Language Modelling with Recurrent Neural Networks: RNN vs. LSTM

I. Introduction

Language modelling estimates the probability of the next token given its preceding context and is a fundamental task in NLP. While modern systems rely on large Transformers, compact sequence models such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) remain useful for studying modelling choices, stability, and efficiency.

This project trains word-level models on a single-domain narrative corpus with three goals: (i) compare RNN and LSTM architectures; (ii) evaluate models both quantitatively (loss, perplexity) and qualitatively (temperature-controlled text generation); and (iii) conduct ablation studies on dropout and context length to isolate their effects.

We follow a standard experimental protocol with an 80/10/10 split, gradient clipping, early stopping, and learning-rate scheduling.

II. Dataset

We used a publicly available *Harry Potter* text file, treated as a continuous narrative for word-level next-token prediction. Minimal preprocessing included collapsing whitespace, retaining alphanumerics and periods, and converting text to lowercase.

Sentences were tokenized with NLTK and wrapped with <sos> and <eos> markers. The vocabulary was built from the training set only, with reserved tokens <pad>, <unk>, <sos>, and <eos>; unseen tokens in validation and test were mapped to <unk>. The data was split into 80% training, 10% validation, and 10% test. For training, fixed-length windows of 128 tokens were extracted with stride equal to the sequence length to create non-overlapping batches.

III. Models

a. Architectures

We implemented two recurrent baselines for word-level language modelling: a vanilla Recurrent Neural Network (RNN) and a Long Short-Term Memory network (LSTM). Both follow the same structure: an embedding layer, stacked recurrent layers, a linear projection to the vocabulary size, and a SoftMax output layer for next-token prediction.

The RNN updates its hidden state at each time step by combining the current input embedding with the previous hidden state. It is computationally efficient but prone to vanishing and exploding gradients, which limit its ability to capture long-range dependencies. The LSTM addresses this limitation with gating mechanisms (input, forget, and output gates) that regulate information flow, enabling it to retain relevant context over longer sequences and achieve more robust text generation.

Unless otherwise stated in ablation studies, both models used an embedding size of 256, hidden size of 256, two recurrent layers, dropout of 0.2, sequence length of 128, and gradient clipping at 1.0. The Adam optimizer with a learning rate of 3e-4 was applied. Ablation experiments focused on the LSTM, varying dropout (0.0 vs. 0.2) and context length (128 vs. 256) to assess their impact.

b. Training

Both models were trained under identical conditions to ensure fair comparison. Cross-entropy loss was used as the training objective, with perplexity reported as the evaluation metric. Gradient clipping at 1.0 was applied to stabilize updates.

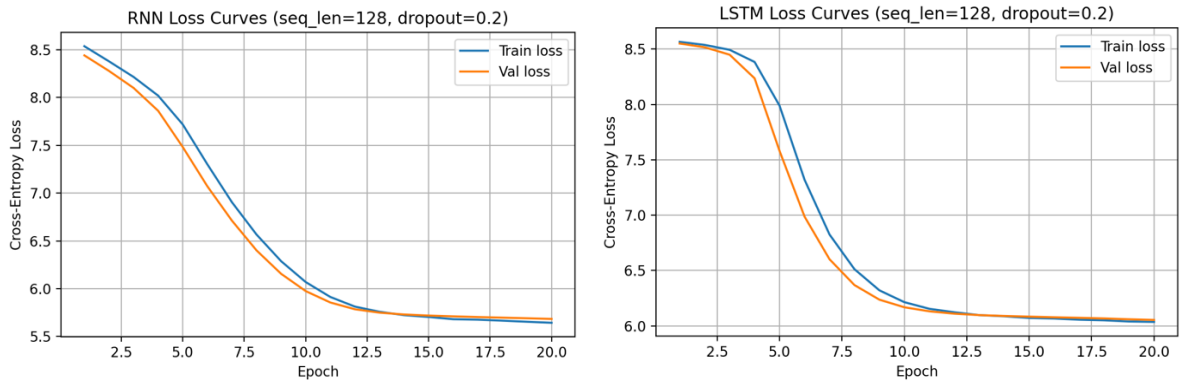
To further improve training stability, two additional techniques were employed:

- Learning rate scheduling: ReduceLROnPlateau reduced the learning rate when validation loss plateaued.
- Early stopping: Training was halted if validation loss did not improve for several epochs, and the best checkpoint was restored.

