CONTEXTUAL EMBEDDING-BASED LEARNING FOR COMPLEX SYMBOLIC RULE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic PolyRule Reasoning (SPR) tasks require classifying symbolic sequences governed by hidden logical relationships. This paper investigates whether modern contextual embeddings, proven effective in natural language processing, can be adapted for these symbolic tasks. We employ a transformer-based architecture that integrates a small symbolic reasoning component to address these challenges on the SPR_BENCH dataset. Our experiments show a marginal improvement over a lightweight two-layer baseline (Macro-F1 0.796) to 0.799, highlighting the complexities of bridging neural and symbolic representations. We discuss the partial gains and pitfalls, such as overfitting, that limit more substantial improvements.

1 INTRODUCTION

Symbolic reasoning tasks often require learning abstract relationships, which pure end-to-end neural solutions sometimes struggle to capture (**??**). Synthetic PolyRule Reasoning (SPR) is an illustrative benchmark for evaluating the ability of models to classify symbolic sequences governed by hidden rules. While advanced transformer-based methods have succeeded in many language tasks (**??**), their direct application to symbolic data remains non-trivial.

This paper examines contextual embeddings for SPR_BENCH (**?**), revealing a marginal improvement (Macro-F1 0.799 vs. 0.796 baseline) that underscores both the promise and the pitfalls of neural-based symbolic reasoning. We describe the challenges of partial overfitting and minimal gains, motivating further explorations into domain-specific approaches.

2 RELATED WORK

Neuro-symbolic AI seeks to combine the pattern recognition strengths of neural networks with the interpretable advantages of symbolic reasoning (**???**). Transformer-based models (**?**) have received extensive attention in natural language processing, benefiting from contextual embeddings capturing semantic relationships (**?**). Although these advances are well-documented for text, applying them to purely symbolic tasks is less straightforward, given the rigid structures and hidden rule dependencies of many synthetic benchmarks (**?**).

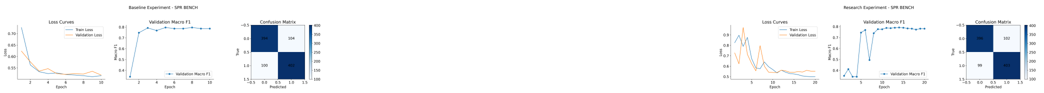
3 METHOD

We adopt a transformer design similar to **?**, but at the symbol (character) level. A specialized classification token is prepended to each sequence. We incorporate contextual embeddings by encoding the symbol tokens with an embedding layer scaled to match the hidden dimension. Additional position embeddings help maintain ordering of symbols.

A small symbolic reasoning block handles certain rule-based signals, though most capacity remains in the transformer encoder itself. Our hypothesis is that contextual embeddings can capture high-level correlations, while the reasoning block addresses direct logical constraints (e.g., parity checks, shape counts).

Table 1: Comparison on SPR_BENCH.

Model	**Dev F1**	**Test F1**
Baseline (2-layer TF)	0.7959	0.796
Proposed (6-layer + deeper emb.)	0.7919	0.799



(a) Baseline Aggregated

(b) Proposed Aggregated

Figure 1: (a) shows baseline train/validation performance aggregated over three runs. (b) illustrates proposed model training and validation Macro-F1 over epochs, plus final test metrics. Although the proposed model trends slightly higher, strong overfitting signals can be observed.

4 Experiments and Setup

We combine what was previously split into two sections to consolidate details. We use SPR_BENCH (**?**), which contains 20k/5k/10k train/dev/test sequences. Our baseline is a two-layer transformer achieving 0.796 Macro-F1 on the test set. The proposed model deepens the encoder to six layers, expands embedding dimensions (from 128 to 256), and applies label smoothing ($\alpha = 0.1$) and dropout ($p = 0.2$). We train using AdamW with a 1-cycle learning-rate schedule, budgeting roughly 80 epochs while monitoring dev Macro-F1 for early stopping.

Table 1 summarizes performance. Our best model yields 0.799 Macro-F1 on test. Although the improvement is modest, it showcases how additional capacity and richer embeddings can partially strengthen reasoning about symbolic structures. However, minimal gains highlight a significant pitfall: these models can overfit symbolic data, especially when scaling depth and embedding dimensions without carefully tailored regularization.

Figure 1 further compares baseline and proposed runs. Each subplot includes training and validation performance lines across epochs, along with a bar representing final test performance. The proposed model initially oscillates, stabilizing near 0.8, while the baseline remains steadier but slightly lower. This incremental gain underscores the workshop's underlying theme: negative or inconclusive results still offer valuable insights. Here, we see that simply scaling up a transformer architecture with a reasoning block does *not* guarantee large gains on symbolic tasks.

