Comparison of Text-to-Math Model Architectures

Anant Kumar

April 6, 2024

Abstract

This document details the implementation, experimentation, and comparison of four different model architectures for converting natural language to Math formulas.

1 Model Implementations

1.1 Seq2Seq Model with GloVe Embeddings

It consists of an encoder, which encodes input sequences into a fixed-size context vector, and a decoder, which generates output sequences based on the context vector. The model is trained to minimize the difference between predicted and actual output sequences. This architecture is commonly used for tasks like machine translation and sequence generation.

1.1.1 Accuracy and Performance

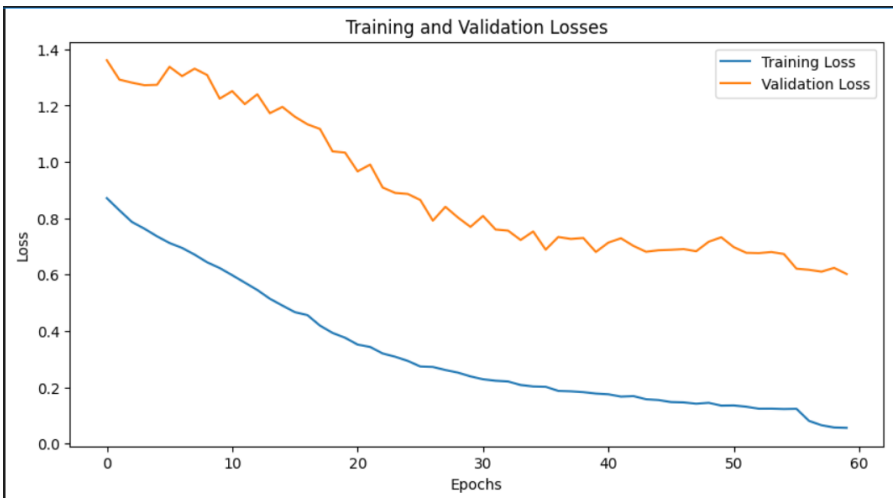
Metric	Test (%)	Validation (%)
Execution Accuracy	51.32	49.81
Exact Match Accuracy	47.06	45.76

Table 1: Performance metrics for the model on test and validation sets.

1.1.2 Hyperparameters

- Encoder embedding dimension (ENC_EMB_DIM): 100
- Decoder embedding dimension (DEC_EMB_DIM): 100
- Encoder hidden dimension (ENC_HID_DIM): 512
- Decoder hidden dimension (DEC_HID_DIM): 1024
- Encoder dropout rate (ENC_DROPOUT): 0.5

- Decoder dropout rate (DEC_DROPOUT): 0.5
- Learning rate: 0.001
- GlOVE embeddings : 100



(a) Loss curves for the Seq2Seq model with GloVe Embeddings.

Figure 1: Model Loss Plots

1.1.3 Insight

This model serves as a foundational approach, utilizing GloVe embeddings to capture semantic meaning from input sequences. Despite its straightforward architecture, it demonstrates a modest performance with an execution accuracy of 51.32 percent on the test set and 49.81 percent on the validation set. Its exact match accuracy also suggests room for improvement. The relatively simple architecture might limit its ability to capture complex relationships in the data.

