Language Models, Mathematics, Embeddings

Zsolt Zombori* [1,3] Pál Zsámboki* [1,3] Ádám Fraknói [1] Máté Gedeon [2] András Kornai [2,4]

Alfréd Rényi Institute of Mathematics, Budapest

Dept. of Algebra, Budapest University of Technology and Economics

Eötvös Loránd University, Budapest, Hungary

SZTAKI Institute of Computer Science

Goals

- Top-level goal: using LLMs to guide symbolic theorem provers
- Subgoal: understanding (evolving or creating) a language whereby the prover can communicate its current state and the LLM can provide hints. This language should have both a vectorial and a formulaic facet allowing human-interpretable communication between the two sides
- Strategy: study how various classes of formulas are represented in LLMs
- Special emphasis on logic formulas potentially suitable for representing thm prover state (as opposed to formulas of arithmetic, algebra, analysis etc)
- Well-formed formulas are already hard (matching parens, quantifier scoping)
- Understanding how LLMs can represent similar formulas is key

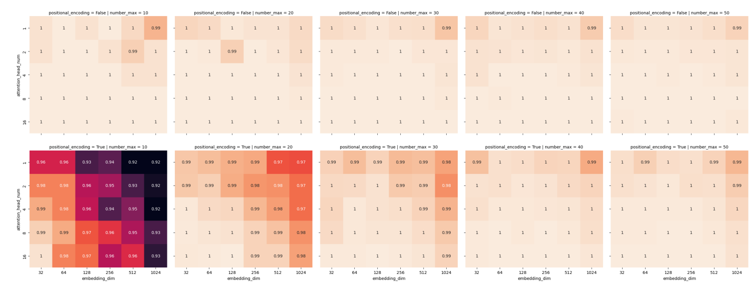
Plan of the talk

Our life is frittered away by detail. Simplify, simplify, simplify! I say, let your affairs be as two or three, and not a hundred or a thousand; instead of a million count half a dozen, and keep your accounts on your thumb-nail (Henry David Thoreau)

- Simplify I: From FOL to propositional calculus
- Using Allamanis et al., 2016 data on converting extended propositional formulas to normal form
- Simplify II: from well-arranged systems of parentheses (Dyck lg) to finding out just how many are there in a string
- Simplify III: from highly capable LLMs to small model systems

Simplifying the simplest task

- There are three tokens '0' corresponding to open paren; '1' to close paren; '2' to non-paren. Find if $\#0 \geq \#1$, emit 3 if it is, 4 if it isn't
- Train set 70k strings where the number of each digit is ≤ 100; validation set (15k strings) with $100 \leq$ strlen(0,1,2) ≤ 150; test set (15k strings) with $150 \leq$ strlen(0,1,2) ≤ 200
- Grid search over positional encoding yes/no; # dimensions, #transformer layers; #attention heads

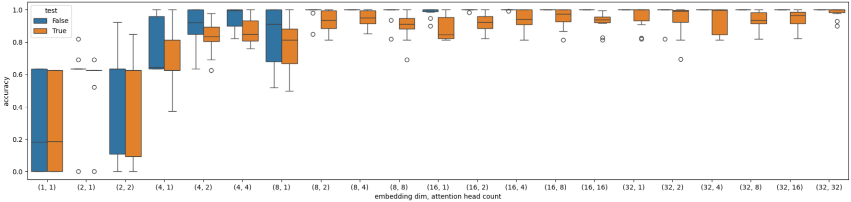


- No need for positional encoding – unsurprising given that the system does *deep sets* (problem is permutation-invariant, see Zaheer et al., 2018)
- No need for more than 32 dimensions (this will be reduced to 2 later, and can in principle be one)
- Just one layer, just one attention head will be good enough for perfect systems that generalize to 100% accuracy on test data 'learned the rule'

