

The leap from LSTM to LSTM w/ Attention is not as significant as between RNN and LSTM. However, adding attention to the LSTM still reduces the loss and perplexity by marginal amounts. Attention allows the LSTM to more readily discard information that will not be helpful in predicting the remaining sequence while placing more emphasis on the associations that will lead to better predictions. The sequences used for training are relatively short (20 words) and this is likely the cause of the marginal improvement over vanilla LSTM—there simply aren’t enough words and we can incorporate nearly all information in both models. As the length of the sequence grows, attention should play a larger role and facilitate better memory of important associations.

#### 3.4 Seq2Seq Best Model

	Training Loss	Training Perplexity	Validation Loss	Validation Perplexity
Best Model	2.3354	10.3331	3.1302	22.8784

Table 2: Results of Seq2Seq training on best hyperparameters found

Hyperparameter	Default	Best
model_type	RNN	LSTM
BATCH_SIZE	128	256
attention	False	True
learning_rate	1e-3	5e-3
encoder_emb_size	128	128
encoder_hidden_size	128	256
encoder_dropout	0.2	0.4
decoder_emb_size	128	128
decoder_hidden_size	128	256
decoder_dropout	0.2	0.4
EPOCHS	20	30

Table 3: Default vs. best values for hyperparameters (highlighted cells were changed from default)

### 3.5 Seq2Seq Best Model Learning Curves (Perplexity)

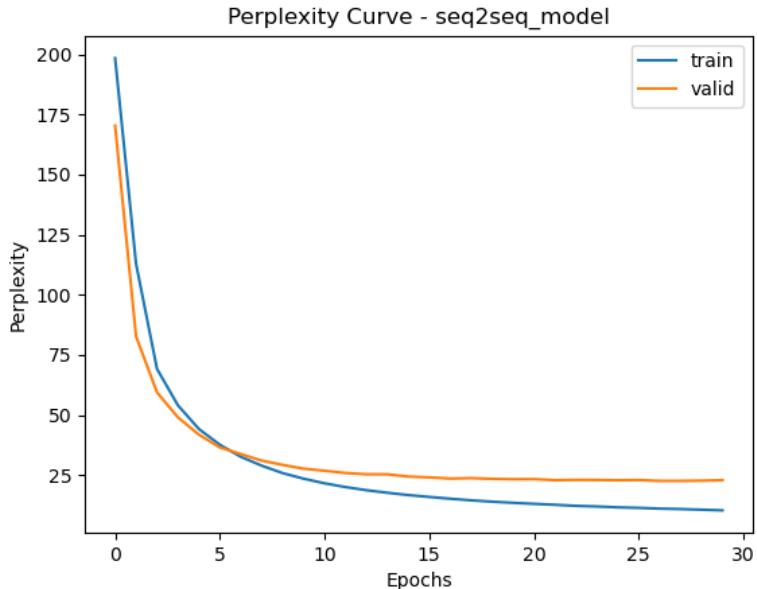


Figure 2: Perplexity curves for best Seq2Seq model

### 3.6 Seq2Seq Explanation: Best Model

I found it difficult to reduce the validation loss and perplexity beyond what the LSTM w/ attention model was able to accomplish. I was eventually able to reduce the loss and perplexity marginally, but at the cost of overfitting the training data significantly. I attempted to mitigate the issue

through the use of regularization in the form of increasing the dropout rates, but was unable to find a good balance between regularization and loss/perplexity. To improve metric scores, I increased the representational power of the hidden layers by increasing their size two-fold while also increasing the batch size to reduce noise in the data per batch. I also increased the learning rate five-fold in an attempt to walk down the steeper stretches of the loss function more quickly. However, oscillation around saddle points was apparent with training/validation loss sub 4.0, and a longer training process with a lower learning rate would likely show benefits here. With more time and better access to compute resources, overfitting could be reduced by lowering the batch size to introduce more noise into the data for better generalization. More work could also be done experimenting with dropout rates (increase) and hidden layer sizes (decrease) to improve fitment and generalization. Scheduling decreases in the learning rate would improve the model’s ability to navigate the loss surface as the reduction in loss decelerates.