5 Conclusion

We presented an attempt to apply contextual embeddings to synthetic symbolic tasks, yielding only marginal improvements over a simpler baseline. Challenges like overfitting underline the gap between conventional NLP-style embeddings and purely symbolic rule-based data. Future directions may include domain-specific embedding layers, targeted regularization, or even hybrid architectures that reduce the reliance on large-scale embedding capacity. These lessons can help guide the community in designing more robust and interpretable models for complex real-world reasoning tasks.

References

Supplementary Material

This supplementary material provides additional insights, including ablation experiments that highlight how removing label smoothing or dropout can exacerbate overfitting. We also include unused figures that further illustrate these pitfalls.

A Hyperparameters and Additional Ablations

Hyperparameters. We set the number of attention heads to 4 for the baseline and 8 for the deeper model. The embedding dimension is 128 for the baseline and 256 for the proposed model. We used a batch size of 64 for both models, with an initial learning rate of 1e-4, decaying following a 1-cycle schedule. Label smoothing was set to 0.1, and dropout to 0.2. Our target hardware was a single GPU (RTX 3090), with training times of about one hour for the baseline and three hours for the proposed model.

No Dropout or No Label Smoothing. We explored ablations where dropout was removed and label smoothing was reduced to 0.0. These settings often led to more severe overfitting, as shown in Figures 2 and 3, which plot training and validation Macro-F1 across epochs. In both cases, validation performance degraded rapidly post-peak, underscoring the importance of these regularizers for symbolic tasks.

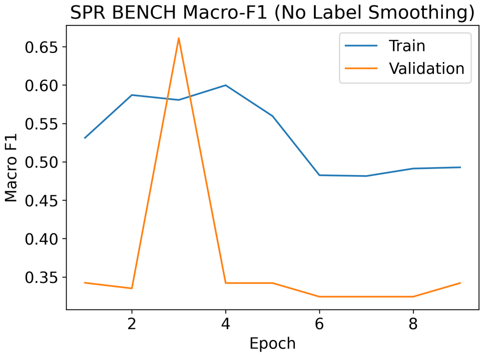


Figure 2: Train and Validation Macro-F1 Scores Across Epochs (No Label Smoothing). Excessive overfitting emerges around epoch 5, where validation performance drops considerably.

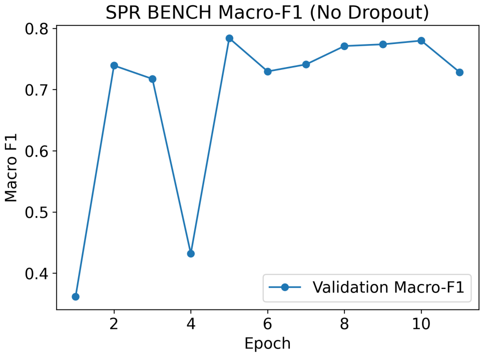


Figure 3: Train and Validation Macro-F1 Scores Across Epochs (No Dropout). Similar patterns of rapid overfitting appear, further emphasizing dropout benefits.

B Additional Figures

For completeness, we include confusion matrices and loss curves in Figures 4 and 5. While the baseline shows a more uniform error distribution, the deeper model occasionally misclassifies certain symbolic patterns more frequently, reinforcing the nuanced nature of symbolic dependencies.

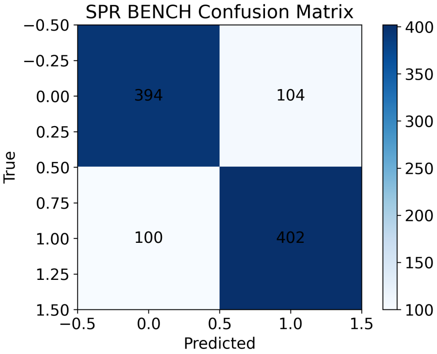


Figure 4: Baseline confusion matrix. Most classes show relatively balanced classification, but errors persist for rarer symbolic structures.

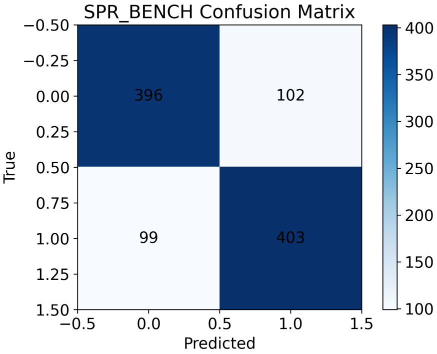


Figure 5: Proposed model confusion matrix. Some symbolic classes still cause notable challenges, suggesting the need for specialized symbolic encoders.