




Computational Expressivity of Neural Language Models

Alexandra Butoi  Robin Chan  Ryan Cotterell  William Merrill 
Franz Nowak  Clemente Pasti  Lena Strobl  Anej Svete 

 ETH Zürich  New York University  Umeå Universitet

{[abutoi](mailto:abutoi@inf.ethz.ch),[chanr](mailto:chanr@nyu.edu),[ryan.cotterell](mailto:ryan.cotterell@nyu.edu),[fnowak](mailto:fnowak@inf.ethz.ch),[cpasti](mailto:cpasti@inf.ethz.ch),[asvete](mailto:asvete@inf.ethz.ch)}@inf.ethz.ch
lenas@cs.umu.se willm@nyu.edu

Abstract

Language models (LMs) are currently at the forefront of NLP research due to their remarkable versatility across diverse tasks. However, a large gap exists between their observed capabilities and the explanations proposed by established formal machinery. To motivate a better theoretical characterization of LMs’ abilities and limitations, this tutorial aims to provide a comprehensive introduction to a specific framework for formal analysis of modern LMs using tools from formal language theory (FLT). We present how tools from FLT can be useful in understanding the inner workings and predicting the capabilities of modern neural LM architectures. We cover recent results using FLT to make precise and practically relevant statements about LMs based on recurrent neural networks and transformers by relating them to formal devices such as finite-state automata, Turing machines, and analog circuits. Altogether, the results covered in this tutorial allow us to make precise statements and explanations about the observed as well as predicted behaviors of LMs, as well as provide theoretically motivated suggestions on the aspects of the architectures that could be improved.

 <https://acl2024.ivia.ch>

1 Introduction and Motivation

Language models pre-trained on massive amounts of web text have revolutionized NLP (Devlin et al., 2019; Liu et al., 2019; Brown et al., 2020). They have demonstrated utility in a variety of NLP tasks and have recently been proposed as a general model of computation for a wide variety of reasoning tasks (Brown et al., 2020; Chowdhery et al., 2022; Wei et al., 2022; Kojima et al., 2023; Kim et al., 2023; Huang and Chang, 2023, *inter alia*) or even as a basis for general AI (Kosinski, 2023).

Such beliefs are an extrapolation from LMs’ prowess at a number of tasks that were long deemed difficult for computers to solve. These realizations

mostly stem from contemporary empirical research on LMs’ abilities. However, empirical exploration can only take us so far; these types of analyses require large amounts of time and resources, and arguably, a bit of luck. Further, the individual findings from such studies rarely advance our actual knowledge of how LMs work and are often not generalizable (Chen et al., 2021; Anil et al., 2022) or reproducible (Melis et al., 2018; Belz et al., 2021; Nityasya et al., 2023). This has resulted in many imprecise claims being put forth in the literature—e.g., the claim that LMs are general-purpose reasoners—without appropriate technical definitions. However, in the context of computer science, the notion of a general-purpose reasoner is concretely defined. In fact, a formal definition for an algorithm was the motivation behind Alan Turing’s famous 1937 paper that introduced a Turing machine. Thus, a fair rephrasing of the claim that LMs are general-purpose reasoners is that they are Turing complete, which is a claim about their expressivity. With this in mind, the successes of large LMs have sparked interest in their representational capabilities. More broadly, a standard way of quantifying the expressive power of computational models is with the complexity of formal languages they can recognize (Delétang et al., 2023). Thus, we can expect that deeper integration of FLT into the study of neural LMs can facilitate a better understanding of the inherent types of computation that LMs are capable of performing.

A formal language L is a subset of Σ^* , the set of all strings over some alphabet Σ . The study of LMs with FLT concerns itself with the question: What classes of formal languages are LMs capable of modeling? Making this connection allows us to draw from the long line of work from classical computer science which has thoroughly described different classes of formal languages. In other words, the machinery allowing us to precisely understand LMs already exists, what is needed is

only the concrete connection to LMs. However, as most modern LMs are implemented as large neural networks, they are notoriously difficult to analyze theoretically, making the connection to formal models nontrivial. To be able to make any claims about their capabilities, existing work has introduced formalizations of modern LM architectures such as RNNs (Elman, 1990; Hochreiter and Schmidhuber, 1997; Cho et al., 2014) and transformers (Vaswani et al., 2017), focusing on investigating what classes of formal languages LMs can recognize. Through this, diverse formal properties of modern LM architectures have been shown (e.g., Siegelmann and Sontag, 1992; Hao et al., 2018; Korsky and Berwick, 2019; Merrill, 2019; Merrill et al., 2020; Hewitt et al., 2020; Hahn, 2020; Merrill et al., 2022; Chiang and Cholak, 2022; Merrill and Sabharwal, 2023; Strobl, 2023; Chiang et al., 2023, inter alia). These reveal, among others, that the behavior of some RNNs can be described as counting, and that transformers struggle to recognize some simple formal languages. These, and many similar results, are covered in the tutorial. However, studying the recognition of formal languages only reveals half of the story. A language model is formally a distribution over Σ^* —as such, LMs do not define formal languages—rather, they define *distributions* over Σ^* . In a sense, attempting to shoehorn LMs into classical FLT is, therefore, somewhat misplaced. In this tutorial, we thus ask an additional question: What classes of *probability distributions* can different LMs represent? In a recent survey, Delétang et al. (2023) tried to locate LMs on the Chomsky hierarchy, a notion that does not easily generalize to probabilistic languages (Chattopadhyay et al., 2020). Nonetheless, a thorough understanding of various probabilistic language classes exists (Icard, 2020), and studying LMs in their raw probabilistic form allows us to draw from those results directly.

