Beyond Decoding: Meta-Generation Algorithms for Large Language Models

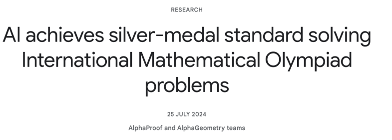
Presenters: Matthew Finlayson, Hailey Schoelkopf, Sean Welleck

December 11, 2024

Algorithms for generating outputs with a language model

Algorithms for generating outputs with a language model

Why? *Use **test-time compute** to improve performance*

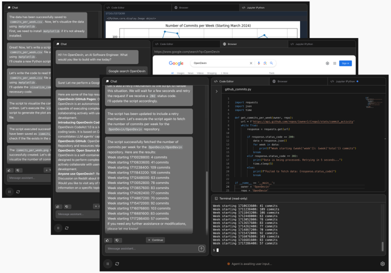

RESEARCH

AI achieves silver-medal standard solving International Mathematical Olympiad problems

25 JULY 2024

AlphaProof and AlphaGeometry teams

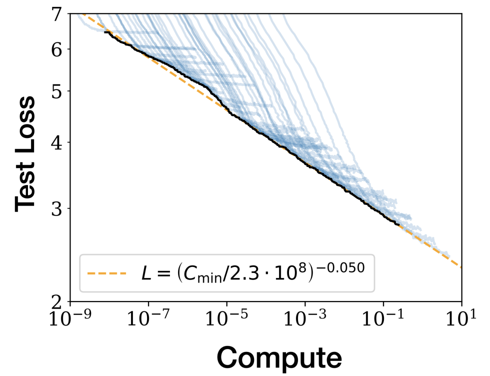
Solving olympiad problems



Writing code

Tasks framed as generating sequences: many other applications

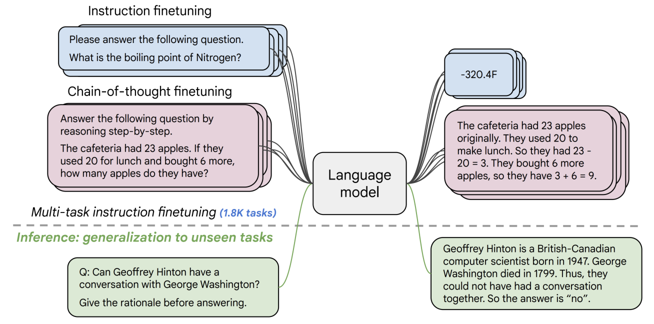
[2020-] **Scaling pretraining:** larger model, larger dataset



Scaling Laws for Neural Language Models [Kaplan et al., 2020]

Approach 2: scale post-training compute

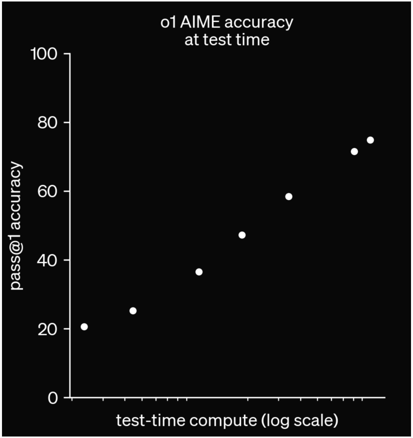
[2022-] **Scaling post-training:** e.g., fine-tune on (input, output) pairs



Scaling Instruction-Finetuned Language Models [Chung et al., 2022]

Approach 3: scale *test-time* compute

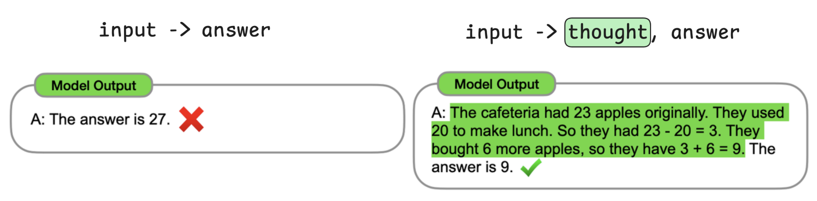
[Now] **Test-time scaling:** increase compute at generation time



Test-time compute vs. accuracy ([OpenAI, 2024])

Approach 3: scale *test-time* compute | How?