### 3.7 Transformer Results

Model	Training Loss	Training Perplexity	Validation Loss	Validation Perplexity
Encoder Only - Default	2.1268	8.3884	2.9512	19.1282
Full - Default	1.3290	3.7771	1.5868	4.8879
Full - Best	1.1785	3.2495	1.5678	4.7961

Table 4: Results of transformer training scenarios

Hyperparameter	Default	Best
BATCH_SIZE	128	128
learning_rate	1e-3	7e-4
hidden_dim	128	128
num_heads	2	2
dim_feedforward	2048	2048
num_layers_enc	2	4
num_layers_dec	2	4
max_length	43	43
dropout	0.2	0.275
EPOCHS	10	20

Table 5: Default vs. best values for hyperparameters (highlighted cells were changed from default)

### 3.8 Full Transformer Best Model Learning Curves (Perplexity)

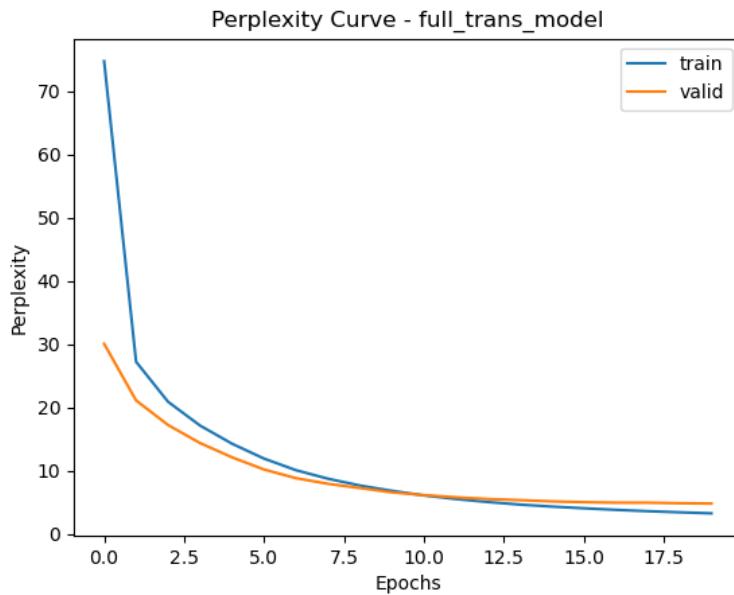


Figure 3: Perplexity curves for best full transformer

### 3.9 Full Transformer Explanation: Best Model

Since the full transformer performed significantly better than all other models previously examined in this exercise, I attempted to increase the representational power of the model in order to learn the last low-level associations the default parameters miss. I accomplished this by increasing the number of encoder and decoder layers from 2 to 4. As the data is encoded (and subsequently decoded) a number of times consecutively, the model becomes more capable of learning richer patterns amongst the data. This additional complexity can lead to overfitting, so I increased dropout by 37.5% as a regularization measure. In spite of this, the model begins to overfit around epoch 11 but does return a marginal improvement over the default parameters by the final epoch. The learning rate was decreased by 30% to accommodate the increase in regularization, and the epochs doubled to account for the slower rate of learning.

### 3.10 Transformer (Encoder Only) Translation Results

Input sentence	Back translation
'<sos>', 'a', 'man', 'in', 'an', 'orange', 'hat', 'starring', 'at', 'something', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'in', 'an', 'orange', 'hat', 'an', 'something', 'something', '.', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'boston', 'terrier', 'is', 'running', 'on', 'lush', 'green', 'grass', 'in', 'front', 'of', 'a', 'white', 'fence', '.', '\n', '<eos>', '<pad>'	'<sos>', 'a', 'boston', 'of', 'runs', 'the', 'the', 'grass', 'in', 'of', 'front', 'white', 'white', 'fence', '.', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'girl', 'in', 'karate', 'uniform', 'breaking', 'a', 'stick', 'with', 'a', 'front', 'kick', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'girl', 'in', 'a', 'a', 'a', 'a', 'with', 'a', 'shawl', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'people', 'are', 'fixing', 'the', 'roof', 'of', 'a', 'house', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'people', 'are', 'the', 'roof', 'of', 'the', 'a', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>, '<eos>'
'<sos>', 'a', 'group', 'of', 'people', 'standing', 'in', 'front', 'of', 'an', 'igloo', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'group', 'of', 'people', 'standing', 'standing', 'front', 'of', 'of', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>, '<eos>'
'<sos>', 'a', 'guy', 'works', 'on', 'a', 'building', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'guy', 'is', 'on', 'a', 'building', 'building', '\n', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'man', 'in', 'a', 'vest', 'is', 'sitting', 'in', 'a', 'chair', 'and', 'holding', 'magazines', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'in', 'a', 'vest', 'a', 'is', 'sitting', 'a', 'chair', 'and', 'a', '.', '.', '<eos>', '<eos>', '<eos>', '<eos>, '<eos>'
'<sos>', 'a', 'mother', 'and', 'her', 'young', 'song', 'enjoying', 'a', 'beautiful', 'day', 'outside', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'mother', 'and', 'her', 'blond', 'son', 'enjoying', 'her', 'a', 'sunny', 'sunny', 'sunny', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'woman', 'holding', 'a', 'bowl', 'of', 'food', 'in', 'a', 'kitchen', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'woman', 'is', 'a', 'kitchen', 'a', 'kitchen', 'a', 'kitchen', 'food', '.', 'kitchen', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'

