A Comparative Analysis of Recurrent and Transformer-Based Architectures for Image Captioning on a Constrained Dataset

Yige Huang
Department of Electrical and Computer Engineering
University of Toronto
yige.huang@mail.utoronto.ca

Yanrong Xiao
Department of Electrical and Computer Engineering
University of Toronto
yanrong.xiao@mail.utoronto.ca

Kaiwen Zhu
Department of Electrical and Computer Engineering
University of Toronto
kaiwin.zhu@mail.utoronto.ca

Abstract

This report presents a comparative analysis of three prominent neural architectures for image captioning on the constrained Flickr8k dataset: a Long Short-Term Memory (LSTM), a Gated Recurrent Unit (GRU), and a Transformer. All models used a pretrained ResNet-50 encoder. The recurrent models were single-layer, while the Transformer employed a 6-layer decoder stack. Performance was quantitatively evaluated using the Bilingual Evaluation Understudy (BLEU) metric. The LSTM achieved the highest score (0.2316), followed closely by the GRU (0.2266), and the Transformer achieved the peak BLEU of 0.2061. These results demonstrate that on a small dataset, simpler recurrent architectures provide a more robust solution, as the high capacity of the Transformer and its sensitivity of hyperparameter tuning may lead to poor generalization while having higher computational costs as well.

Assentation of Teamwork

The project was a collaborative effort with each member taking primary responsibility for one of the core models. Yige Huang implemented and trained the GRU model, Yanrong Xiao was responsible for the LSTM model, and Kaiwen Zhu handled the Transformer model. All members also performed quality assurance and contributed to the final analysis and report.

1 Introduction

Automatic image captioning is a fundamental challenge at the intersection of computer vision and natural language processing [1; 2; 3]. The task, which requires a machine to generate a human-like descriptive sentence for an image, is significantly more complex than traditional vision tasks like image classification. A successful captioning system must not only identify the objects present in an

image but also comprehend their attributes, their spatial and semantic relationships, and the actions they are involved in, and then articulate this complex understanding in a syntactically correct and fluent natural language sentence.

The dominant architectural framework for tackling this multimodal task is the encoder-decoder paradigm, a concept largely inspired by successes in machine translation [1; 4]. In this framework, an encoder network, typically a deep Convolutional Neural Network (CNN), processes the input image and compresses its salient visual information into a rich, fixed-length vector representation. This vector then conditions a decoder network, an autoregressive language model, which generates the caption one word at a time, with each new word being conditioned on the image representation and the previously generated words [1; 5]. This project adopts this well-established paradigm to structure its models.

The central objective of this project is to design, implement, and critically evaluate three distinct autoregressive language models for image captioning: one based on a Long Short-Term Memory (LSTM) network, a second on a Gated Recurrent Unit (GRU), and an attention-based Transformer architecture. The scope of the project involves leveraging a pretrained ResNet-50 as the vision encoder, training the models on the small-scale Flickr8k dataset, and assessing performance using the standard BLEU evaluation metric [5].

2 Preliminaries and Problem Formulation

The field of image captioning has evolved rapidly, moving from recurrent models to attention-based systems. The models in this project represent key milestones in this progression.

The foundational "Show and Tell" model by Vinyals et al. (2015) established the viability of the CNN-RNN architecture [1; 2; 6]. This approach uses a CNN to encode an image into a single feature vector, which then initializes the hidden state of a Long Short-Term Memory (LSTM) network that decodes it into a caption [1; 4]. The Gated Recurrent Unit (GRU), introduced by Cho et al. (2014), offers a streamlined alternative to the LSTM with a simpler gating mechanism that often yields comparable performance [7]. The LSTM and GRU models in this project are direct descendants of these recurrent architectures.

A key limitation of these initial models was their reliance on a single, static image vector, creating an information bottleneck. The "Show, Attend and Tell" model by Xu et al. (2015) introduced a visual attention mechanism to address this [2]. This allowed the decoder to dynamically focus on different spatial regions of the image while generating each word, improving performance and interpretability [2].

The 2017 paper "Attention Is All You Need" by Vaswani et al. marked a paradigm shift by introducing the Transformer [3]. This architecture dispenses with recurrence entirely, relying solely on self-attention mechanisms to model dependencies. This design captures long-range relationships more effectively and allows for massive parallelization during training, making it the standard for many NLP tasks [3]. The Transformer model in this project is a direct application of this architecture. This project's comparison of the classic recurrent approaches against the modern Transformer highlights the importance of considering factors beyond architectural novelty, such as data availability and model capacity.

