Revisiting RNNs in the Age of Transformers: A Comparative Analysis of Data Dependence and Robustness

Xiangyu Tu[*]
Dept. of Computer Science
University of Toronto
xiangyu.tu@mail.utoronto.ca

Jiahao Xu[*]
Dept. of Computer Science
University of Toronto
mikeke.xu@mail.utoronto.ca

Zhuo Zhang[*]
Dept. of Computer Science
University of Toronto
zhuocd.zhang@mail.utoronto.ca

Abstract

Despite the widespread adoption of transformer architectures across various domains, their limitations in certain settings remain unresolved. Transformers excel at handling large-scale datasets and complex tasks, but their dependence on vast amounts of training data and their vulnerability to noisy inputs highlight persistent challenges. Meanwhile, recurrent neural networks (RNNs), though increasingly overshadowed, exhibit strengths in robustness under noisy conditions and efficiency in low-data regimes. This paper conducts a systematic comparison between RNNs and transformers, focusing on tasks of varying complexity—sentiment classification of IMDB movie reviews and English-to-French translation. By analyzing the effects of data size and quality, as well as key model parameters, we aim to uncover complementary strengths and identify potential synergies between these architectures. Our results provide actionable insights for leveraging RNN and transformer properties to improve model design and inform practical applications.

1 Introduction

The transformer architecture has revolutionized the field of artificial intelligence, achieving state-of-the-art results in natural language processing, computer vision, and beyond. However, the rapid rise of transformers has not been without challenges. Core issues such as the quadratic complexity of their attention mechanism[6], high memory consumption[14], and difficulties with interpretability[12] remain prominent. However, the most pressing challenges involve transformers' significant dependence on large datasets for effective training and their vulnerability in low-resource or noisy data environments[2]. While various adaptations, such as optimizing deeper transformers on small datasets through regularization and tailored architecture modifications[15], and employing few-shot learning techniques with noisy labels[8], have attempted to address these data-related issues, they often fall short in practical applications with limited or noisy data, underscoring the need for further investigation.

In contrast, recurrent neural networks (RNNs) have seen a decline in popularity but still retain unique advantages. Their inherent sequential structure provides robustness in noisy data settings and

[*] Alphabetical order. Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

better generalization in low-data regimes. Several RNN frameworks are particularly well-suited to addressing these problems, leveraging their sequential processing to handle noisy inputs effectively and requiring less data to achieve strong performance[10]. RNNs also benefit from a more compact architecture, making them computationally less demanding in resource-constrained environments. These characteristics prompt a reconsideration of the dichotomy between RNNs and transformers, particularly in situations where transformers' limitations may hinder their performance.

This paper seeks to bridge the gap by revisiting the comparison between RNNs and transformers through a focused lens. We aim to address two critical areas: dependence on large datasets and robustness in noisy data. Our experimental framework spans two tasks of differing complexity—IMDB sentiment classification and English-to-French translation—allowing us to evaluate performance across simple and challenging domains. Through controlled experiments involving data augmentation, quality variations, and parameter tuning, we explore the distinct strengths of each architecture and their implications for practical applications. By situating this exploration within the context of modern challenges in machine learning, this study aims to provide actionable insights for optimizing neural network design and addressing data-related issues in diverse tasks.

2 Related Works

The growing body of research comparing RNNs and transformers highlights the strengths and weaknesses of these architectures across various domains. For instance, transformers excel in speech-related tasks due to their capacity to model global dependencies, whereas RNNs perform well in scenarios requiring fine-grained temporal modeling, as demonstrated in a comparative study on speech applications [5]. However, such comparisons generally overlook the impact of noisy or low-resource data.

Theoretical explorations, such as those examining recurrence versus attention in human sentence processing, provide insights into the fundamental capabilities of these models but lack practical relevance to data constraints [4]. Similarly, studies like [1] analyze bottlenecks in RNNs for retrieval tasks, yet their focus diverges from our emphasis on noisy and limited data settings.