Figure 4: Transformer (encoder only) translations

### 3.11 Full Transformer Translation Results

Input sentence	Back translation
'<sos>', 'a', 'man', 'in', 'an', 'orange', 'hat', 'starring', 'at', 'something', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'with', 'an', 'orange', 'hat', 'is', 'welding', 'something', '.', '\n', '<eos>', '.', '\n', '<eos>', '.', '\n', '<eos>, .'
'<sos>', 'a', 'boston', 'terrier', 'is', 'running', 'on', 'lush', 'green', 'grass', 'in', 'front', 'of', 'a', 'white', 'fence', '.', '\n', '<eos>', '<pad>'	'<sos>', 'a', 'lone', 'puppy', 'running', 'over', 'a', 'white', 'fence', 'in', 'front', 'of', 'a', 'white', 'fence', '.', '\n', '<eos>', '\n', '<eos>'
'<sos>', 'a', 'girl', 'in', 'karate', 'uniform', 'breaking', 'a', 'stick', 'with', 'a', 'front', 'kick', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'girl', 'in', 'a', 'karate', 'uniform', 'is', 'jumping', 'with', 'a', 'ribbon', '.', '\n', '<eos>, .', '\n', '<eos>', '\n', '<eos>'
'<sos>', 'people', 'are', 'fixing', 'the', 'roof', 'of', 'a', 'house', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'people', 'are', 'fixing', 'the', 'roof', 'of', 'a', 'house', '.', '\n', '<eos>, .', '\n', '<eos>, .', '\n', '<eos>, .'
'<sos>', 'a', 'group', 'of', 'people', 'standing', 'in', 'front', 'of', 'an', 'igloo', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'group', 'of', 'people', 'standing', 'in', 'front', 'of', 'a', 'messy', 'glass', 'building', '.', '\n', '<eos>, .', '\n', '<eos>, .'
'<sos>', 'a', 'guy', 'works', 'on', 'a', 'building', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'guy', 'working', 'on', 'a', 'building', '.', '\n', '<eos>, .', '\n', '<eos>, .', '\n', '<eos>, .'
'<sos>', 'a', 'man', 'in', 'a', 'vest', 'is', 'sitting', 'in', 'a', 'chair', 'and', 'holding', 'magazines', '.', '\n', '<eos>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'in', 'a', 'vest', 'is', 'sitting', 'on', 'a', 'chair', 'holding', 'a', '<unk>', '.', '\n', '<eos>, .', '\n', '<eos>'
'<sos>', 'a', 'mother', 'and', 'her', 'young', 'song', 'enjoying', 'a', 'beautiful', 'day', 'outside', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'mother', 'and', 'her', 'son', 'enjoying', 'a', 'beautiful', 'day', 'outdoors', '.', '\n', '<eos>, .', '\n', '<eos>, .', '\n', '<eos>'
'<sos>', 'a', 'woman', 'holding', 'a', 'bowl', 'of', 'food', 'in', 'a', 'kitchen', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'woman', 'holding', 'a', 'bowl', 'of', 'food', 'in', 'a', 'kitchen', '.', '\n', '<eos>, .', '\n', '<eos>, .', '\n', '<eos>'