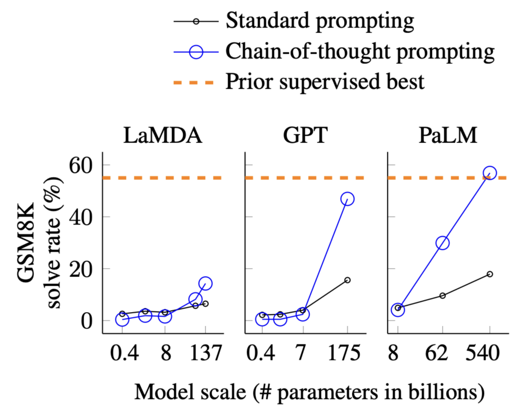
1. Generate extra tokens



[Wei et al., 2022]

Approach 3: scale *test-time* compute | How?

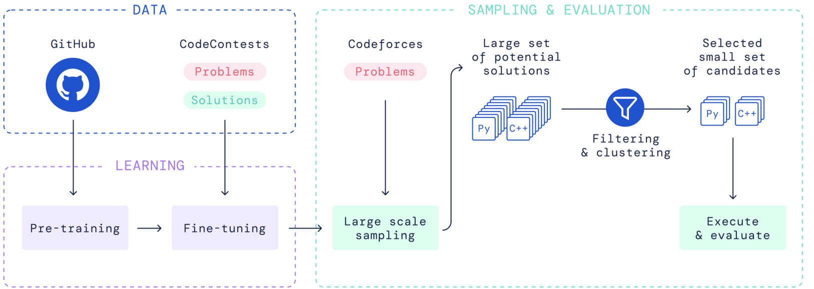
1. Generate extra tokens



[Wei et al., 2022]

Approach 3: scale *test-time* compute | How?

1. Generate extra tokens
2. Call generator multiple times

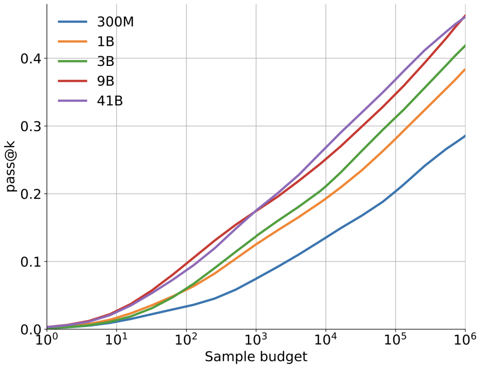


Overview of AlphaCode.

AlphaCode [Li et al., 2022]

Approach 3: scale *test-time* compute | How?

1. Generate extra tokens
2. Call generator multiple times

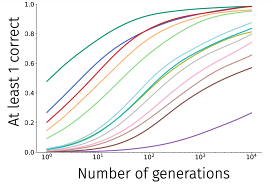


(b) Unlimited attempts per problem

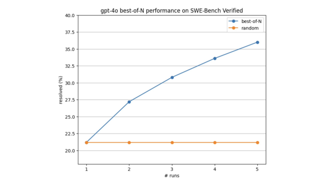
AlphaCode [Li et al., 2022]

Approach 3: scale *test-time* compute | How?

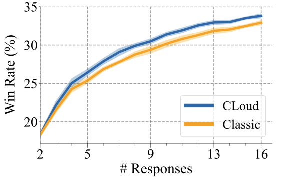
1. Generate extra tokens
2. Call generator multiple times



Math [Brown et al., 2024]



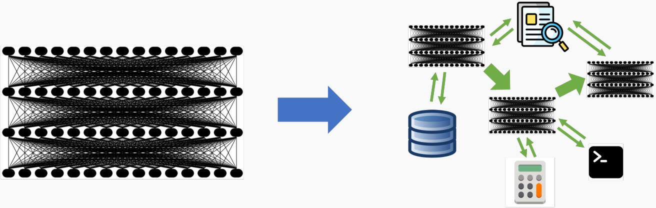
Agents [Nebius, 2024]



Chat [Ankner et al., 2024]

Approach 3: scale *test-time* compute | How?

