Formal Languages and the NLP Black Box

William Merrill
willm@nyu.edu

New York University, New York, NY 10011, USA

1 Introduction

The field of natural language processing (NLP) has been transformed in two related ways in recent years. First, the field moved towards using neural network architectures like LSTMs[1] [12] and transformers [31], in contrast to approaches that explicitly represent grammatical rules. Another innovation was a move towards *semi-supervised learning* [24,26,7]: language models have been used in various ways to solve downstream NLP tasks that previously would have required large labeled datasets. Transformer-based language models in particular have been remarkably empirically successful across a range of NLP tasks [28], and making the models and datasets bigger tends to not only improve performance on benchmarks [13] and linguistic generalization [30,32], but can also lead to the emergence of new algorithmic behavior, such as arithmetic and logical reasoning [33].

I will argue that there are many intriguing mysteries about these empirical results that formal language theory can clarify. It seems as if large transformer language models are implicitly learning some aspects of natural language grammar [30]. If so, it seems useful to understand what kinds of formal grammars such networks can simulate, and how grammatical dependencies are represented within them. For comparing and extending neural network architectures, it would also be useful to have a theory of how different types of neural networks compare in expressive power to one another.

Rather than just focusing on grammatical dependencies, we may also adopt a similar formal language theoretic perspective for understanding reasoning in transformer language models. Can we characterize the kinds of computational problems transformers can solve? Can we use this theory to extract algorithmic behavior from a transformer into a discrete, human-readable program? Can we predict the amount of language modeling data needed to solve various reasoning problems, or find problems that transformers can *never* solve, even at massive scale?

I will survey recent work that provides some insight on the computational model of RNNs and transformers. I will start by discussing an older line of work analyzing the power of different RNN variants in relation to one another and the Chomsky hierarchy. I will then discuss a newer line of work that analyzes the computational power of transformers using circuit complexity theory.

[1] An LSTM is a special kind of recurrent neural network [8,9].

While the techniques used in these two lines of work may be different, some unified insights emerge for understanding the capabilities and inner workings of both RNNs and transformers. In particular, *counting* seems to be a central computational ability to the kinds of processing possible in both LSTMs and transformers. Yet the benefit of counting is not the same for each architecture: transformers—but not LSTMs—can use counting to recognize k-Dyck for any k, a capability often taken to embody sensitivity to hierarchical structure [2]. On the other hand, a fundamental weakness of transformers compared to LSTMs is parallelism: while RNNs can simulate certain computation graphs linear in the size of the input sequence, transformers have a constant-depth computation graph, and thus must do much more processing in parallel.

2 RNNs

There are deep connections between neural networks with recurrence and automata: historically, one motivation for developing finite automata was to model the computation of networks of biological neurons with binary activation patterns [15]. The 1990s saw the analysis of RNN models with linear-thresholded activations, which, with infinite precision and run time, are Turing-complete and thus significantly more expressive than finite automata [27]. See [17] for further discussion of RNN results with infinite precision.

But the infinite-precision, infinite-runtime model [27] does not capture the type of RNNs used in modern deep learning, which are typically unrolled with one step per input token ("real-time"), and suffer from practical precision constraints that prevent storing an unbounded Turing machine tape in a finite number of numbers in $[-1, 1]$.[2] How then should we understand the set of languages that RNNs can learn to recognize in practice? A central finding here is that LSTMs, one RNN extension, can recognize languages requiring counting, whereas most other RNNs cannot [34]. [16] then proposed saturated RNNs as a simplified theoretical model of bounded-precision RNNs, and showed that the expressive power of saturated RNNs often predicts the empirical abilities of RNNs.

2.1 LSTMs Can Count, Other RNNs Cannot

Aiming to understand the practical power of RNNs, [34] empirically evaluate the ability of basic RNNs and their variants, LSTMs [12] and GRUs [5], to recognize the formal language $a^n b^n$. Empirically, they show that LSTMs can recognize $a^n b^n$, while RNNs and GRUs cannot. Moreover, they show the LSTM achieves

[2] There are at least two reasons to view this practical precision setting as more realistic. First, hardware imposes a maximum precision on each number in the RNN, which may reasonably be considered to be finite or logarithmic in the input sequence length n. Second, RNNs are trained by gradient descent, and we would like to understand the class of languages that can be learned by an RNN. Intuitively, constructions that are sensitive to low-order imprecision may be hard to learn by gradient descent [34].

