What Formal Languages can Transformers Learn In-Context?

Ali Rahim, Noor Rahim

Depatment of Mathematics, University of Rochester, Department of Computer Science, University of Colorado Boulder

Problem Definition

We investigate the capabilities of transformer models in learning formal languages in an in-context setting:

- **Task**: Given a set of in-context examples $\{(x_i, y_i)\}_{i=1}^n$ from a formal language L, generate a new string x_{n+1} that belongs to L.
- **Key Questions**:
  - How does the sample complexity n scale for languages in REG, CFL, CSL, and RE?
  - What are the theoretical bounds on transformer model complexity for each language class?
  - Can transformers achieve efficient in-context learning for context-free and more complex languages?
  - What are the fundamental limitations of in-context learning for recursively enumerable languages?
- **Approach**: Analyze transformers' ability to implicitly learn and apply the rules of formal grammars from examples, without explicit training.

Background

- Transformers act as algorithm approximators, can perform in-context learning at least as well as gradient descent
  - [1] showed decoder-only transformers are few shot learners
  - [2] showed they learn by gradient descent on in-context examples!
  - [3] showed they can actually do better than gradient descent and select better algorithms
  - [4] showed they perform higher order optimization on ICL examples, approximating iterative Newton's method

Methodology

- Analyze in-context learning capabilities for each language class:
  - Regular Languages (REG)
  - Context-Free Languages (CFL)
  - Context-Sensitive Languages (CSL)
  - Recursively Enumerable Languages (REL)
- Derive bounds on:
  - Model complexity (attention heads, hidden dimension, layers)
  - Sample complexity for in-context examples
  - Convergence rates in the in-context setting

Key Results: Regular Languages

- Model Specification for in-context learning:
  - $H = O(\log(|Q| + |\Sigma|))$
  - $D = O(\sqrt{|Q| \log |\Sigma|})$
  - $L = O(\log n)$
- In-context Sample Complexity ($\sim$ Coupon Collector's Problem [5]):

$$n \geq O\left(\frac{|Q| \cdot |\Sigma|}{l} \cdot \log\left(\frac{|Q| \cdot |\Sigma|}{\delta}\right)\right)$$

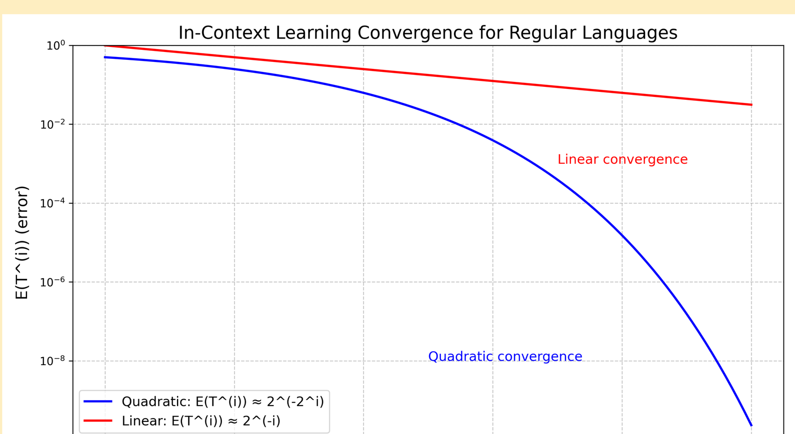
- Potential quadratic convergence in context:

$$E(T^{(i+1)}) \leq c \cdot (E(T^{(i)}))^2$$

- Where:
  - H: Number of attention heads, D: Hidden dimension, L: Number of layers
  - $|Q|$: Number of states in minimal DFA
  - $|\Sigma|$: Size of the alphabet
  - n: Number of in-context examples
  - l: Average length of example strings
  - δ: Confidence parameter
  - $E(T^{(i)})$: Error after i samples