Other works address data challenges within individual models. For example, research on few-shot learning with noisy labels highlights methods to enhance transformer robustness in low-resource settings [8], while a comprehensive review of RNN applications discusses their inherent ability to handle noisy data and generalize effectively with limited samples [10]. While informative, these studies focus on single architectures rather than comparative analysis. By contrast, our work aims to fill this gap by systematically evaluating RNNs and transformers under these conditions, identifying complementary strengths and potential synergies.

3 Methodology

3.1 Overview of Experiments

This study conducts a comprehensive evaluation of RNNs and transformers across two representative tasks of varying complexity: sentiment classification and machine translation. The experiments are designed to systematically compare the strengths and weaknesses of these architectures, focusing on convergence speed, performance under varying data distributions and quality, and the impact of augmentation. By analyzing these factors in controlled settings, we aim to uncover complementary properties and actionable insights to inform the design of hybrid or task-specific neural architectures.

3.2 Experimental Setup

3.2.1 Datasets

The sentiment classification task involves determining whether a given IMDB movie review is positive or negative. We use the IMDB Large Movie Review Dataset [9] for this purpose. The dataset is preprocessed by tokenizing the reviews, converting all text to lowercase, and padding or truncating the sequences to a fixed length for uniformity. The model's performance on this task is evaluated using the Mean Squared Error (MSE) metric.

The machine translation task involves English-to-French sequence-to-sequence translation. The dataset used for this task is an open-source English-to-French dataset [7]. Preprocessing includes tokenizing both source and target texts and constructing vocabularies with a fixed maximum size. The evaluation metric for translation accuracy is the BLEU score, implemented as per the methodology described in this repository [3].

3.2.2 Recurrent Neural Networks (RNNs)

For sentiment classification, we use an open-source Long Short-Term Memory (LSTM) network [13], consisting of an embedding layer for token representation, LSTM layers for capturing temporal dependencies, and a fully connected layer for binary classification.

For machine translation, we adopt a GRU-based sequence-to-sequence (Seq2Seq) model [11], with GRU-based encoder and decoder components. The encoder processes input English tokens, while the decoder generates French output using GRU layers and a fully connected layer for vocabulary prediction.

3.2.3 Transformer Models

We implement a transformer model with task-specific configurations. The architecture includes an embedding layer, positional encoding, multi-head self-attention, feed-forward layers, and residual connections. For sentiment classification, we use a transformer with four attention heads and two layers, while for machine translation, we use eight heads and three layers. The encoder encodes input sequences, and the decoder generates outputs via cross-attention and autoregressive predictions.

3.3 Experiments

Convergence Speed: This experiment measures the number of epochs needed for RNNs and transformers to achieve stable validation loss on sentiment classification and English-to-French translation. Both tasks used the same preprocessing for their datasets. LSTM and transformer models were evaluated for sentiment classification, while GRU-based Seq2Seq and transformer models were applied to translation.

Architecture Sensitivity: This experiment examines how Transformers respond to task complexity and architectural changes. For both tasks, transformer models with varying attention heads were analyzed for their effect on accuracy. Results reveal the impact of task complexity on optimal configurations and model responses to parameter changes.

Data Quality and Distribution: This experiment evaluates the robustness of RNNs and transformers to imbalanced or varied data. Sentiment analysis used datasets with different proportions of positive and negative samples (e.g., 50% vs. 80% positive). For translation, short-to-long sentence ratios were varied similarly. Performance was assessed to identify each architecture's sensitivity to uneven distributions and its ability to generalize under noisy data conditions.

Dataset Augmentation: This experiment assesses the impact of data augmentation. Sentiment datasets were augmented using synonym replacement, while translation data involved paraphrasing sentences. RNN and transformer models retrained on these augmented datasets were compared to those trained on original data. Results highlight the benefits of augmentation in improving accuracy and robustness, especially in low-resource scenarios.