1. Generate extra tokens
2. Call generator multiple times
3. Incorporate other models/tools



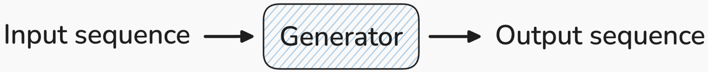
[Zaharia et al., 2024]

Verifiers, code interpreters, search engines, ...

This tutorial

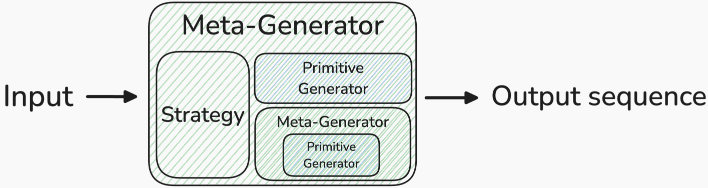
This tutorial: How? *Meta-Generation Algorithms*

Generator: Generates a sequence with a language model.



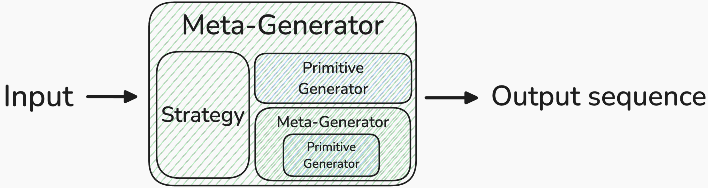
- Example: calling an LLM API
- Traditional algorithms
  - Greedy decoding
  - Temperature sampling
  - ...

Meta-generator: High-level strategies for calling generators and using external information.



- Example: call API multiple times, select the best sequence with a separate model

Meta-generator: High-level strategies for calling generators and using external information.



Why?

- Generate more to improve task performance
- Combine multiple models (verifiers, retrievers, . . .)
- Incorporate external information (tools, feedback, . . .)

This tutorial

Beyond Decoding: Meta-Generation Algorithms for LLMs

- I: **Primitive generators:** Generating one token at a time
- II: **Meta-generators:** High-level strategies for calling generators
- III: **Efficient meta-generation:** Generating quickly and efficiently

Panel session at the end!

Presenters

Part I

Matthew Finlayson
USC
@mattf1n

Intro/Part II

Sean Welleck
CMU
@wellecks

Part III

Hailey Schoelkopf
EleutherAI
@haileysch__

Panel

Beidi Chen
CMU
@BeidiChen

Nouha Dziri
AI2
@nouhadziri

Rishabh Agarwal
DeepMind/McGill
@agarwl_

Jakob Foerster
Oxford/Meta AI
@j_foerst

Noam Brown
OpenAI
@polynoamial

Ilia Kulikov (Moderator)
Meta AI
@uralik1

This tutorial | team and survey

Neurips 2024 Tutorial:
Beyond Decoding: Meta-Generation Algorithms for Large Language Models





Sean Welleck[1] Amanda Bertsch[1] Matthew Finlayson[2] Alex Xie[1] Graham Neubig[1]




Konstantin Golobokov[5] Hailey Schoelkopf[3] Ilia Kulikov[4] Zaid Harchaoui[5]

[1]Carnegie Mellon University [2]University of Southern California [3]Work done while at EleutherAI [4]Meta AI

[5]University of Washington

Survey (TMLR 2024): *From Decoding to Meta-Generation: Inference-time Algorithms for Large Language Models* [Welleck et al., 2024]

This tutorial | resources

cmu-l3.github.io/neurips2024-inference-tutorial

Code examples, reading list, slides

References i

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985).
A learning algorithm for boltzmann machines.
Cognitive Science, 9(1):147–169.

Adams, G., Ladhak, F., Schoelkopf, H., and Biswas, R. (2024).
Cold compress: A toolkit for benchmarking kv cache compression approaches.

Aggarwal, P., Parno, B., and Welleck, S. (2024).
Alphaverus: Bootstrapping formally verified code generation through self-improving translation and treefinement.
`https://arxiv.org/abs/2412.06176`.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. (2023).
Gqa: Training generalized multi-query transformer models from multi-head checkpoints.