1.2 Seq2Seq + Attention Model with GloVe Embeddings

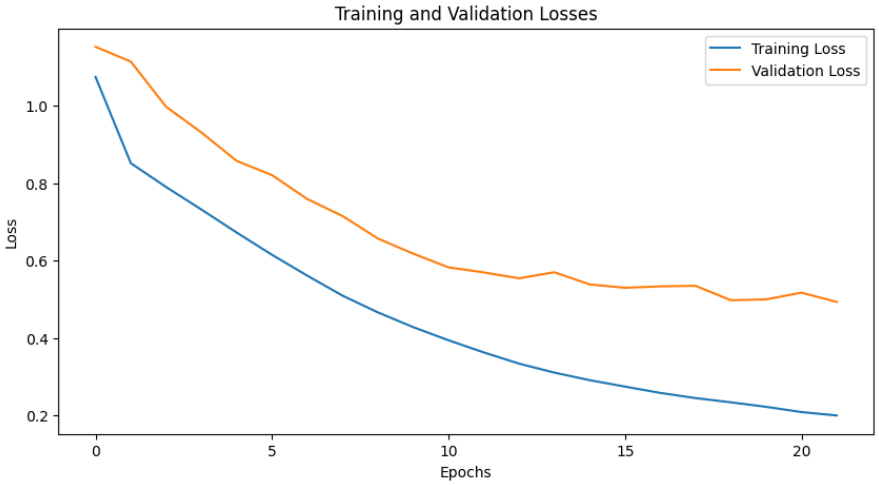
1.2.1 Accuracy and Performance

Teacher Forcing Ratio	Metric	Value (%)
0.3	Test Execution Accuracy	40.10
	Test Exact Match Accuracy	38.54
	Valid Execution Accuracy	XX.X
	Valid Exact Match Accuracy	XX.X
0.6	Test Execution Accuracy	58.12
	Test Exact Match Accuracy	54.28
	Valid Execution Accuracy	58.22
	Valid Exact Match Accuracy	54.67
0.9	Test Execution Accuracy	43.94
	Test Exact Match Accuracy	40.0
	Valid Execution Accuracy	44.20
	Valid Exact Match Accuracy	40.39

Table 2: Test and validation metrics for beam width 10 with various teacher forcing ratios.

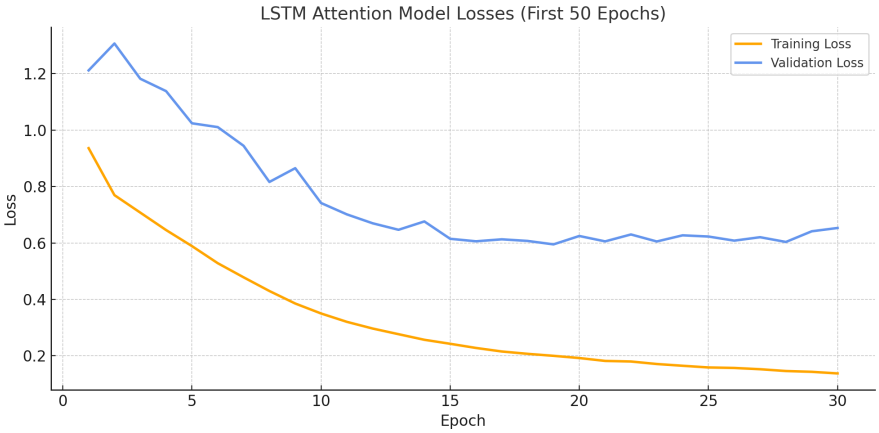
1.2.2 Hyperparameters

- Encoder embedding dimension (ENC_EMB_DIM): 100
- Decoder embedding dimension (DEC_EMB_DIM): 100
- Encoder hidden dimension (ENC_HID_DIM): 256
- Decoder hidden dimension (DEC_HID_DIM): 512
- Encoder dropout rate (ENC_DROPOUT): 0.5
- Decoder dropout rate (DEC_DROPOUT): 0.5
- Learning rate: 0.001
- GlOVE embeddings : 100



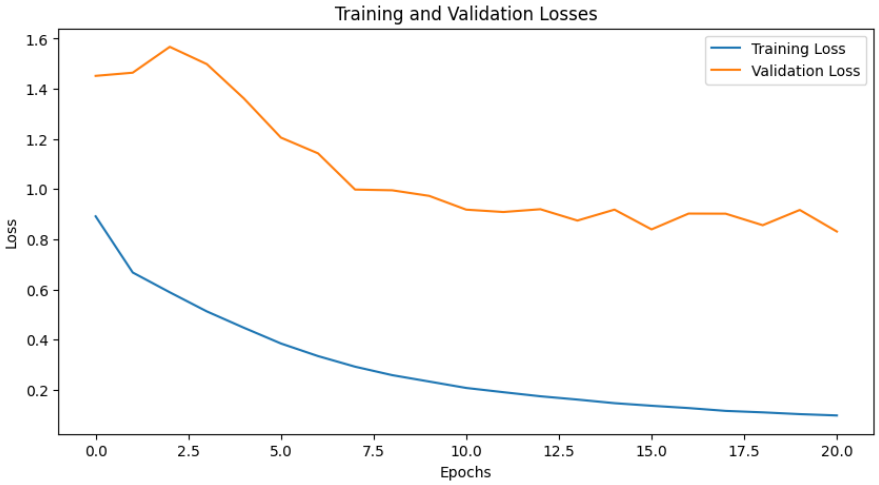
(a) Loss curves for the Seq2Seq + Attention model with GloVe Embeddings with 0.3 Teacher forcing ratio.