Toy Example Convergence Rate Derivation

1. Model state as probability distribution over DFA states: $p = [p_0, p_1]$
2. Error defined as: $E(T^{(i)}) = 1 - p_{\text{correct}}$
3. Key insight: Each transformer layer can combine multiple transitions
4. For a single transition: $p' \approx 1 - (1 - p)^2$ (error squared)
5. Generalizing: $E(T^{(i+1)}) \leq c \cdot (E(T^{(i)}))^2$, for some constant $c < 1$
6. This quadratic reduction leads to convergence rate of $O(2^{-2^i})$



Key Results: Context-Free Languages

- In-context model with k implicit stack-like representations:
  - $H = O(\log(|R| + k))$
  - $D = O(\sqrt{|R| \log(|V| + |\Sigma|) + k \log(|V| + |\Sigma|)})$
  - $L = O(\log n + \log k)$, $k = O(\log n)$
- Naive Bound on In-context Sample Complexity:

$$n \geq O\left(\frac{|R| \cdot \log(|V| + |\Sigma|) \cdot \log(1/\varepsilon)}{\log(H \cdot D) + \log k}\right) \cdot \log(1/\delta)$$

- **Tighter Lattice-Based Bound:**

$$n \geq O(|R| \cdot \log(|V| + |\Sigma|) \cdot (\log \log |R| + \log(1/\varepsilon)))$$

- Where:
  - $|R|$: Number of production rules
  - $|V|$: Number of non-terminal symbols
  - k: Number of implicit stack-like representations
  - ε: Desired accuracy

Key Results: Context-Sensitive Languages

- In-context model with m tape-like representations in output:
  - $H = O(\log(|P| + m))$
  - $D = O(\sqrt{|P| \log(|V| + |\Sigma|) + m \log(|V| + |\Sigma| + 1)})$
  - $L = O(\log n \cdot \log m)$, $m = O(n)$
- In-context Sample Complexity:

$$n \geq O\left(\frac{|P| \cdot \log(|V| + |\Sigma|) \cdot \log(1/\varepsilon)}{\log(H \cdot D) + \log m}\right) \cdot \log(1/\delta)$$

- Where:
  - $|P|$: Number of production rules in CSG
  - m: Number of implicit tape-like representations

Recursively Enumerable Languages

- Conjecture: No finite transformer model can in-context learn to probabilistically generate from all RELs (the non-probabilistic case is trivial)
- In-context approximation possibilities:
  - Bounded-length input approximations
  - Probabilistic in-context recognition
  - In-context learning of decidable subsets

Conclusions & Future Work

- Clear correspondence between transformer in-context learning capabilities and formal language complexity
- Efficient in-context learning of regular and context-free languages
- Significant increase in sample complexity for in-context learning of context-sensitive languages
- Future questions and directions:
  - Can we find the convergence rate on CFL, CSL, and so on? Empirical testing suggests similar rate but we don't know why.
  - Can we find 'scaling laws' for in-context learning?
  - Optimize transformer architectures for in-context learning
  - Find out how the model represents grammar inference algorithms internally in hidden states and activations. (Significantly, this can help investigate the behavior of "meta-parameters' that seem to emerge at inference time.)

References

[1] T. B. Brown et al., "Language Models are Few-Shot Learners," *arXiv preprint arXiv:2005.14165*, 2020.

[2] J. von Oswald et al., "Transformers learn in-context by gradient descent," *arXiv preprint arXiv:2212.07677*, 2023.

[3] Y. Bai, F. Chen, H. Wang, C. Xiong, and S. Mei, "Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection," *arXiv preprint arXiv:2306.04637*, 2023.

[4] D. Fu, T.-Q. Chen, R. Jia, and V. Sharan, "Transformers Learn Higher-Order Optimization Methods for In-Context Learning: A Study with Linear Models," *arXiv preprint arXiv:2310.17086*, 2024.

[5] M. Mitzenmacher and E. Upfal, "Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis," Cambridge University Press, 2017.