Figure 5: Full transformer (best) translations

### 3.12 Seq2Seq (Best Model) Translation Results

Input sentence	Back translation
'<sos>', 'a', 'man', 'in', 'an', 'orange', 'hat', 'starring', 'at', 'something', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'in', 'a', 'orange', 'hat', 'is', 'something', 'something', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>, '<eos>'
'<sos>', 'a', 'boston', 'terrier', 'is', 'running', 'on', 'lush', 'green', 'grass', 'in', 'front', 'of', 'a', 'white', 'fence', '.', '\n', '<eos>', '<pad>'	'<sos>', 'a', 'white', 'white', 'dog', 'runs', 'white', 'dog', 'runs', 'in', 'a', 'white', '\n', 'white', '\n', 'white', '\n', 'white', '\n', 'white'
'<sos>', 'a', 'girl', 'in', 'karate', 'uniform', 'breaking', 'a', 'stick', 'with', 'a', 'front', 'kick', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'girl', 'in', 'a', 'karate', 'karate', 'to', 'a', 'to', 'a', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'people', 'are', 'fixing', 'the', 'roof', 'of', 'a', 'house', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'people', 'are', 'moving', 'moving', 'pictures', 'of', 'of', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'group', 'of', 'people', 'standing', 'in', 'front', 'of', 'an', 'igloo', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'group', 'of', 'people', 'are', 'standing', 'in', 'front', 'a', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'guy', 'works', 'on', 'a', 'building', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'guy', 'is', 'working', 'on', 'a', 'building', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'man', 'in', 'a', 'vest', 'is', 'sitting', 'in', 'a', 'chair', 'and', 'holding', 'magazines', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'man', 'in', 'a', 'sitting', 'on', 'a', 'a', 'a', 'a', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>'
'<sos>', 'a', 'mother', 'and', 'her', 'young', 'song', 'enjoying', 'a', 'beautiful', 'day', 'outside', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'mother', 'and', 'enjoying', 'enjoying'
'<sos>', 'a', 'woman', 'holding', 'a', 'bowl', 'of', 'food', 'in', 'a', 'kitchen', '.', '\n', '<eos>', '<pad>', '<pad>', '<pad>', '<pad>'	'<sos>', 'a', 'woman', 'mixing', 'a', 'kitchen', 'in', 'a', 'in', 'a', 'a', '.', '\n', '<eos>', '<eos>', '<eos>', '<eos>', '<eos>'

Figure 6: Seq2Seq (best) translations

### 3.13 Compare Transformer (Encoder Only) to Full Transformer

The transformer encoder performed well when compared to Seq2Seq models. However, it lagged the full transformer significantly in both loss (twice that of full) and perplexity (nearly five times full). The transformer encoder was, however, markedly faster to train with a runtime of approximately six minutes while the best full transformer took 56 minutes to train on my CPU.

The difference between the transformer encoder and full transformer can be seen in their translations in *Figures 4 and 5*, respectively. While the full transformer comes very close on many

translations and appears to understand the relationship between the words in the sequence, the transformer encoder focuses on subjects/nouns and some descriptive adjectives. For example, the full transformer translation ‘a woman holding a bowl of food in a kitchen’ is translated as ‘a woman is a kitchen a kitchen a kitchen food. kitchen’ by the transformer encoder. This is a result of the transformer encoder not possessing the means to learn the associations between the terms via a decoder’s self-attention mechanism. Instead, the transformer encoder learns high-level features of the text, such as ‘woman’, ‘bowl’, ‘food’ and ‘kitchen’ in the previous example.

As a result, the transformer encoder could be a reliable tool for generating features quickly, but pales in comparison to the full transformer for translation tasks.

### 3.14 Compare Seq2Seq to Transformer (Best Models)

The best full transformer performs much better than the best Seq2Seq model. Like the transformer encoder, the best Seq2Seq had a loss and perplexity nearly two and five times that of the best full transformer, respectively. The Seq2Seq model took roughly half the time to train (26 minutes), but was also significantly harder to train and required many more runs in order to find tunings that improved the model.

As mentioned previously, the best full transformer performed relatively well from a human perspective on the translation task, while the translations of the best Seq2Seq model are nearly nonsensical. Using the same example as the previous section, Seq2Seq produced the translation ‘a woman mixing a kitchen in a in a a.’. Like the transformer encoder, Seq2Seq is unable to grasp the associations between the words even though it is capable of identifying the main subjects in the text. Even so, it fails to produce as many subjects as the transformer encoder, which perhaps indicates that the model’s attention mechanism does not have the same capacity as the full transformer’s self-attention layer and cannot attend to the same level of features.

I can see an LSTM model being deployed in a resource-constrained environment to provide ”good enough” translations or light feature extraction. However, when unconstrained it is clear the full transformer is the superior model.

## References

- [1] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. ArXiv. /abs/2005.14165