this by "counting": using a memory cell to track the difference between the number of a's and b's in the input. Similar results were then observed for other languages requiring counting, such as 1-Dyck or shuffled Dyck [29]. In contrast, LSTMs were not able to reliably learn 2-Dyck, which requires a stack as opposed to just counting [29].[3]

2.2 Saturated RNNs as a Model of Practical RNNs

[16,22] analyze the expressive power of *saturated* RNNs as a proxy for what unsaturated RNNs can learn by gradient descent. This technique is motivated by the hypothesis that networks requiring bounded parameter norm are unlikely to be acquired by a training process where the parameter norm is growing consistently over the course of training.[4] Given a network $f(x;\theta)$, a saturated network is the function $f'(x;\theta)$ obtained by making the parameters θ large:

$$f'(x;\theta) = \lim_{\rho \to \infty} f(x;\rho\theta).$$

The effect of making the weights large in this way is to convert the activation functions in all parts of the network to step functions. [16] place saturated RNNs, GRUs, and LSTMS in the Chomsky hierarchy. Saturated RNNs and GRUs are equivalent to finite automata [16], whereas saturated LSTMs can simulate a constrained class of counter automata that can recognize 1-Dyck, $a^n b^n$, or $a^n b^n c^n d^n$ but do not have enough memory to recognize 2-Dyck. Thus, LSTMs can be understood to cross-cut the conventional Chomsky hierarchy: able to recognize some context-sensitive languages, but unable to simulate a stack or process arbitrary hierarchical structure. This analysis of saturated networks places different types of RNNs in different relations to the Chomsky hierarchy, and these predictions largely match the type of languages that unsaturated RNNs can learn to recognize in practice.

2.3 Summary and Open Questions

Both empirical results and the theoretical model of saturated networks suggests that counting is a key capability of LSTMs that RNNs and GRUs do not have. While counting enables LSTMs to recognize some languages like $a^n b^n$ and 1-Dyck, it does not allow LSTMs to process arbitrary hierarchical or context-free structure (e.g., 2-Dyck or palindromes).

3 Transformers

Over the last 5 years, transformers have largely replaced RNNs as the backbone of neural NLP systems. A difficulty in extending automata-based analysis of

[3] See also [6] for more recent, but similar, empirical results on the ability of different RNN variants to recognize formal languages.

[4] See [18] for thorough empirical exploration of norm growth and saturation during the training of large transformer language models.

RNNs to transformers is that the transformer neural network architecture lacks autoregressive structure. Instead, recent work has made progress understanding the power of transformers by relating transformers to formal language classes defined by circuit families and logics.

3.1 Transformers with Hard Attention

[10] prove that the transformers with hard attention cannot recognize even simple formal languages like parity or 1-Dyck. [11] extend this result to prove that hard-attention transformers can be simulated[5] by constant-depth, poly-size circuit families, which recognize the formal language class AC^0. This implies [10]'s results, as well as demonstrating new languages that hard-attention transformers cannot recognize, such as majority (taking the majority vote of a sequence of bits).

3.2 Transformers with Soft Attention

[3] show that, like LSTMs, transformers have the ability to count, and can use this to recognize 1-Dyck and other related formal languages. [21,25] show that transformers can also use counting to recognize majority, implying that soft attention is stronger than hard attention.

[21] then analyze *saturated* transformers, which have simplified attention patterns compared to soft attention, but can still count. They find that saturated transformers over a floating-point data type can be simulated by constant-depth, poly-size *threshold* circuit families (i.e., the complexity class TC^0). Intuitively, counting is one of the key capabilities achievable in TC^0 but not AC^0, suggesting that counting is a good way to understand the gain in power that saturated attention grants relative to hard attention.

[19] then extend [21]'s result to show that arbitrary soft-attention transformers with precision logarithmic in the input length can be simulated in the tighter class log-space-uniform TC^0. Log-space-uniform TC^0 is conjectured to be separated from other complexity classes like L, NL, or P, which would imply that transformers cannot solve complete problems for these classes. Thus, accepting these separation conjectures, transformers cannot compute connectivity in directed or undirected graphs, solve the universal context-free grammar recognition problem, or solve linear systems of matrix equations.