Training was performed with batch size 64, shuffling at each epoch. Validation was run at the end of every epoch, and the best-performing checkpoint was evaluated on the test set. For qualitative analysis, text generation was conducted using sampling with temperatures of 0.7, 1.0, and 1.3.

IV. Results & Analysis

a. Model Comparison: RNN vs LSTM



Both models converged stably, as shown in the loss curves. The LSTM required slightly longer to train (1.9 minutes) compared to the RNN (1.3 minutes), owing to its more complex gating structure. However, both models reached low cross-entropy loss values by epoch 20, with validation curves closely following training curves, indicating limited overfitting.

In terms of quantitative evaluation, the RNN achieved a lower test perplexity (298.18) compared to the LSTM (428.92). This is somewhat surprising, as LSTMs are typically expected to outperform vanilla RNNs on longer sequences. A likely explanation is the small dataset size and relatively short sequence length (128 tokens), which may reduce the advantage of gating mechanisms. In this setting, the simpler RNN may generalize better by avoiding over-parameterization.

Qualitative generation further highlights the trade-offs. At temperature 0.7, both models produce highly repetitive outputs dominated by <sos> and <eos> tokens, reflecting conservative sampling. At temperature 1.0, outputs become more coherent: for example, the RNN generates sentences with plausible structure (*"harry looked at ron and said wizards pulled both confused…"*), while the LSTM produces slightly more diverse but also fragmented continuations. At temperature 1.3, both models generate highly varied text, though coherence deteriorates; the LSTM introduces more fantastical or disjoint imagery (*"hedwig dirty no or he them shadowy poker mood…"*), while the RNN tends toward long, run-on sentences with occasional syntactic plausibility.

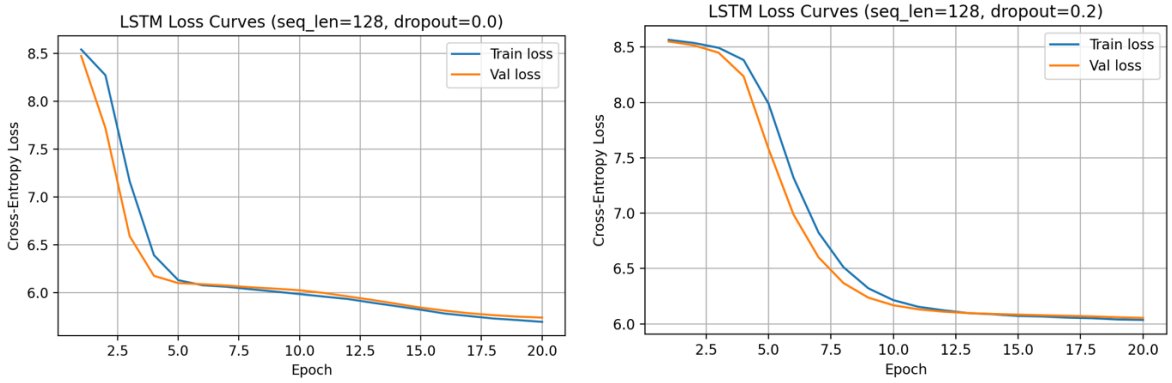
Overall, these results suggest that in this small-scale setup, the RNN offers better perplexity and faster training, while the LSTM provides somewhat richer but less stable generations at higher temperatures. The comparison highlights that architectural advantages of LSTMs may only emerge more clearly in larger datasets or with longer context lengths, which will be further examined in ablation studies.

b. Ablation Studies: LSTM

In addition to comparing architectures, we conduct within-model ablation studies on the LSTM. Specifically, we vary dropout (0.0 vs 0.2) to assess the effect of regularization and context length (128 vs 256) to evaluate the impact of longer dependencies. For consistency, all qualitative generations in these ablations are sampled at temperature T = 1.0, allowing differences to be attributed directly to

hyperparameter changes rather than sampling variance. These experiments help isolate how individual factors influence perplexity, stability, and training efficiency.