In this **cutting-edge** tutorial, we give a comprehensive overview of various results characterizing the computation power of modern LMs. The tutorial is divided into four main parts. First, we formally introduce language models and the array of tools at our disposal in the form of computational devices used in FLT. In the second part, we explore the relationship between RNN LMs and formal models, covering both classic results spanning back to the first works on neural networks as well as some very recent findings. Then, we bring

our analysis over to the current state-of-the-art LM architecture, the transformer, and show how, despite its empirical success, its theoretical abilities are very nuanced. Finally, we discuss the implications of these theoretical results and outline some potential future research avenues. We believe that this tutorial can provide the NLP community with tools to better understand the capabilities and limitations of existing LMs, with the hope that this knowledge will help spark a more grounded discussion on their abilities as well as provide ideas for developing new methods that overcome their limitations.

2 Target Audience

Our tutorial is targeted at members of the NLP community interested in the theoretical capabilities of modern LMs. This includes both researchers seeking to gain insight into the inner workings of such models as well as those actively developing methods that might leverage the results presented in this tutorial. Although we intend to give a brief overview of the FLT concepts required for this tutorial, we expect some basic knowledge of (weighted) formalisms, including finite-state automata (FSAs), pushdown automata (PDAs), and Turing machines (TMs). Additionally, we expect familiarity with contemporary neural network architectures such as RNNs and transformers. While do not require any reading in particular, we have compiled a list of papers that we encourage reviewing (marked with an asterisk in the bibliography of the proposal).

3 Outline

3.1 Part 1: Background

A **language model** p is formally a probability distribution over Σ^* . Most modern LMs define $p(\mathbf{y})$ autoregressively, i.e., as a product of conditional probability distributions: $p(\mathbf{y}) \stackrel{\text{def}}{=} p(\text{EOS} \mid \mathbf{y}) \prod_{t=1}^{|\mathbf{y}|} p(y_t \mid \mathbf{y}_{<t})$, where $\text{EOS} \notin \Sigma$ is a special end of sequence symbol. In the first part of the tutorial, we formally introduce LMs and discuss some subtleties that arise in this definition, e.g., the notion of tightness (Du et al., 2023). To enable our formal analysis, we then introduce and motivate a suite of tools that FLT puts at our disposal for precisely characterizing the computational capacity of LMs. We describe common computational devices such as FSAs, PDAs, and TMs, both in their classical formulation as well as their probabilistic interpretations.

We then discuss a crucial difference between two paradigms of the analysis of LMs using FLT: (i) using LMs as *discrete recognizers* and studying which classes of formal languages they can recognize and (ii) embracing the true probabilistic nature of LMs and studying the classes of *probability distributions* over strings that they can represent. To give a concrete example, previous work has shown that modern language models correctly model syntax (Linzen et al., 2016). This raises the question of whether an LM architecture is formally able to model specific grammatical structures such as parse trees and existing work has shed light on this phenomenon (Hewitt et al., 2020). At an even more basic level, for example, computing a product of a sequence of -1 's requires a model to be able to recognize the PARITY language, a very simple regular language, which was shown to be difficult to model by transformers (Hahn, 2020). Considering the probabilistic nature of an LM, on the other hand, allows us to investigate them in their raw form, which is motivated both by a more direct analysis as well as the suitability of stochastic modeling of human cognition (Icard, 2020). A concrete question one might ask, for example, is whether an LM is capable of modeling a distribution over parse trees that appear in human language.

3.2 Part 2: Expressivity of RNN LMs

In the second part of our tutorial, we use the introduced toolset to characterize the computational capacity of various RNN-based LMs such as the Elman RNN (Elman, 1990), the LSTM (Hochreiter and Schmidhuber, 1997), and the GRU (Cho et al., 2014), using formal computational devices. More precisely, we explore the relationship between RNN LMs and probabilistic finite-state automata, counter machines, and Turing machines. We demonstrate that, under certain realistic conditions, Elman RNN LMs have the same expressive capacity as probabilistic FSAs. In contrast, LSTMs, due to their more elaborate recurrent mechanism, allow for a natural interpretation as counter devices, making them more powerful. Furthermore, we showcase that under a different set of assumptions even the simplest RNN LMs can emulate Turing machines, providing a mechanism for executing algorithms. We consider both the weighted as well as unweighted versions of this relationship through a generalization of the classic construction due to Siegelmann and Sontag (1992).