Figure 2: Model Loss Plots



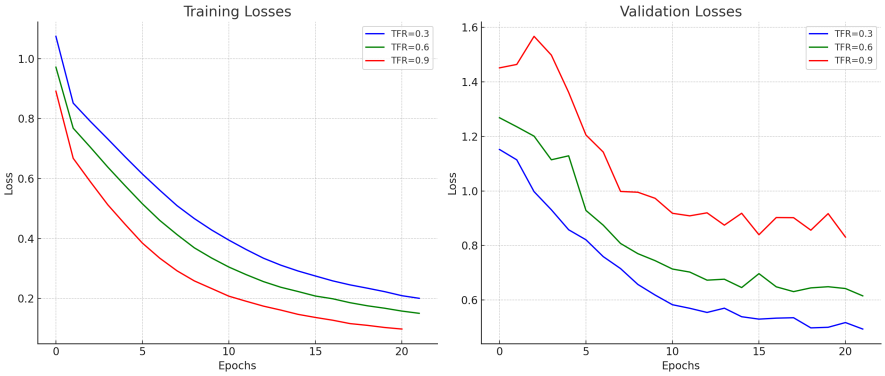
(a) Loss curves for the Seq2Seq + Attention model with GloVe Embeddings with 0.6 teacher forcing ratio.

Figure 3: Model Loss Plots



(a) Loss curves for the Seq2Seq + Attention model with GloVe Embeddings with 0.9 teacher forcing ratio.

Figure 4: Model Loss Plots



(a) Comparision of Loss curves for the Seq2Seq + Attention model with different teacher forcing ratios

Figure 5: Model Loss Plots

1.2.3 Insights

Incorporating attention mechanisms improves the model's ability to focus on relevant parts of the input sequence when generating Mathematical Formulas, leading to a significant performance leap. The model performs best with a teacher forcing ratio of 0.6, achieving 58.12 percent execution accuracy on the test set and 58.22 percent on the validation set.

1.2.4 Teacher Forcing Ratio Impact

The Seq2Seq + Attention model's performance clearly depends on the teacher forcing ratio, demonstrating a nuanced interaction between this ratio and model accuracy. The optimal performance is observed with a teacher forcing ratio of 0.6, showcasing the highest execution accuracy (58.12 percent on test, 58.22 percent on validation) and exact match accuracy (54.28 percent on test, 54.67 percent on validation). However, an interesting aspect is evident from the loss plots at a teacher forcing ratio of 0.9, where the model shows signs of overfitting. Specifically, the model likely learns to rely too much on the ground truth data provided during training, diminishing its ability to generalize to unseen data. This overfitting is detrimental to model performance, as indicated by reduced execution and exact match accuracies at this ratio.
In contrast, at lower ratios (e.g., 0.3), the model might not be leveraging the available ground truth efficiently, leading to underperformance compared to the 0.6 ratio. Thus, the loss plots and performance metrics together reveal a critical insight: while a moderate teacher forcing ratio (0.6) strikes a balance between learning from the ground truth and generating independent predictions, a high ratio (0.9) pushes the model towards overfitting, where it performs well on the training data but less so on unseen data.

1.3 BERT Seq2Seq + Attention Model with Frozen BERT-base-cased

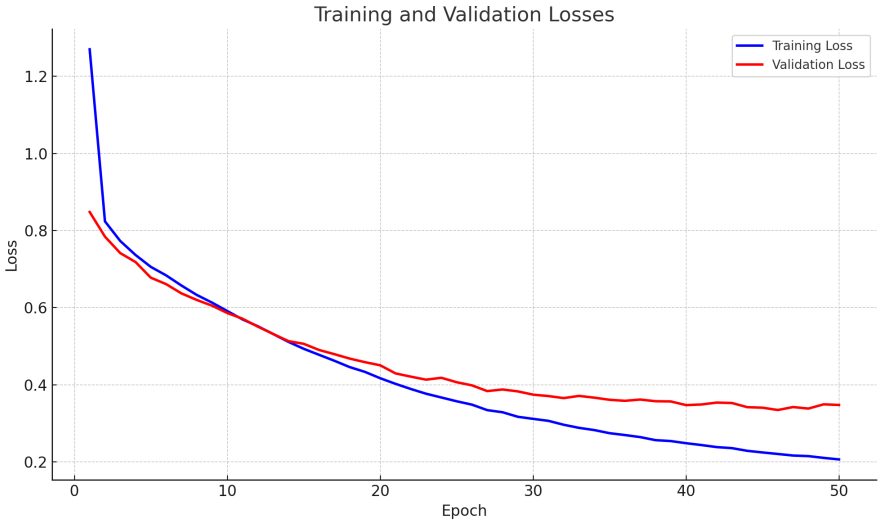
1.3.1 Accuracy and Performance

Metric	Test (%)	Validation (%)
Execution Accuracy	56.98	57.37
Exact Match Accuracy	53.61	54.44

Table 3: Performance metrics for the BERT Seq2Seq + Attention model with Frozen BERT-base-cased.