Ankner, Z., Paul, M., Cui, B., Chang, J. D., and Ammanabrolu, P. (2024).
Critique-out-loud reward models.

Asai, A., He⋆, J., Shao⋆, R., Shi, W., Singh, A., Chang, J. C., Lo, K., Soldaini, L., Feldman, Tian, S., Mike, D., Wadden, D., Latzke, M., Minyang, Ji, P., Liu, S., Tong, H., Wu, B., Xiong, Y., Zettlemoyer, L., Weld, D., Neubig, G., Downey, D., Yih, W.-t., Koh, P. W., and Hajishirzi, H. (2024).
OpenScholar: Synthesizing scientific literature with retrieval-augmented language models.

Arxiv.

Basu, S., Ramachandran, G. S., Keskar, N. S., and Varshney, L. R. (2021).
Mirostat: a neural text decoding algorithm that directly controls perplexity.
In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Bertsch, A., Xie, A., Neubig, G., and Gormley, M. (2023).
It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk.
In Elazar, Y., Ettinger, A., Kassner, N., Ruder, S., and A. Smith, N., editors, *Proceedings of the Big Picture Workshop*, pages 108–122, Singapore. Association for Computational Linguistics.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. (2024).
Large language monkeys: Scaling inference compute with repeated sampling.
https://arxiv.org/abs/2407.21787.

Chen, J., Tiwari, V., Sadhukhan, R., Chen, Z., Shi, J., Yen, I. E.-H., and Chen, B. (2024a).
Magicdec: Breaking the latency-throughput tradeoff for long context generation with speculative decoding.

Chen, X., Lin, M., Schärli, N., and Zhou, D. (2024b).
Teaching large language models to self-debug.
In *The Twelfth International Conference on Learning Representations*.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022).
Scaling instruction-finetuned language models.
https://arxiv.org/abs/2210.11416.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. (2021).
Training verifiers to solve math word problems.
https://arxiv.org/abs/2110.14168.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and R'e, C. (2022).
Flashattention: Fast and memory-efficient exact attention with io-awareness.
ArXiv preprint, abs/2205.14135.

Dohan, D., Xu, W., Lewkowycz, A., Austin, J., Bieber, D., Lopes, R. G., Wu, Y., Michalewski, H., Saurous, R. A., Sohl-dickstein, J., Murphy, K., and Sutton, C. (2022).
Language model cascades.
`https://arxiv.org/abs/2207.10342`.

Fan, A., Lewis, M., and Dauphin, Y. (2018).
Hierarchical neural story generation.
In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898. Association for Computational Linguistics.

Fedus, W., Zoph, B., and Shazeer, N. (2022).
Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.

Feng, G., Zhang, B., Gu, Y., Ye, H., He, D., and Wang, L. (2023).
Towards revealing the mystery behind chain of thought: A theoretical perspective.
In *Thirty-seventh Conference on Neural Information Processing Systems*.

Finlayson, M., Hewitt, J., Koller, A., Swayamdipta, S., and Sabharwal, A. (2024).
Closing the curious case of neural text degeneration.
In *The Twelfth International Conference on Learning Representations*.

Freitag, M. and Al-Onaizan, Y. (2017).
Beam search strategies for neural machine translation.
In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60. Association for Computational Linguistics.

He, H. (2022).
Making deep learning go brrrr from first principles.

Hewitt, J., Manning, C., and Liang, P. (2022).
Truncation sampling as language model desmoothing.
In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427. Association for Computational Linguistics.

Hobbhahn, M., Heim, L., and Aydos, G. (2023).
Trends in machine learning hardware.
Accessed: 2024-11-26.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020).
The curious case of neural text degeneration.
In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., and Zhou, D. (2024).
Large language models cannot self-correct reasoning yet.
In *The Twelfth International Conference on Learning Representations*.

Jiang, A. Q., Welleck, S., Zhou, J. P., Lacroix, T., Liu, J., Li, W., Jamnik, M., Lample, G., and Wu, Y. (2023).
Draft, sketch, and prove: Guiding formal theorem provers with informal proofs.
In *The Eleventh International Conference on Learning Representations*.