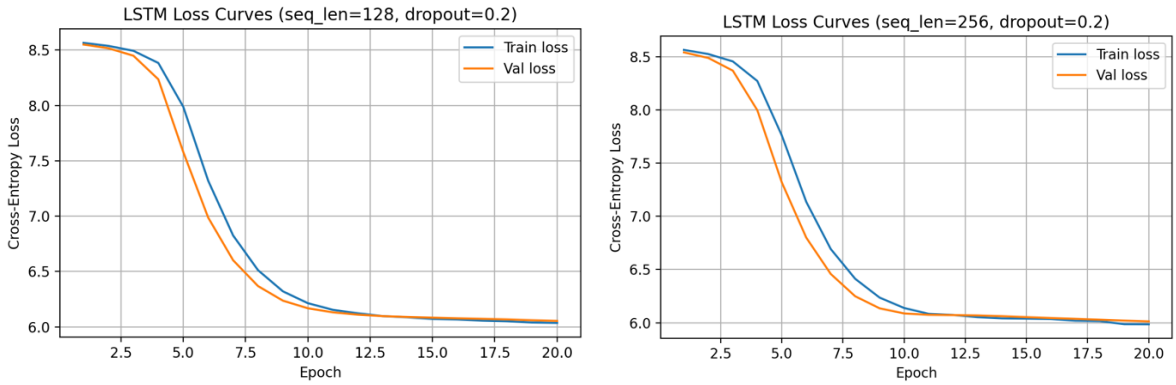
i. Dropout (0.0 vs 0.2)



The effect of dropout on LSTM performance is shown above. With no dropout (0.0), the model trained for 2.0 minutes and achieved a lower test perplexity (313.16), indicating stronger fit to the data. However, the validation loss curve suggests a greater risk of overfitting. In contrast, applying dropout 0.2 reduced training time to 1.0 minute but resulted in a higher test perplexity (482.25). This suggests that while dropout helps regularize training, in this small-scale setup it may underutilize model capacity, leading to weaker generalization compared to the no-dropout condition.

At dropout = 0.0, the model produces longer and sometimes more coherent continuations, often weaving in plausible characters or events (e.g., “ollivander,” “hermione,” “hagrid”). However, it also shows signs of overfitting, with irregular phrasing and abrupt topic shifts. With dropout = 0.2, generations are shorter and more fragmented, frequently recycling structural markers (<sos>, <eos>). This supports the quantitative results, where dropout reduced training time but increased perplexity, suggesting that stronger regularization limited the model’s expressive capacity in this small-scale setting.

ii. Context length (128 vs 256)



The impact of context length is illustrated in above figures. With a sequence length of 128, training completed in 1.0 minute, yielding a test perplexity of 482.25. Extending the context to 256 tokens increased training time to 1.8 minutes but improved generalization, lowering test perplexity to 416.00. Both models converged smoothly, but the longer context enabled the LSTM to capture additional dependencies, resulting in modest gains in performance at the cost of higher computation.

At sequence length 128, the model produces short continuations that are often fragmented and repetitive, with limited long-range structure. Increasing the context to 256 tokens yields longer and more elaborate

continuations, incorporating more varied entities (e.g., “malfoy,” “mcgonagall,” “hogwarts”). However, coherence remains limited, and the outputs sometimes drift into disjointed fragments. These qualitative differences align with the perplexity results, where the longer context improved test PPL (416.00 vs. 482.25), suggesting that additional context helps capture richer dependencies at the cost of slower training.