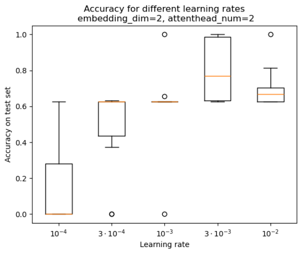
The attention mechanism

- Suppose static embedding has n dimensions, and we have k attention heads. By convention, the dimension of an attention head is chosen to be $d = n/k$
- A head is characterized by three $n \cdot d$ matrices called the query Q, the key K, and the value V, each producing a d-dim vector called the (token- and head-specific) query, key, and value
- In a single layer we compute in parallel at each token t, and for each head h, the sum of tV_h weighted by the scalar product $(t'Q'_h, t'V'_h)$. Afterwards, we concatenate the k resulting d-dim vectors and add the original input vector

More dimensions help the search



Getting to the simplicity maximum

- Reverse engineering the 32 dim 32 head model shows 9 "winning" attention heads that classify to 100% by themselves
- With 16 dim and 16 heads we still find winning heads (but fewer)
- With 8/8 and 4/4 we no longer find winners, but we know they exist!
- With 2/2 other hyperparameters, in particular the learning rate, become a big deal
- 

At the simplicity maximum

- Actually we can produce a perfect 1-dimensional head for $n = 2$ data, we just cannot find it by random initialization and training
- A simple setup with value $v(0) = -1; v(1) = 1$, key $k(0) = k(1) = 1; k(2) = -100$ and query $q(1) = 1$ will do the work
- `tracr` (Lindner et al., 2023) lets you generate transformer weights based on RASP descriptions (Weiss, Goldberg, and Yahav, 2021) but we just use `numpy`

Collaboration among heads

Quite often, we can find heads that are in themselves imperfect, but in combination perfect.

head	accuracy	model
1	0.5693	-0.20998879 * (head_1 out) + 0.87861097
29	0.9493	-0.15839106 * (head_29 out) + 1.031981
1+29	1.0	(0.17374; 0.83133) * (pred_1; pred_29) - 0.00226

How much we need to simplify?

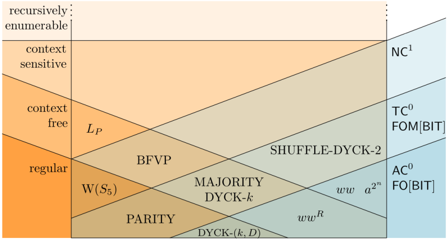


Figure 1: Relationship of some languages and language classes discussed in this paper (right) to the Chomsky hierarchy (left), assuming that $\mathsf{TC}^0 \subsetneq \mathsf{NC}^1$ and $\mathsf{L} \subsetneq \mathsf{NL}$. Circuit classes are DLOGTIME-uniform.

Figure from Strobl et al., 2024

Acknowledgements

Zombori has been supported by Hungarian National Excellence Grant 2018-1.2.1-NKP-00008, the Hungarian Artificial Intelligence National Laboratory (RRF-2.3.1-21-2022-00004) and the ELTE TKP 2021-NKTA-62 funding scheme. Kornai has been partially supported by 2018-1.2.1-NKP-00008: Exploring the Mathematical Foundations of Artificial Intelligence, and the Hungarian Artifical Intelligence Laboratory (MILAB). Fraknói has been supported by the ÚNKP-23-2 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

European Research Council
Established by the European Commission

National Research, Development and Innovation Office
Hungary

PROJECT FINANCED
FROM THE NRDI FUND

Thank You

References

- Allamanis, Miltiadis et al. (2016). "Learning Continuous Semantic Representations of Symbolic Expressions". In: *arXiv preprint arXiv:1611.01423*.
- Lindner, David et al. (2023). *Tracr: Compiled Transformers as a Laboratory for Interpretability*. arXiv: 2301.05062 [cs.LG]. URL: https://arxiv.org/abs/2301.05062.
- Strobl, Lena et al. (2024). "What Formal Languages Can Transformers Express? A Survey". In: *Transactions of the Association for Computational Linguistics* 12, pp. 543–561. DOI: https://doi.org/10.1162/tacl.a.00663.
- Weiss, Gail, Yoav Goldberg, and Eran Yahav (2021). *Thinking Like Transformers*. arXiv: 2106.06981 [cs.LG]. URL: https://arxiv.org/abs/2106.06981.
- Zaheer, Manzil et al. (2018). *Deep Sets*. arXiv: 1703.06114 [cs.LG]. URL: https://arxiv.org/abs/1703.06114.