4 Results

4.1 Convergence Speed

Figure 1 shows that the RNN model converged faster than the transformer in Task 1 (sentiment classification), requiring only 6 epochs compared to 27. This can be attributed to the simplicity of the task and RNN's efficiency in capturing short-range dependencies, while transformers' attention mechanisms add overhead. The test accuracy difference (0.60 for RNN vs. 0.61 for transformer) is minor, as both effectively model the dataset, with transformers gaining a slight edge from their ability to capture global dependencies.

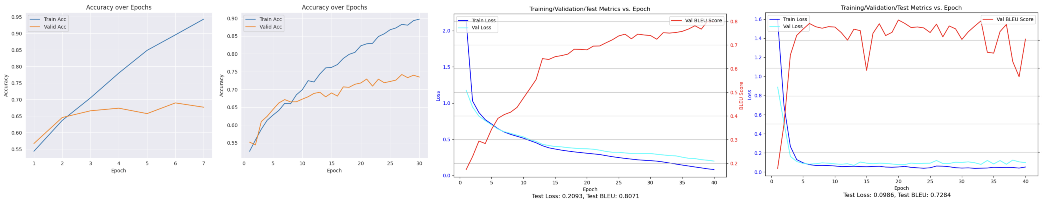

Figure 1: Training and validation metrics for classification and translation using RNNs and Transformers. The RNNs in leftmost and third plots and the Transformers in second and rightmost plots.

For Task 2 (English-to-French translation), the transformer converged in just 6 epochs, far quicker than the RNN's 40 epochs. Transformers' parallel computation and long-range dependency handling make them ideal for complex sequence-to-sequence tasks, unlike RNNs, which process sequentially and are slower. However, the RNN achieved a higher BLEU score (0.81 vs. 0.73), likely due to the smaller dataset favoring RNNs' generalization under low-resource conditions. These results underscore the need to match model choice with task complexity and dataset size for optimal outcomes.

4.2 Architecture Sensitivity

Table 1: Transformer Heads Comparison for Classification and Translation Tasks

Task	Heads	Accuracy (%)	Epochs	Task	Heads	BLEU Score	Epochs
Classification	1	60.91	27	Translation	4	0.68	18
Classification	4	61.97	27	Translation	8	0.76	5
Classification	10	61.92	27	Translation	16	0.72	11

The number of transformer heads influenced performance across both tasks, as shown in Table 1. For Task 1 (classification), varying the number of attention heads from 1 to 10 resulted in minimal accuracy improvement (60.91% to 61.97%), while the required epochs remained constant at 27. This limited sensitivity highlights that sentiment classification, a relatively simple task, does not benefit substantially from increased model complexity. The slight improvement with 4 heads suggests that moderate granularity in capturing attention is sufficient, and additional heads provide diminishing returns.

In contrast, Task 2 (translation) showed a stronger dependence on the number of heads. Increasing the heads from 4 to 8 improved the BLEU score from 0.68 to 0.76 while reducing the epochs needed for convergence from 18 to 5. However, further increasing the heads to 16 decreased the BLEU score to 0.72 and raised convergence time to 11 epochs. This non-linear trend suggests that while more heads enhance the model's capacity to capture complex dependencies, excessive heads can lead to overfitting or inefficient learning, particularly with smaller datasets. These findings underline the importance of carefully balancing head configurations with task requirements, offering valuable guidance for designing transformers tailored to specific applications.

4.3 Data Quality and Distribution

The models exhibited distinct responses to data quality and distribution changes. For Task 1 (classification), the RNN's accuracy improved steadily with increasing proportions of positive samples, plateauing at 0.60 with 40%-50% positive data, while the transformer showed no improvement until reaching 40%, where it achieved a slightly higher accuracy of 0.61. This suggests that transformers require a more balanced dataset to effectively leverage their capacity for global dependency modeling. For Task 2 (translation), the RNN achieved its highest BLEU score of 0.81 with 30%-40% long sentences but slightly dropped at 50%, likely due to overfitting. In contrast, the transformer demonstrated gradual improvement, reaching its highest BLEU score of 0.73 at 50%, reflecting its robustness in handling longer sequences with more balanced distributions. These trends emphasize the importance of aligning data distribution with the architecture's strengths for optimal performance.