3.3 Part 3: Expressivity of Transformer LMs

Transformer-based LMs have shown unparalleled performance, which raises the natural question of the mechanisms behind their success, and in the third part of the tutorial, we cover many recent results trying to explain these successes. The study of transformers through the lens of FLT is both relatively new as well as more nuanced than that of RNNs. Due to their inherent parallelizability, transformers do not keep any internal state, which makes it difficult to formalize their connection to sequential models of computation such as automata. Indeed, a number of negative results have shown some theoretical limitations of self-attention (Hahn, 2020; Chiang and Cholak, 2022) in processing languages requiring sequentiality, which has sparked a line of work relating transformers to the less-expressive boolean circuits (Hao et al., 2022; Merrill et al., 2022; Merrill and Sabharwal, 2023; Strobl, 2023; Chiang et al., 2023). Moreover, the statelessness of the architecture invites an interpretation through the lens of simpler local models. In this spirit, we showcase that transformer LMs can exactly simulate n -gram LMs. While this establishes a concrete lower bound on the expressivity of transformer LMs, it can be seen as somewhat disappointing— n -gram models are examples of very simple *subregular* languages, and characterizing state-of-the-art models with respect to such formalisms leaves much to be desired. We then showcase a trick that allows one to significantly increase the computational power of transformer LMs by storing additional information in the generated string, which effectively gives the model access means of a memory structure. We present a construction whereby transformers can simulate sequential models from simple finite-state automata to Turing machines (Pérez et al., 2021). Interestingly, storing additional information in the output invites a very natural connection to the very popular mechanism of chain-of-thought prompting, and we showcase how chain-of-thought reasoning can be concisely formulated in this framework, bridging the gap between abstract theoretical work and tools used in practice.

3.4 Part 4: Implications of the Results

We finish the tutorial with a discussion of some immediate as well as less obvious implications of these results. An example is the undecidability of many standard NLP tasks such as finding the most

probable string under an LM or model minimization (Chen et al., 2018). These hold both for RNN LMs as well as those based on self-attention. While this sounds limiting, it justifies the use of heuristic approaches to language generation, including various sampling strategies. Some of these results also provide an explanation for the limited generalization capabilities of these models. Additionally, the theoretical upper bounds of their expressiveness provide insight into when these models should be able to truly learn algorithmic patterns and be able to execute them. Lastly, we cover a suite of possible future directions of research for those who are interested in continuing to study the relationship between neural LMs and FLT.

Included work. The tutorial presents a collection of works from a diverse set of authors who have helped establish the current formal understanding of LMs. We cover classical results from the 20th century as well as contemporary research from different research groups working in this field (as referenced above), together with some published and current work by the presenters of the tutorial.

4 Presenters

- **Alexandra Butoi** is a PhD student at ETH Zürich. Her current interests include formal language theory and its applications in understanding the abilities of modern neural network architectures.
- **Robin Chan** is a PhD student at ETH Zürich. His main research interests are formal language theory and human-AI collaboration.
- **Ryan Cotterell** is an assistant professor at ETH Zürich in the Institute for Machine Learning. His research focuses on a wide range of topics, including information-theoretic linguistics, parsing, computational typology and morphology, and bias and fairness in NLP systems.
- **Franz Nowak** is a PhD student at ETH Zürich. His research revolves around the formal properties of neural sequence models.
- **William Merrill** is a PhD student at NYU. His research focuses on using formal methods to understand the computational capabilities and limitations of language models, including their ability to represent linguistic structure and solve reasoning tasks.
- **Clemente Pasti** is a PhD student at ETH Zürich. His main research interests focus on

formal language theory and its applications in constraining and controlling the text generation of language models.

- **Lena Strobl** is a PhD student at the Foundations of Language Processing research group at Umeå University. Her main research interests focus on making modern neural network architectures more interpretable through formal language theory.
- **Anej Svete** is a PhD student at ETH AI Center at ETH Zürich. His main research interests lie at the intersection of formal language theory and LMs, where he is working on improving our understanding of the formal properties of modern architectures.

The presenters have collaborated to organize multiple relevant NLP-related courses at ETH Zürich together in the last few years: [Large Language Models](#), [Advanced Formal Language Theory](#), and [Natural Language Processing](#). Besides, they prepared a course on very similar material at ESSLLI 2023. Ryan has additionally given another ESSLLI course and taught at the University of Cambridge. He was also recently involved in teaching a tutorial on generating from LMs at ACL 2023.