3.3 Logics and Programming Languages for Expressing Transformer Computation

One converging theme in recent work has been attempting to propose symbolic formalisms describing computation in transformers.

[5] Here, "simulate" means that any function computed by a transformer can also be computed by such a circuit family.

[20] further refine the circuit-based upper on soft-attention transformers, showing that transformers can be simulated in log-time-uniform TC^0. This class has an equivalent characterization as the formal languages definable in first-order logic with majority quantifiers [23], or $\mathsf{FO(M)}$. This immediately implying that soft-attention transformers can be "translated" to first-order logic formulae with majority quantifiers that compute the same function.

Along similar lines, [4] propose $\mathsf{FOC}(+, \mathsf{MOD})$, or first-order logic with counting quantifiers[6], as a logical model of transformers. [4] prove that $\mathsf{FOC}(+, \mathsf{MOD})$ is an upper bound on finite-precision transformers and a lower bound on transformers with arbitrary precision, although figuring out whether there is some model of transformers for which it is a tight bound remains open.

Finally, [35] propose a programming language called RASP for expressing computation in a transformer-like way. [14] create a compiler that compiles constrained RASP programs into actual transformers. [35] shows that RASP can recognize arbitrary Dyck languages (not just 1-Dyck), and, empirically, transformers can as well.[7]

3.4 Summary and Open Questions

Counting—a key capability separating LSTMs from RNNs—also separates soft-attention transformers from hard-attention transformers. Upper bounds on the power of transformers derived via circuit complexity give us classes of problems that transformers cannot solve, but which are efficiently solvable by a recurrent model of computation like a Turing machine. The intuition behind why these problems are hard for transformers is that transformers are fundamentally constrained to parallel computation, and it is conjectured in complexity theory that certain problems are fundamentally unparallelizable.

Significant progress has been made on the analysis of transformers in recent years, and connections to deep questions in complexity theory have been revealed. Yet, there are still many things that are unclear. Is saturated attention fundamentally weaker than soft attention? Can we make upper bounds and lower bounds on transformers tighter? Can we leverage theoretical insights to extract discrete computational mechanisms from trained transformers?

4 Conclusion

There has been a wide range of work in recent years analyzing the capabilities of RNNs and transformers as formal grammars. One unifying insight that has emerged for both types of neural networks is that they can leverage counting to process structure in their input, although potentially in different ways. Transformers in particular can use counting to recognize arbitrary Dyck languages,

[6] with addition and mod but not ordering over positions

[7] In their experiments, [35] add special regularization to the attention patterns of the transformer to get it to learn Dyck languages properly.

which are often viewed as an exemplar of hierarchical structure. Due to the transformer's lack of autoregressive structure, circuit complexity and logic have been more useful than automata theory for understanding transformers' capabilities. Hopefully, insights from these theories may continue to refine our ability to peer into the black box of transformers, and perhaps understanding transformers may even inspire new research questions or advances in these fields.

5 Resources

A key area for active research is the Formal Languages and Neural Networks (FLaNN) Discord server and talk series. For more extensive (but out of date) surveys, the reader should see [1,17]. A more up-to-date survey of the circuit complexity-based analysis of transformers is under preparation by members of FLaNN.

Acknowledgments

Thank you to Michael Hu for his feedback.

References

1. Ackerman, J., Cybenko, G.: A survey of neural networks and formal languages (2020)
2. Autebert, J.M., Berstel, J., Boasson, L.: Context-free languages and pushdown automata. Handbook of Formal Languages: Volume 1 Word, Language, Grammar pp. 111–174 (1997)
3. Bhattamishra, S., Ahuja, K., Goyal, N.: On the Ability and Limitations of Transformers to Recognize Formal Languages. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7096–7116. Association for Computational Linguistics, Online (Nov 2020), https://aclanthology.org/2020.emnlp-main.576
4. Chiang, D., Cholak, P., Pillay, A.: Tighter bounds on the expressivity of transformer encoders (2023)
5. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. In: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. pp. 103–111. Association for Computational Linguistics, Doha, Qatar (Oct 2014), https://aclanthology.org/W14-4012
6. Deletang, G., Ruoss, A., Grau-Moya, J., Genewein, T., Wenliang, L.K., Catt, E., Cundy, C., Hutter, M., Legg, S., Veness, J., Ortega, P.A.: Neural networks and the chomsky hierarchy. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=WbxHAzkeQcn
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019), https://aclanthology.org/N19-1423