3 Design

The core of this comparative study lies in the implementation of three distinct decoder architectures. The specific hyperparameters for each model are detailed in the Appendix (Table 2).

3.1 Model I: Recurrent Neural Network (LSTM) Decoder

The first model implements a recurrent decoder based on Long Short-Term Memory (LSTM) units. The RNN decoder module consists of an embedding layer, a single LSTM layer, and a final fully connected (linear) layer that projects the LSTM's output hidden state to a logit distribution over the entire vocabulary.

The LSTM, a specialized type of RNN, is designed to manage long-term dependencies through a series of gating mechanisms. These gates control the flow of information into and out of the cell state, enabling the model to selectively remember or forget past information.

The image feature vector from the ResNet-50 encoder is projected to initialize the LSTM's initial hidden and cell states, providing the visual context for the captioning process [1].

The hidden state initialization method was chosen over the prepend feature method for the final LSTM model because it directly primes the LSTM's internal hidden and cell states with the image features, providing a more robust and persistent visual context than simply prepending the feature as a token to the input sequence..

3.2 Model II: Gated Recurrent Unit (GRU) Decoder

The Gated Recurrent Unit (GRU) is a streamlined variant of recurrent neural networks that balances modeling capacity with computational efficiency. Its architecture combines the memory cell and hidden state into a single vector, eliminating the separate cell state used in LSTMs. Instead, it uses two key gating mechanisms: the update gate, which determines how much of the past information should be carried forward, and the reset gate, which controls how much of the previous state should be forgotten when incorporating new input.

In this project's GRU decoder, an embedding layer first converts input word tokens into dense vector representations. The GRU layer then processes these embeddings sequentially, maintaining a hidden state that evolves over time as each new word is generated. At each time step, the hidden state integrates the linguistic context from earlier words with the visual context provided by the image features, which are used to initialize the decoder's hidden state at the start of caption generation.

The output from the GRU at each step is passed through a fully connected layer to produce a probability distribution over the vocabulary, allowing the model to select the most likely next word.

3.3 Model III: Transformer-Based Decoder

The third model uses a Transformer decoder, which is configured with multiple decoder layers, each containing several attention heads. The image feature vector is projected to the model dimension and serves as the static memory input to the decoder stack. The decoder employs masked multi-head self-attention to process the generated text prefix and multi-head cross-attention to ground the generated words in the visual content. The Transformer architecture dispenses with recurrence entirely, relying solely on self-attention mechanisms to model dependencies [3].

This design captures long-range relationships more effectively and allows for massive parallelization during training, making it the standard for many NLP tasks. The hyperparameter choices reveal a fundamental disparity in model capacity between the recurrent models and the Transformer, which is a critical factor in understanding their respective performance on the limited Flickr8k dataset.

4 Methodology

The project utilizes the Flickr8k dataset, a widely used benchmark for image captioning researc [4]. The dataset contains 8,092 images, each paired with five distinct, human-annotated reference captions. Its manageable size makes it well-suited for training and iteration on a local machine.

A consistent preprocessing pipeline was applied to both image and text data. All images are resized and normalized using standard ImageNet mean and standard deviation values. The caption data undergoes tokenization, vocabulary creation with a frequency threshold, addition of special tokens, and padding to create uniformly sized tensors for batch processing.

A pretrained ResNet-50 model serves as the common vision encoder for all pipelines. The final fully-connected classification layer is removed, and the output from the preceding convolutional blocks is processed to yield a feature vector for each image. A critical decision was to keep the vision encoder frozen during the training of all decoders by disabling gradient calculations for its parameters. This strategy dramatically reduces the number of trainable parameters and the overall computational load. Furthermore, the recurrent model implementations employed pre-extracting and caching all image features to disk before training began, significantly accelerating the training cycle.

5 Numerical Experiments

All models were trained using the Adam optimizer and the Cross-Entropy Loss function. The quality of the generated captions is quantitatively evaluated using the Bilingual Evaluation Understudy (BLEU) score [5]. BLEU measures the n-gram precision between a generated caption and the set of reference captions. The evaluation was performed on the validation set (810 images). All models used beam search algorithm with beam size = 5.