5 Diversity Considerations

The tutorial centers on the theoretical properties of LMs used for a wide array of tasks and languages, making our presentation relevant to a large proportion of the NLP community. It is agnostic to the specifics of particular trained models and can be applied to many languages. Moreover, understanding the theoretical limitations to generalization can reveal when LMs can be expected to generalize to less-represented languages and capture their specific phenomena. Finally, our team of presenters comprises of both junior and senior researchers, including PhD students from three different universities and an assistant professor.

6 Ethics Statement

The tutorial motivates and outlines the theoretical investigation of the computational abilities of various modern LM architectures, aiming to better understand their limitations. As such, the presenters do not foresee any ethical issues.

References

- Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. [Exploring length generalization in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 38546–38556. Curran Associates, Inc.
- Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021. [A systematic review of reproducibility research in natural language processing](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 381–393, Online. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Agnishom Chattopadhyay, Filip Mazowiecki, Anca Muscholl, and Cristian Riveros. 2020. [Pumping lemmas for weighted automata](#). *CoRR*, abs/2001.06272.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Łukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Arun Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. [Evaluating large language models trained on code](#). *CoRR*, abs/2107.03374.
- Yining Chen, Sorcha Gilroy, Andreas Maletti, Jonathan May, and Kevin Knight. 2018. [Recurrent neural networks as weighted language recognizers *](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2261–2271, New Orleans, Louisiana. Association for Computational Linguistics.
- David Chiang and Peter Cholak. 2022. [Overcoming a theoretical limitation of self-attention](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7654–7664, Dublin, Ireland. Association for Computational Linguistics.
- David Chiang, Peter Cholak, and Anand Pillay. 2023. [Tighter bounds on the expressivity of transformer encoders](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 5544–5562. PMLR.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. [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*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. [PaLM: Scaling language modeling with pathways](#).
- Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2023. [Neural networks and the Chomsky hierarchy *](#). In *11th International Conference on Learning Representations*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Li Du, Lucas Torroba Hennigen, Tiago Pimentel, Clara Meister, Jason Eisner, and Ryan Cotterell. 2023. [A](#)

- measure-theoretic characterization of tight language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9744–9770, Toronto, Canada. Association for Computational Linguistics.
- Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.
- Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.
- Yiding Hao, Dana Angluin, and Robert Frank. 2022. Formal language recognition by hard attention transformers: Perspectives from circuit complexity. *Transactions of the Association for Computational Linguistics*, 10:800–810.
- Yiding Hao, William Merrill, Dana Angluin, Robert Frank, Noah Amsel, Andrew Benz, and Simon Mendelsohn. 2018. Context-free transductions with neural stacks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 306–315, Brussels, Belgium. Association for Computational Linguistics.
- John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D. Manning. 2020. RNNs can generate bounded hierarchical languages with optimal memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1978–2010, Online. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.
- Thomas F. Icard. 2020. Calibrating generative models: The probabilistic Chomsky–Schützenberger hierarchy *. *Journal of Mathematical Psychology*, 95:102308.
- Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.
- Samuel A. Korsky and Robert C. Berwick. 2019. On the computational power of RNNs. *CoRR*, abs/1906.06349.
- Michał Kosinski. 2023. Theory of mind might have spontaneously emerged in large language models. *CoRR*, abs/2302.02083.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *6th International Conference on Learning Representations*.
- William Merrill. 2019. Sequential neural networks as automata. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 1–13, Florence. Association for Computational Linguistics.
- William Merrill and Ashish Sabharwal. 2023. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545.
- William Merrill, Ashish Sabharwal, and Noah A. Smith. 2022. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856.
- William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. 2020. A formal hierarchy of RNN architectures *. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 443–459, Online. Association for Computational Linguistics.
- Made Nindyatama Nityasya, Haryo Wibowo, Alham Fikri Aji, Genta Winata, Radityo Eko Prasajo, Phil Blunsom, and Adhiguna Kuncoro. 2023. On “scientific debt” in NLP: A case for more rigour in language model pre-training research. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8554–8572, Toronto, Canada. Association for Computational Linguistics.
- Jorge Pérez, Pablo Barceló, and Javier Marinkovic. 2021. Attention is Turing-complete *. *Journal of Machine Learning Research*, 22(75):1–35.
- Hava T. Siegelmann and Eduardo D. Sontag. 1992. On the computational power of neural nets. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, page 440–449, New York, NY, USA. Association for Computing Machinery.
- Lena Strobl. 2023. Average-hard attention transformers are constant-depth uniform threshold circuits.

A. M. Turing. 1937. [On computable numbers, with an application to the Entscheidungsproblem](#). *Proceedings of the London Mathematical Society*, s2-42(1):230–265.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. [Emergent abilities of large language models](#).