Juravsky, J., Brown, B., Ehrlich, R., Fu, D. Y., Ré, C., and Mirhoseini, A. (2024).
Hydragen: High-throughput llm inference with shared prefixes.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020).
Scaling laws for neural language models.
https://arxiv.org/abs/2001.08361.

- Khattab, O., Santhanam, K., Li, X. L., Hall, D. L. W., Liang, P., Potts, C., and Zaharia, M. A. (2022).
  Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp.
  ArXiv, abs/2212.14024.
- Kim, S., Suk, J., Longpre, S., Lin, B. Y., Shin, J., Welleck, S., Neubig, G., Lee, M., Lee, K., and Seo, M. (2024).
  Prometheus 2: An open source language model specialized in evaluating other language models.
  `https://arxiv.org/abs/2405.01535`.
- Koh, J. Y., McAleer, S., Fried, D., and Salakhutdinov, R. (2024).
  Tree search for language model agents.
  arXiv preprint arXiv:2407.01476.

Kudo, T. (2018).
Subword regularization: Improving neural network translation models with multiple subword candidates.
In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Kumar, A., Zhuang, V., Agarwal, R., Su, Y., Co-Reyes, J. D., Singh, A., Baumli, K., Iqbal, S., Bishop, C., Roelofs, R., Zhang, L. M., McKinney, K., Shrivastava, D., Paduraru, C., Tucker, G., Precup, D., Behbahani, F., and Faust, A. (2024).
Training language models to self-correct via reinforcement learning.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. (2023).
Efficient memory management for large language model serving with pagedattention.

Li, X. L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., and Lewis, M. (2023a).
Contrastive decoding: Open-ended text generation as optimization.
In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312. Association for Computational Linguistics.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Lago, A. D., Hubert, T., Choy, P., de Masson d'Autume, C., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Cherepanov, A., Molloy, J., Mankowitz, D. J., Robson, E. S., Kohli, P., de Freitas, N., Kavukcuoglu, K., and Vinyals, O. (2022).
Competition-level code generation with alphacode.
Science, 378(6624):1092–1097.

Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., and Chen, W. (2023b).
Making language models better reasoners with step-aware verifier.
In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. (2024).
Let's verify step by step.
In *The Twelfth International Conference on Learning Representations*.

Liu, A., Han, X., Wang, Y., Tsvetkov, Y., Choi, Y., and Smith, N. A. (2024).
Tuning language models by proxy.

Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N. A., and Choi, Y. (2021).
DExperts: Decoding-time controlled text generation with experts and anti-experts.
In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706. Association for Computational Linguistics.

Lu, X., Brahman, F., West, P., Jung, J., Chandu, K., Ravichander, A., Ammanabrolu, P., Jiang, L., Ramnath, S., Dziri, N., Fisher, J., Lin, B., Hallinan, S., Qin, L., Ren, X., Welleck, S., and Choi, Y. (2023). **Inference-time policy adapters (IPA): Tailoring extreme-scale LMs without fine-tuning.** In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6863–6883. Association for Computational Linguistics.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., and Clark, P. (2023).
Self-refine: Iterative refinement with self-feedback.
In *Thirty-seventh Conference on Neural Information Processing Systems.*

Meister, C., Cotterell, R., and Vieira, T. (2020).
If beam search is the answer, what was the question?
In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185. Association for Computational Linguistics.

Meister, C., Pimentel, T., Wiher, G., and Cotterell, R. (2022).
Locally typical sampling.
Transactions of the Association for Computational Linguistics, 11:102–121.

Meister, C., Pimentel, T., Wiher, G., and Cotterell, R. (2023).
Locally typical sampling.
Transactions of the Association for Computational Linguistics, 11:102–121.

Merrill, W. and Sabharwal, A. (2024).
The expressive power of transformers with chain of thought.
In *The Twelfth International Conference on Learning Representations*.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. (2022).
Webgpt: Browser-assisted question-answering with human feedback.
https://arxiv.org/abs/2112.09332.

Nebius (2024).
Leveraging training and search for better software engineering agents.
https://nebius.com/blog/posts/training-and-search-for-software-engineering-agents.