8. Elman, J.L.: Finding structure in time. Cognitive science **14**(2), 179–211 (1990)
9. Goldberg, Y.: A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research **57**, 345–420 (2016)
10. Hahn, M.: Theoretical limitations of self-attention in neural sequence models. Transactions of the Association for Computational Linguistics **8**, 156–171 (2020), https://aclanthology.org/2020.tacl-1.11
11. Hao, Y., Angluin, D., Frank, R.: Formal language recognition by hard attention transformers: Perspectives from circuit complexity. Transactions of the Association for Computational Linguistics **10**, 800–810 (2022), https://aclanthology.org/2022.tacl-1.46
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
13. Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C.D., Ré, C., Acosta-Navas, D., Hudson, D.A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., Wang, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N., Khattab, O., Henderson, P., Huang, Q., Chi, R., Xie, S.M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., Koreeda, Y.: Holistic evaluation of language models (2022)
14. Lindner, D., Kramár, J., Rahtz, M., McGrath, T., Mikulik, V.: Tracr: Compiled transformers as a laboratory for interpretability (2023)
15. Mcculloch, W., Pitts, W.: A logical calculus of ideas immanent in nervous activity. Bulletin of Mathematical Biophysics **5**, 127–147 (1943)
16. Merrill, W.: Sequential neural networks as automata. In: Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges. pp. 1–13. Association for Computational Linguistics, Florence (Aug 2019), https://aclanthology.org/W19-3901
17. Merrill, W.: Formal language theory meets modern nlp (2021)
18. Merrill, W., Ramanujan, V., Goldberg, Y., Schwartz, R., Smith, N.A.: Effects of parameter norm growth during transformer training: Inductive bias from gradient descent. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 1766–1781. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.133, https://aclanthology.org/2021.emnlp-main.133
19. Merrill, W., Sabharwal, A.: The parallelism tradeoff: Limitations of log-precision transformers (2023)
20. Merrill, W., Sabharwal, A.: Transformers can be expressed in first-order logic with majority (2023)
21. Merrill, W., Sabharwal, A., Smith, N.A.: Saturated transformers are constant-depth threshold circuits. Transactions of the Association for Computational Linguistics **10**, 843–856 (2022), https://aclanthology.org/2022.tacl-1.49
22. Merrill, W., Weiss, G., Goldberg, Y., Schwartz, R., Smith, N.A., Yahav, E.: A formal hierarchy of RNN architectures. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 443–459. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.43, https://aclanthology.org/2020.acl-main.43
23. Mix Barrington, D.A., Immerman, N., Straubing, H.: On uniformity within nc1. Journal of Computer and System Sciences **41**(3), 274–306 (1990), https://www.sciencedirect.com/science/article/pii/002200009090022D

24. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018), https://aclanthology.org/N18-1202
25. Pérez, J., Marinković, J., Barceló, P.: On the turing completeness of modern neural network architectures. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HyGBdo0qFm
26. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
27. Siegelmann, H., Sontag, E.: On the computational power of neural nets. Journal of Computer and System Sciences **50**(1), 132–150 (1995), https://www.sciencedirect.com/science/article/pii/S0022000085710136
28. Srivastava, A., Rastogi, A., Rao, A., Shoeb, A.A.M., Abid, A., Fisch, A., Brown, A.R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A.W., Safaya, A., Tazarv, A., Xiang, A., Parrish, A., Nie, A., Hussain, A., Askell, A., Dsouza, A., Slone, A., Rahane, A., Iyer, A.S., Andreassen, A., Madotto, A., Santilli, A., Stuhlmüller, A., Dai, A., La, A., Lampinen, A., Zou, A., Jiang, A., Chen, A., Vuong, A., Gupta, A., Gottardi, A., Norelli, A., Venkatesh, A., Gholamidavoodi, A., Tabassum, A., Menezes, A., Kirubarajan, A., Mullokandov, A., Sabharwal, A., Herrick, A., Efrat, A., Erdem, A., Karakaş, A., Roberts, B.R., Loe, B.S., Zoph, B., Bojanowski, B., Özyurt, B., Hedayatnia, B., Neyshabur, B., Inden, B., Stein, B., Ekmekci, B., Lin, B.Y., Howald, B., Diao, C., Dour, C., Stinson, C., Argueta, C., Ramírez, C.F., Singh, C., Rathkopf, C., Meng, C., Baral, C., Wu, C., Callison-Burch, C., Waites, C., Voigt, C., Manning, C.D., Potts, C., Ramirez, C., Rivera, C.E., Siro, C., Raffel, C., Ashcraft, C., Garbacea, C., Sileo, D., Garrette, D., Hendrycks, D., Kilman, D., Roth, D., Freeman, D., Khashabi, D., Levy, D., González, D.M., Perszyk, D., Hernandez, D., Chen, D., Ippolito, D., Gilboa, D., Dohan, D., Drakard, D., Jurgens, D., Datta, D., Ganguli, D., Emelin, D., Kleyko, D., Yuret, D., Chen, D., Tam, D., Hupkes, D., Misra, D., Buzan, D., Mollo, D.C., Yang, D., Lee, D.H., Shutova, E., Cubuk, E.D., Segal, E., Hagerman, E., Barnes, E., Donoway, E., Pavlick, E., Rodola, E., Lam, E., Chu, E., Tang, E., Erdem, E., Chang, E., Chi, E.A., Dyer, E., Jerzak, E., Kim, E., Manyasi, E.E., Zheltonozhskii, E., Xia, F., Siar, F., Martínez-Plumed, F., Happé, F., Chollet, F., Rong, F., Mishra, G., Winata, G.I., de Melo, G., Kruszewski, G., Parascandolo, G., Mariani, G., Wang, G., Jaimovitch-López, G., Betz, G., Gur-Ari, G., Galijasevic, H., Kim, H., Rashkin, H., Hajishirzi, H., Mehta, H., Bogar, H., Shevlin, H., Schütze, H., Yakura, H., Zhang, H., Wong, H.M., Ng, I., Noble, I., Jumelet, J., Geissinger, J., Kernion, J., Hilton, J., Lee, J., Fisac, J.F., Simon, J.B., Koppel, J., Zheng, J., Zou, J., Kocoń, J., Thompson, J., Kaplan, J., Radom, J., Sohl-Dickstein, J., Phang, J., Wei, J., Yosinski, J., Novikova, J., Bosscher, J., Marsh, J., Kim, J., Taal, J., Engel, J., Alabi, J., Xu, J., Song, J., Tang, J., Waweru, J., Burden, J., Miller, J., Balis, J.U., Berant, J., Frohberg, J., Rozen, J., Hernandez-Orallo, J., Boudeman, J., Jones, J., Tenenbaum, J.B., Rule, J.S., Chua, J., Kanclerz, K., Livescu, K., Krauth, K., Gopalakrishnan, K., Ignatyeva, K., Markert, K., Dhole, K.D., Gimpel, K., Omondi, K., Mathewson, K., Chiafullo, K., Shkaruta, K., Shridhar, K., McDonell, K., Richardson, K., Reynolds, L., Gao, L., Zhang, L., Dugan, L., Qin, L., Contreras-Ochando, L., Morency, L.P., Moschella, L., Lam, L., Noble, L., Schmidt, L., He, L., Colón, L.O., Metz, L., Şenel,