V. Conclusion

This report compared RNN and LSTM models for word-level language modeling on a single-domain narrative corpus. Both architectures trained stably, but the RNN achieved faster training and lower test perplexity, while the LSTM produced more diverse generations at higher temperatures. These findings suggest that in small-scale settings with limited context, the simpler RNN can generalize better, whereas the advantages of LSTMs may only emerge with larger datasets or longer dependencies.

Ablation studies on the LSTM further highlighted the role of hyperparameters. Removing dropout improved perplexity but increased signs of overfitting, while applying dropout led to shorter, less coherent generations. Extending context length from 128 to 256 tokens reduced perplexity and enriched generations, albeit with higher computational cost. Together, these results show how architectural and hyperparameter choices shape both quantitative performance and qualitative behaviour.

Overall, the experiments demonstrate the importance of balancing model complexity, regularization, and context length when designing recurrent language models. While RNNs remain competitive under constrained conditions, LSTMs offer flexibility for capturing richer dependencies, provided sufficient data and compute resources are available.

VI. Appendix

a. Generation Samples

i. Model Comparison: RNN vs LSTM

1. Prompt 1: <sos> harry looked at ron and said

Temperature	RNN Output	LSTM Output
T = 0.7	the the dudley the on i . <eos> …	the and <sos> around <eos> …
T = 1.0	wizards pulled both confused right of to knocked …	. <sos> start <sos> dudley because goyle full thought …
T = 1.3	four the bewitch romania breath on at confused …	witch at said up by stared a next didn . gargoyles him …

2. Prompt 2: <sos> dumbledore whispered

Temperature	RNN Output	LSTM Output
T = 0.7	room the was yer harry a it the harry t on he …	wand had s <sos> <eos> hagrid the all <eos> …
T = 1.0	looked pointed eyes stool nine edge …	cold harry sc on probably apart as i sweet curious was …
T = 1.3	you made but talk cloak solutions great s t too tall …	hedwig dirty no or he them shadowy poker mood quiet …

3. Prompt 3: <sos> the castle was

Temperature	RNN Output	LSTM Output
T = 0.7	. <eos> <sos> he did the . <eos> <sos> she …	then s to . <sos> was the <sos> <sos> <eos> …

T = 1.0	deal . <eos> <sos> listen angry slippery bell …	fact our the why <sos> close whenever his edge his go …
T = 1.3	rock to a triumph careful catch poster mother game …	attractive breaking forces plowed eggs her more as he harry …

ii. Ablation Studies: LSTM

1. Dropout (0.0 vs 0.2), T=1.0

Prompt	Dropout = 0.0	Dropout = 0.2
<sos> harry looked at ron and said	harry looked at ron and said s mum get <sos> was never suppose ollivander t …	harry looked at ron and said . <sos> start <sos> dudley because goyle full thought …
<sos> dumbledore whispered	dumbledore whispered from and lot was to polite realize a even and said …	dumbledore whispered cold harry sc on probably apart as i sweet curious was …
<sos> the castle was	the castle was life to and ve wave there alley t his in go except great if with should hadn entering was him hermione yeh …	the castle was fact our the why <sos> close whenever his edge his go <eos> …

2. Context Length (128 vs 256), T=1.0

Prompt	SeqLen = 128	SeqLen = 256
<sos> harry looked at ron and said	harry looked at ron and said . <sos> start <sos> dudley because goyle full thought …	harry looked at ron and said s hundred malfoy he you a look of as took to managed marched fluffy …
<sos> dumbledore whispered	dumbledore whispered cold harry sc on probably apart as i sweet curious was …	dumbledore whispered hand to glasses their swooping his the right need <sos> dog this own jar quickly …
<sos> the castle was	the castle was fact our the why <sos> close whenever his edge his go …	the castle was <sos> just by too s you the dudley her at . and or smiled that ever dived …