1.3.2 Hyperparameters

- Decoder embedding dimension (DEC_EMB_DIM): 100
- Encoder hidden dimension (ENC_HID_DIM): 128
- Decoder hidden dimension (DEC_HID_DIM): 256
- Decoder dropout rate (DEC_DROPOUT): 0.5
- Learning rate: 0.001



(a) Loss curves for the BERT Seq2Seq + Attention model with Frozen BERT-base-cased.

Figure 6: Model Loss Plots

1.3.3 Insight

The loss plots for the BERT Seq2Seq + Attention model with a frozen BERT-base-cased encoder reveal that the training and validation losses closely mirror each other throughout the training process. This parallel movement indicates a well-balanced model that is learning effectively without overfitting or underfitting significantly. The congruence between training and validation loss curves suggests that the model's learning from the pre-trained BERT embeddings is effectively transferred to the task of problem to formula conversion, maintaining consistency in performance across both seen (training) and unseen (validation) data. This harmony between the losses is particularly noteworthy, considering the model's architecture restricts further adaptation of the BERT encoder to the specific task. It underscores the robustness of utilizing deep contextual embeddings from pre-trained language models like BERT, even when they are not fine-tuned, to achieve solid performance across different datasets.

1.4 BERT Seq2Seq + Attention Model with Fine-tuned BERT-base-cased

1.4.1 Accuracy and Performance

Beam Width	Metric	Test Value (%)
1	Execution Accuracy	69.40
	Exact Match Accuracy	66.49
10	Execution Accuracy	69.14
	Exact Match Accuracy	66.54
20	Execution Accuracy	69.09
	Exact Match Accuracy	66.38

Table 4: Test metrics for different beam widths.

Metric	Value (%)
Test Execution Accuracy	69.14
Test Exact Match Accuracy	66.54
Valid Execution Accuracy	67.87
Valid Exact Match Accuracy	65.56

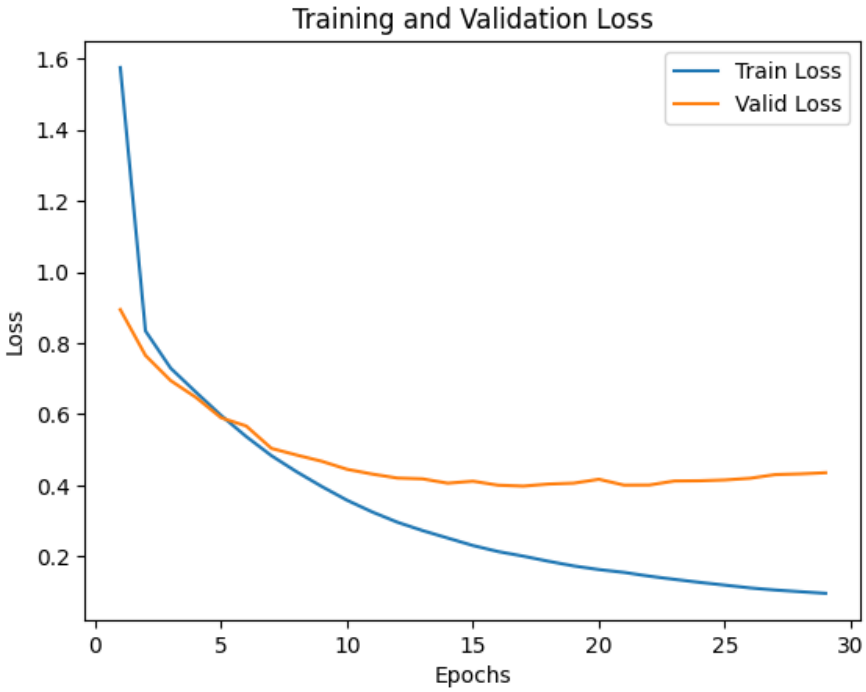
Table 5: Test and validation metrics for beam width 10.