L.K., Bosma, M., Sap, M., ter Hoeve, M., Farooqi, M., Faruqui, M., Mazeika, M., Baturan, M., Marelli, M., Maru, M., Quintana, M.J.R., Tolkiehn, M., Giulianelli, M., Lewis, M., Potthast, M., Leavitt, M.L., Hagen, M., Schubert, M., Baitemirova, M.O., Arnaud, M., McElrath, M., Yee, M.A., Cohen, M., Gu, M., Ivanitskiy, M., Starritt, M., Strube, M., Swedrowski, M., Bevilacqua, M., Yasunaga, M., Kale, M., Cain, M., Xu, M., Suzgun, M., Tiwari, M., Bansal, M., Aminnaseri, M., Geva, M., Gheini, M., T, M.V., Peng, N., Chi, N., Lee, N., Krakover, N.G.A., Cameron, N., Roberts, N., Doiron, N., Nangia, N., Deckers, N., Muennighoff, N., Keskar, N.S., Iyer, N.S., Constant, N., Fiedel, N., Wen, N., Zhang, O., Agha, O., Elbaghdadi, O., Levy, O., Evans, O., Casares, P.A.M., Doshi, P., Fung, P., Liang, P.P., Vicol, P., Alipoormolabashi, P., Liao, P., Liang, P., Chang, P., Eckersley, P., Htut, P.M., Hwang, P., Miłkowski, P., Patil, P., Pezeshkpour, P., Oli, P., Mei, Q., Lyu, Q., Chen, Q., Banjade, R., Rudolph, R.E., Gabriel, R., Habacker, R., Delgado, R.R., Millière, R., Garg, R., Barnes, R., Saurous, R.A., Arakawa, R., Raymaekers, R., Frank, R., Sikand, R., Novak, R., Sitelew, R., LeBras, R., Liu, R., Jacobs, R., Zhang, R., Salakhutdinov, R., Chi, R., Lee, R., Stovall, R., Teehan, R., Yang, R., Singh, S., Mohammad, S.M., Anand, S., Dillavou, S., Shleifer, S., Wiseman, S., Gruetter, S., Bowman, S.R., Schoenholz, S.S., Han, S., Kwatra, S., Rous, S.A., Ghazarian, S., Ghosh, S., Casey, S., Bischoff, S., Gehrmann, S., Schuster, S., Sadeghi, S., Hamdan, S., Zhou, S., Srivastava, S., Shi, S., Singh, S., Asaadi, S., Gu, S.S., Pachchigar, S., Toshniwal, S., Upadhyay, S., Shyamolima, Debnath, Shakeri, S., Thormeyer, S., Melzi, S., Reddy, S., Makini, S.P., Lee, S.H., Torene, S., Hatwar, S., Dehaene, S., Divic, S., Ermon, S., Biderman, S., Lin, S., Prasad, S., Piantadosi, S.T., Shieber, S.M., Misherghi, S., Kiritchenko, S., Mishra, S., Linzen, T., Schuster, T., Li, T., Yu, T., Ali, T., Hashimoto, T., Wu, T.L., Desbordes, T., Rothschild, T., Phan, T., Wang, T., Nkinyili, T., Schick, T., Kornev, T., Telleen-Lawton, T., Tunduny, T., Gerstenberg, T., Chang, T., Neeraj, T., Khot, T., Shultz, T., Shaham, U., Misra, V., Demberg, V., Nyamai, V., Raunak, V., Ramasesh, V., Prabhu, V.U., Padmakumar, V., Srikumar, V., Fedus, W., Saunders, W., Zhang, W., Vossen, W., Ren, X., Tong, X., Zhao, X., Wu, X., Shen, X., Yaghoobzadeh, Y., Lakretz, Y., Song, Y., Bahri, Y., Choi, Y., Yang, Y., Hao, Y., Chen, Y., Belinkov, Y., Hou, Y., Hou, Y., Bai, Y., Seid, Z., Zhao, Z., Wang, Z., Wang, Z.J., Wang, Z., Wu, Z.: Beyond the imitation game: Quantifying and extrapolating the capabilities of language models (2022)

29. Suzgun, M., Belinkov, Y., Shieber, S., Gehrmann, S.: LSTM networks can perform dynamic counting. In: Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges. pp. 44–54. Association for Computational Linguistics, Florence (Aug 2019). https://doi.org/10.18653/v1/W19-3905, https://aclanthology.org/W19-3905

30. Tenney, I., Das, D., Pavlick, E.: BERT rediscovers the classical NLP pipeline. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4593–4601. Association for Computational Linguistics, Florence, Italy (Jul 2019), https://aclanthology.org/P19-1452

31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

32. Warstadt, A., Zhang, Y., Li, X., Liu, H., Bowman, S.R.: Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language

Processing (EMNLP). pp. 217–235. Association for Computational Linguistics, Online (Nov 2020), https://aclanthology.org/2020.emnlp-main.16
33. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E.H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., Fedus, W.: Emergent abilities of large language models. Transactions on Machine Learning Research (2022), https://openreview.net/forum?id=yzkSU5zdwD, survey Certification
34. Weiss, G., Goldberg, Y., Yahav, E.: On the practical computational power of finite precision RNNs for language recognition. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 740–745. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), https://aclanthology.org/P18-2117
35. Weiss, G., Goldberg, Y., Yahav, E.: Thinking like transformers (2021), https://openreview.net/forum?id=TmkN9JmDJx1