Nowak, F., Svete, A., Butoi, A., and Cotterell, R. (2024).
On the representational capacity of neural language models with chain-of-thought reasoning.
In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12510–12548, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI (2024).
Learning to reason with llms.
`https://openai.com/index/learning-to-reason-with-llms/`.

Polu, S. and Sutskever, I. (2020).
Generative language modeling for automated theorem proving.

- Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N., and Lewis, M. (2023).
  Measuring and narrowing the compositionality gap in language models.
  In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711. Association for Computational Linguistics.
- Schlag, I., Sukhbaatar, S., Celikyilmaz, A., tau Yih, W., Weston, J., Schmidhuber, J., and Li, X. (2023).
  Large language model programs.
  `https://arxiv.org/abs/2305.05364`.
- Shazeer, N. (2019).
  Fast transformer decoding: One write-head is all you need.

Shi, C., Yang, H., Cai, D., Zhang, Z., Wang, Y., Yang, Y., and Lam, W. (2024).
A thorough examination of decoding methods in the era of LLMs.
In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8601–8629, Miami, Florida, USA. Association for Computational Linguistics.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016).

Mastering the game of go with deep neural networks and tree search.
Nature, 529:484–503.

Stahlberg, F. and Byrne, B. (2019).
On nmt search errors and model errors: Cat got your tongue?
ArXiv, abs/1908.10090.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020).
Learning to summarize with human feedback.
In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Sun, Z., Yu, L., Shen, Y., Liu, W., Yang, Y., Welleck, S., and Gan, C. (2024).
Easy-to-hard generalization: Scalable alignment beyond human supervision.
In *The Thirty-eighth Annual Conference on Neural Information Processing Systems.*

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. (2022).
Solving math word problems with process- and outcome-based feedback.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. (2024a).
Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations.
In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. (2023).
Self-consistency improves chain of thought reasoning in language models.
In *The Eleventh International Conference on Learning Representations*.

- Wang, Y., Wu, Y., Wei, Z., Jegelka, S., and Wang, Y. (2024b).
A theoretical understanding of self-correction through in-context alignment.
https://arxiv.org/abs/2405.18634.

- Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022).
Chain of thought prompting elicits reasoning in large language models.
In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Welleck, S., Bertsch, A., Finlayson, M., Schoelkopf, H., Xie, A., Neubig, G., Kulikov, I., and Harchaoui, Z. (2024).
From decoding to meta-generation: Inference-time algorithms for large language models.
https://arxiv.org/abs/2406.16838.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. (2020).
Neural text generation with unlikelihood training.
In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Welleck, S., Lu, X., West, P., Brahman, F., Shen, T., Khashabi, D., and Choi, Y. (2023).
Generating sequences by learning to self-correct.
In *The Eleventh International Conference on Learning Representations*.

Weston, J. and Sukhbaatar, S. (2023).
System 2 attention (is something you might need too).

Wu, I., Fernandes, P., Bertsch, A., Kim, S., Pakazad, S., and Neubig, G. (2024a).
Better instruction-following through minimum bayes risk.
`https://arxiv.org/abs/2410.02902`.

Wu, Y., Sun, Z., Li, S., Welleck, S., and Yang, Y. (2024b).
Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models.
https://arxiv.org/abs/2408.00724.

Xia, H., Yang, Z., Dong, Q., Wang, P., Li, Y., Ge, T., Liu, T., Li, W., and Sui, Z. (2024).
Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding.

Zaharia, M., Khattab, O., Chen, L., Davis, J. Q., Miller, H., Potts, C., Zou, J., Carbin, M., Frankle, J., Rao, N., and Ghodsi, A. (2024).
The shift from models to compound ai systems.
https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/.

Zhang, L., Hosseini, A., Bansal, H., Kazemi, M., Kumar, A., and Agarwal, R. (2024).
Generative verifiers: Reward modeling as next-token prediction.

Zheng, L., Yin, L., Xie, Z., Sun, C., Huang, J., Yu, C. H., Cao, S., Kozyrakis, C., Stoica, I., Gonzalez, J. E., Barrett, C., and Sheng, Y. (2024).
Sglang: Efficient execution of structured language model programs.