1.4.2 Hyperparameters

- Decoder embedding dimension (DEC_EMB_DIM): 100
- Encoder hidden dimension (ENC_HID_DIM): 128
- Decoder hidden dimension (DEC_HID_DIM): 256
- Decoder dropout rate (DEC_DROPOUT): 0.5
- Learning rate: 1e-5 for BERT and 1e-3 for all other parameters

1.4.3 Insights

Fine-tuning the BERT-base-cased encoder results in the highest performance among the evaluated models, achieving an execution accuracy of 69.14 percent and an exact match accuracy of 66.54 percent on the test set for a beam width of 10. This superior performance underscores the significant advantage of leveraging the nuanced language understanding capabilities of BERT, further enhanced by the process of fine-tuning,



(a) Loss curves for the BERT Seq2Seq + Attention model with Fine-tuned BERT-base-cased.

Figure 7: Model Loss Plots

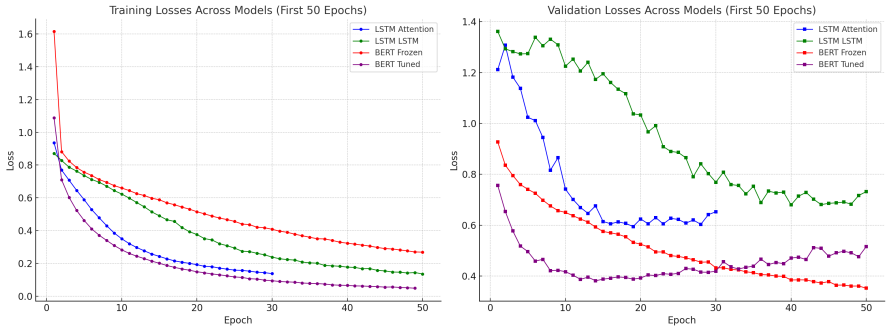
which tailors the model more precisely to the task of converting complex natural language queries into Math formulas. However, an important observation from the loss plots is that, at later stages of training, the validation loss begins to increase, diverging from the training loss. This divergence suggests the onset of overfitting, where the model starts to memorize the training data, reducing its generalization capability on unseen data. Such a trend highlights a critical area for improvement in the training regimen.

1.5 Effect of BEAM width

: For the BERT Seq2Seq + Attention model with Fine-tuned BERT-base-cased, varying the beam width during decoding shows a relatively stable performance across different beam widths (1, 10, 20). The execution accuracies hover around 69 percent, and exact match accuracies are slightly over 66 percent, with minimal fluctuations. This stability indicates that the model is robust to changes in beam width, suggesting a well-tuned decoding strategy that effectively explores multiple hypotheses without significant losses in performance.

2 Model Comparison

This section compares all the models based on their execution and exact match accuracies, loss curves.



(a) Loss curves for the all the 4 models

Figure 8: Model Loss Plots

3 Comparative Performance Insights from Plots

Observation from Comparison Plots: The comparison plots show a clear difference in how the four models perform. The BERT Seq2Seq + Attention model with a Fine-tuned BERT-base-cased encoder is the best performer. It not only has higher accuracy in executing and exactly matching queries but also shows a stable learning process. This model's success comes from using BERT's detailed understanding of language, which gets even better after fine-tuning for converting natural language into math formulas. Models without BERT or with a frozen BERT encoder have more difficulty and inconsistency, showing the importance of deep learning and the ability to adjust to the task.

4 What Could Have Been Better

Potential Areas for Improvement: The fine-tuned BERT model, despite its strong results, shows an area for improvement as the validation loss increases over time. This suggests that the model could be better optimized to avoid overfitting. Strategies like early stopping, adjusting the learning rate, or using new regularization methods might improve the model's ability to work well with new data. Also, models using GloVe embeddings might do better by trying different embedding methods or adding features to better understand the language.

5 Insights for Future Directions

Task-specific Fine-tuning: Fine-tuning models on a wide range of language tasks before focusing on converting text to math formulas could make the models even more versatile.

Data Augmentation Techniques: Using data augmentation to increase the training data could expose models to more ways of expressing queries, helping them perform better.

Exploring New Evaluation Metrics: New metrics that look at the accuracy of the converted formulas or how efficiently they run could offer deeper insights into how well the models are doing.

part 1 models