4.4 Dataset Augmentation

Data augmentation was applied only to Task 1 (classification), as synonym replacement suits text classification tasks but is less applicable to machine translation, where semantic fidelity is critical. RNN performance showed negligible improvement with synonym replacement (0.60 accuracy for both 20% and 50% augmentation), indicating limited benefits from simple data augmentation. Similarly, the transformer exhibited minimal variation (0.66 and 0.65 for 20% and 50%, respectively), suggesting that synonym replacement did not significantly enhance its global dependency modeling. These results highlight that simplistic augmentation techniques may not effectively improve model performance for tasks where data diversity is not a primary constraint.

5 Conclusion

This paper compares RNNs and transformers, highlighting their distinct strengths and weaknesses. RNNs excel in low-resource and noisy data scenarios, while transformers handle complex dependencies and larger datasets more effectively. The analysis of sentiment classification and translation tasks provides practical guidance for selecting models based on task complexity and data availability.

These findings stress the need for problem-specific model design. RNNs' efficiency suits resource-constrained tasks, while transformers thrive in data-rich, complex settings. Future research could explore hybrid models combining RNNs' robustness with transformers' scalability or optimize transformers for low-resource scenarios. Investigating advanced data augmentation techniques also holds potential for enhancing model performance and generalization.

References

[1] Rnns are not transformers (yet): The key bottleneck on in-context retrieval. *arXiv preprint arXiv:2402.18510v2*, 2024.

[2] K. Bagla, A. Kumar, S. Gupta, and A. Gupta. Noisy text data: Achilles' heel of popular transformer-based nlp models. *arXiv preprint arXiv:2110.03353*, 2021.

[3] Bangoc123. Implementation for paper bleu: A method for automatic evaluation of machine translation. GitHub repository. Retrieved from `https://github.com/bangoc123/BLEU`.

[4] Stefan L. Frank, L. Charlotte Otten, Willem Zuidema, and Vera Demberg. Human sentence processing: Recurrence or attention? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 12–22. Association for Computational Linguistics, 2021.

[5] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang. A comparative study on transformer vs rnn in speech applications. In *Proceedings of Interspeech 2019*, 2019.

[6] F. D. Keles, P. M. Wijewardena, and C. Hegde. On the computational complexity of self-attention. In *Proceedings of NeurIPS 2022*, 2022.

[7] LaurentVeyssier. Machine-translation-english-french-with-deep-neural-network/data. GitHub repository. Retrieved from `https://github.com/LaurentVeyssier/Machine-translation-English-French-with-Deep-neural-Network/tree/main/data`.

[8] K. J. Liang, S. B. Rangrej, V. Petrovic, and T. Hassner. Few-shot learning with noisy labels. *arXiv preprint arXiv:2204.00000.*

[9] A. Maas. Large movie review dataset. *Sentiment Analysis*. Retrieved from `https://ai.stanford.edu/~amaas/data/sentiment/`.

[10] I. D. Mienye, T. G. Swart, and G. Obaido. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *MDPI Information*, 15(9):517, 2024.

[11] PyTorch. Nlp from scratch: Translation with a sequence to sequence network and attention. PyTorch Tutorials. Retrieved from `https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html`.

[12] D. Rai, Y. Zhou, S. Feng, A. Saparov, and Z. Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

[13] Saadarshad102. Sentiment analysis using recurrent neural networks (rnn-lstm) and google news word2vec. GitHub repository. Retrieved from `https://github.com/saadarshad102/Sentiment-Analysis-RNN-LSTM`.

[14] Y. Tang, Y. Wang, J. Guo, Z. Tu, K. Han, H. Hu, and D. Tao. A survey on transformer compression. *arXiv preprint arXiv:2400.00000.*

[15] P. Xu, D. Kumar, W. Yang, W. Zi, K. Tang, C. Huang, J. C. K. Cheung, S. J. D. Prince, and Y. Cao. Optimizing deeper transformers on small datasets. *arXiv preprint arXiv:2010.00000.*