5.1 Including Tables

Table 1: BLEU Scores

Metric	LSTM Model	GRU Model	Transformer Model
Avg. BLEU (Epoch 20)	0.2080	0.2213	0.1773
Avg. BLEU (Peak)	0.2316 (Epoch 9)	0.2266 (Epoch 12)	0.2061 (Epoch 12)

Figures illustrating the BLEU scores per epoch can be found in the Appendix.

6 Discussion

The empirical results from this comparative study demonstrate that the single-layer LSTM model performance is very close to the single layer GRU model, both outperforming the 6-layer Transformer models on the constrained Flickr8k dataset. As shown in Table 1, the LSTM model has the BLEU score peaking at Epoch 13 at 0.2316, and 0.2080 on Epoch 20. The Transformer model achieved an average BLEU score of 0.2061 at its peak (at Epoch 5) and 0.1773 at Epoch 20. While the GRU model had a final BLEU-4 score of 0.2213 and a peak of 0.2266. Contrary to the typical performance trends observed on large-scale datasets, the recurrent models (LSTM and GRU) consistently outperformed the Transformer. This section discusses the likely reasons for these findings and highlights the critical factors at play.

6.1 The Effectiveness of Recurrent Models on Constrained Data

The superior performance of the LSTM and GRU models on this task can be attributed to their architectural efficiency and suitability for a small, specialized dataset. The Flickr8k dataset, with its limited vocabulary and relatively short, descriptive captions, does not contain the long-range dependencies that often necessitate more complex models like the Transformer.

- **Architectural Simplicity**: Both LSTM and GRU models are fundamentally simpler than a multi-layer Transformer. Their parameters are primarily focused on learning the sequential flow of language.
- **Data Efficiency**: With fewer parameters to train than a 6-layer Transformer, the recurrent models were less prone to overfitting. The 8,000 images in the Flickr8k dataset provided sufficient examples for the LSTM and GRU to learn meaningful patterns without memorizing the training data.

The recurrent models' peaked performance at an early epoch and sustained high scores demonstrate their robust ability to generalize effectively from the limited data provided.

6.2 Challenges with the Transformer Architecture

The underperformance of the 6-layer Transformer model, which achieved a lower peak BLEU-4 score points to a classic mismatch between model capacity and dataset size.

- **High Parameter Count**: A 6-layer Transformer is a computationally expensive and data-hungry model. Its self-attention mechanism, while powerful, requires a vast amount of data to learn meaningful relationships between words. The small size of the Flickr8k dataset was likely insufficient to train this model effectively.

- **High Computational Expense**: The parallelizable nature of the Transformer's self-attention mechanism, while optimized for GPU-accelerated training on massive datasets, proved to be a liability here. The computationally expensive nature of the model necessitated the use of a GPU, whereas the recurrent models (LSTM and GRU) could be efficiently trained on a CPU in approximately 7 to 10 minutes per epoch.
- **Difficulty in Hyperparameter Tuning**: Transformer models are highly sensitive to hyperparameters, such as learning rate schedules and warmup steps. The observed performance drop after Epoch 5 strongly suggests that the model began to overfit the training data or that the learning rate was not optimally managed. The model's complexity makes it difficult to tune for a small dataset, a problem not as pronounced in the simpler recurrent architectures.
- **Potential for Overfitting and Poor Generalization**: All models in this study began to show signs of overfitting around the same epoch numbers as shown in Figure 1, 2, and 3 in the Appendix. However, the Transformer model's validation loss plateaued at a higher value than the recurrent models, indicating that it was less effective at learning generalizable features from the data. The large number of parameters in the Transformer, relative to the limited training data, made it highly susceptible to memorizing the captions and failing to generalize, even when the recurrent models were still performing well.

6.3 Feasible adjustment methods

To improve the BLEU score of the Transformer-based image captioning model compared to the LSTM baseline, several optimization strategies can be applied. First, enhance generalization and stability with label smoothing, mixed precision training, and gradient clipping. Second, adjust beam search using a suitable beam size and length penalty, ensuring consistent token handling and optionally experimenting with nucleus or top-k sampling. Third, maintain consistency in BLEU evaluation by using the same tokenizer and preprocessing as the baseline, applying smoothing functions, and evaluating on a sufficiently large validation set. Fourth, improve the model architecture by adding dropout, ensuring positional encoding length coverage, and tuning attention head counts. Fifth, refine hyperparameters through learning rate scheduling, adjusting batch size, and setting a minimum epoch threshold before early stopping. Finally, ensure reproducibility by saving and loading the vocabulary, preventing unnecessary retraining across sessions.

The findings indicate that while the Transformer is a state-of-the-art architecture for large-scale language modeling tasks, its immense capacity is a liability on small datasets. For this project, the data was not complex enough to justify the computational and parametric overhead of a multi-layer Transformer.

7 Conclusions

This project successfully implemented and compared three distinct architectures for image captioning on the Flickr8k dataset. The primary conclusion is that under the constraint of a small-scale dataset, the single-layer recurrent model architecture like the GRU and the LSTM was the most effective in terms of final performance. The Transformer model, despite its high capacity, had the lowest BLEU score.

The Transformer's lower performance is likely due to the small dataset size, limited training epochs, and restricted hyperparameter tuning, which doesn't suit its high capacity. In contrast, GRU and LSTM architectures are more robust and data-efficient on limited datasets. The key limitations of this study are the small-scale dataset, limited hyperparameter search, and a primary reliance on BLEU scores for evaluation. The Transformer's capability should not be underestimated—its architecture has demonstrated state-of-the-art results in numerous large-scale vision-language tasks. It is highly probable that the Transformer model's performance would surpass the recurrent models if all were trained on a much larger dataset with more extensive optimization.

Future work could involve systematically investigating the impact of model capacity by training variants of the GRU, LSTM, and Transformer models. Further research could also explore more advanced regularization techniques, fine-tuning the vision encoder, and incorporating a broader suite of evaluation metrics, such as METEOR and CIDEr, to provide a more holistic assessment of caption quality.

References

[1] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164).

[2] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International conference on machine learning* (pp. 2048-2057). PMLR.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. In *Advances in neural information processing systems* (Vol. 30).

[4] Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47, 853-899.

[5] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer.

[6] Vedantam, R., Zitnick, C. L., & Parikh, D. (2015). CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4566-4575).

[7] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Appendix

Hyperparameter Tables

Table 2: Comparative Hyperparameter Specification

Hyperparameter	LSTM Model	GRU Model	Transformer Model
Architecture			
Decoder Type	LSTM	GRU	Transformer Decoder
Embedding Size	512	512	512
Hidden/Model Dim (d_{model})	512	512	512
Num. Layers	1	1	6
Num. Attention Heads	N/A	N/A	8
Dropout	0.3	0.3	0.1
Training			
Optimizer	Adam	Adam	Adam
Learning Rate	1×10^{-4}	1×10^{-4}	1×10^{-4} (initial)
LR Scheduler	No	No	ReduceLROnPlateau
Batch Size	32	64	32
Num. Epochs	20	20	20 (with early stopping)
Vocabulary			
Freq. Threshold	5	5	5
Vocab Size	3005	3005	Not specified

Including Figures

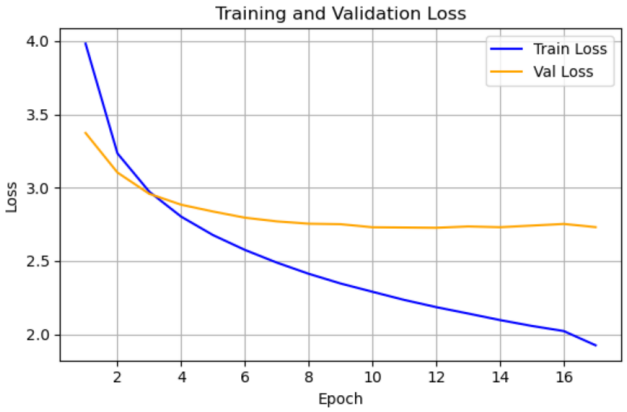

Figure 1: Training and Validation Loss for the Transformer model.

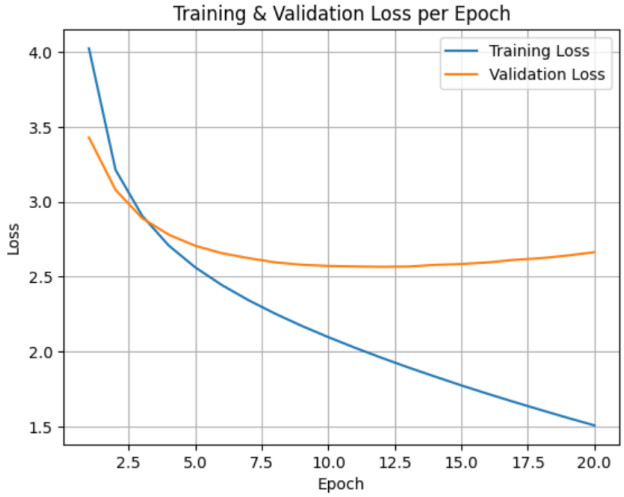

Figure 2: Training and Validation Loss for the LSTM model.

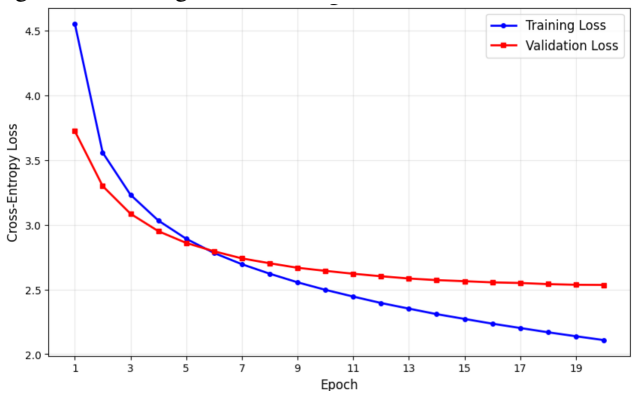

Figure 3: Training and Validation Loss for the GRU model.

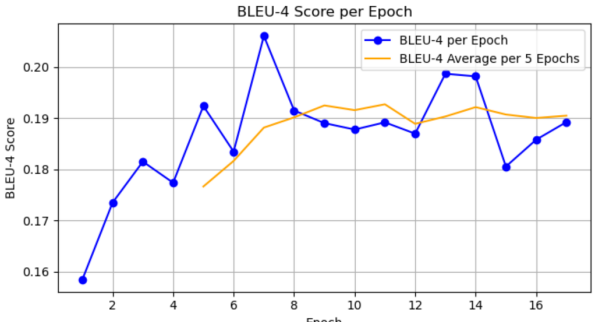

Figure 4: Training and Validation Loss for the Transformer model.

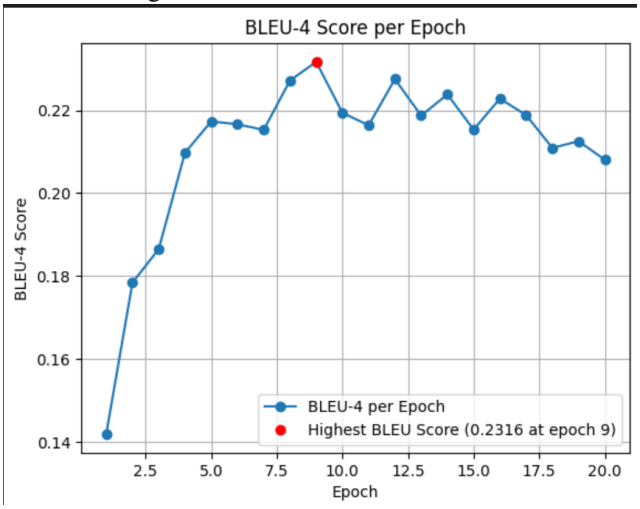

Figure 5: Training and Validation Loss for the LSTM model.

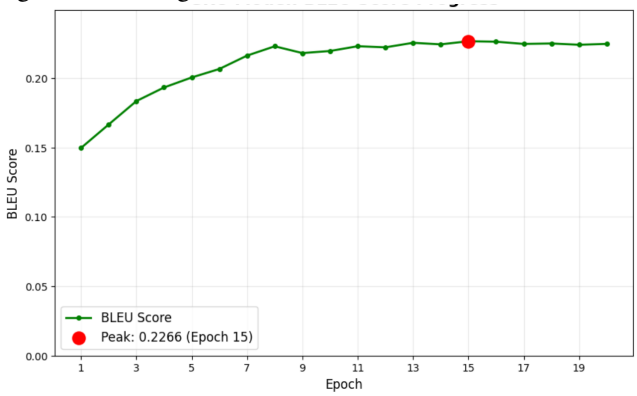

Figure 6: Training and Validation Loss for the GRU model.