Improving Language Modeling by Increasing Test-time Planning Compute

Florian Mai, Nathan Cornille, Marie-Francine Moens
Department of Computer Science
KU Leuven

Abstract

Modern language models predict the next token in the sequence by considering the past text through a powerful function. However, language models have no explicit mechanism that allows them to spend computation time for planning long-distance future text, leading to a suboptimal token prediction. In this paper, we propose a planner that predicts a latent plan for many sentences into the future. By sampling multiple plans at once, we condition the language model on an accurate approximation of the distribution of text continuations, which leads to better next token prediction accuracy. In effect, this allows trading computation time for prediction accuracy.

1 Introduction

Current LLMs can be considered *fast, intuitive reasoners* (Bengio et al., 2021), analogous to the type 1 reasoning systems found in humans according to the dual-process theory (Evans, 1984; Kahneman, 2011). System 1 allows solving intuitive tasks such as perception and talking, but it is insufficient for tasks that require planning, such as writing coherent, long stretches of text. For planning tasks, humans instead invoke a *slow, deliberate* reasoning system 2. Most works that attempt to integrate deliberate planning and reasoning ability into LLMs pose the problem as a *post-training* process: by finetuning on reasoning datasets (Hendrycks et al., 2021; Havrilla et al., 2024), by learning to invoke external task-specific planners (Schick et al., 2023; Nye et al., 2021), or by employing advanced prompting methods like Chain-of-Thought (Wei et al., 2022). However, neuroscientific studies have revealed that *predictive coding*, the ability to continuously predict, update and draw on multiple hypotheses about future inputs, is central to language learning and production (Casillas and Frank, 2013; Ylinen et al., 2017; Shain et al., 2020; Aitchison and Lengyel, 2017; Kellogg, 2013; Mallahi, 2019). The working memory plays a critical role by providing a cognitive scratch pad to store a few relevant concepts (Cowan, 2001) and continuously update them based on the ongoing cognitive task (Cashdollar et al., 2017), particularly in the context of language production and writing (Casillas and Frank, 2013; Kellogg, 2013; Mallahi, 2019). This suggests that the ability to plan originates, at least in part, from learning from unlabeled data and should hence be fostered in LLMs *during pre-training*. Cornille et al. (2024) propose a pretraining method in which language modeling is factorized into 1) first predicting a high-level latent plan via a separate planner module and 2) then conditioning the language model on generated plans when predicting the next token. However, their method only predicts a single one-step plan, which predicts merely one sentence ahead. As such, it neither performs long-term planning nor allows to draw on multiple hypotheses through variable compute. In this paper, we propose an extension of the framework by Cornille et al. (2024) through two crucial changes (Figure 1): 1) We learn a planner that predicts multiple steps ahead to enable long-term predictive coding. 2) We sample a variable amount of hypotheses from the planner to condition the language model on, allowing to trade off computation time for better prediction accuracy.

2 Method

The key idea of the method is to transform an unlabeled text corpus into sequences of abstract writing actions and use these actions to guide the language model. Our method consists of three steps (cmp. Figure 1): ①: Inferring action sequences from unlabeled texts. ②: Training a multi-step planner to predict the next actions. ③: Sampling multiple paths from the planner to condition the LM.

①: Following Cornille et al. (2024), we chunk each text into sentences, embed them, and map

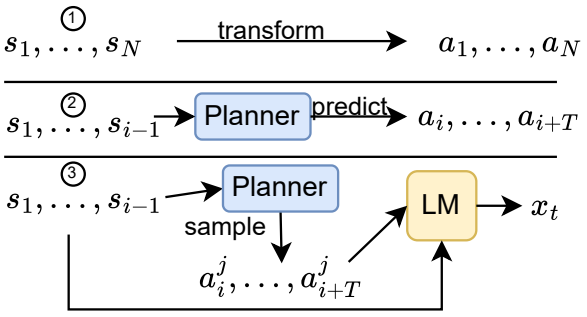

Figure 1: Overview of our method.

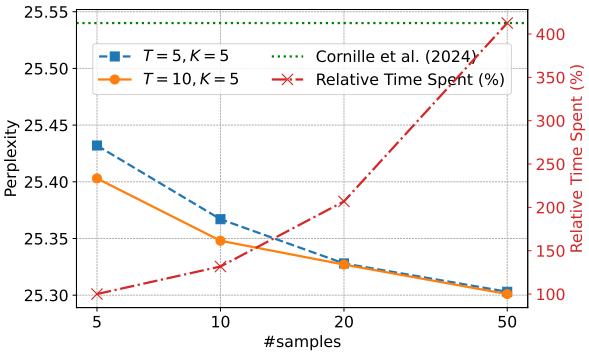

Figure 2: Performance and relative generation time as a function of the number of samples K drawn.

them to the nearest action embedding. Action embeddings are obtained via K-means clustering on the set of embedded sentences from the corpus.

②: Inspired by the MuZero architecture (Schrittwieser et al., 2020), we propose a multi-step planner consisting of three components. The *representation function* encodes the current context into a multi-vector state representation, where each vector corresponds to one sentence in the input. Given the current state, the *prediction function* predicts a probability distribution over the actions. Finally, given the current state and the predicted action, the *dynamics function* predicts the representation of the next state.

③: After the planner training is completed, we condition the language model on planner outputs as follows. First, we obtain K different paths of length T by repeatedly sampling from the planner. Second, we encode each path independently using a single-layer Transformer encoder and averaging the output representations into a single vector path representation. Third, we use another Transformer encoder to contextualize and aggregate the path representations, yielding a single vector to represent all planner predictions. Last, it is merged into the LM via a LLaMA-adapter (Zhang et al., 2023).

3 Experiments

The purpose of our experiments is to demonstrate the benefit of our contributions for language modeling: 1) Multi-step planning and 2) conditioning on multiple sampled plans.

Baselines and metrics Cornille et al. (2024) can be viewed as a special case of our model with $T = 1, K = 1$ and no additional Transformers aggregating across time and paths. It thus serves as our primary baseline. Following Cornille et al. (2024), all experiments are performed based on GPT-2 small (128M parameters) finetuned on 285310 articles of English Wikipedia. All models are evaluated via the perplexity metric.

Results In Figure 2, we show the results of two models with $T = 5, 10$ and $K = 5$, respectively. We increase the number of sampled paths K *at inference time only*. This experiment demonstrates that a) our method improves over the single-step baseline by Cornille et al. (2024), b) predicting multiple steps ahead is advantageous, and c) performance continues to improve until at least $K = 50$. Naturally, this comes at the expense of additional compute.

4 Discussion & Conclusion

Our consistent improvements in perplexity indicate that both integrating long-term predictions of the future writing process and modeling multiple future paths provide an LM with information that is valuable even for making local predictions. Consequently, our model outperforms the single-step planner by Cornille et al. (2024).

Moreover, a core motivation of our work is to allow a language model to spend additional test-time compute to improve its predictions, similar to how AlphaGo (Silver et al., 2017) uses a lot of inference-time compute to achieve superhuman performance in Go. Demonstrating that our model, too, can trade off compute for better performance, we take a first step towards enabling this property for LMs, opening exciting research directions.

Future work will investigate how to learn actions that are tailored for language modeling, how to learn when to invoke the planner to minimze compute overhead, and how to improve the controllability of the language model through planner finetuning.

References

Laurence Aitchison and Máté Lengyel. 2017. With or without you: predictive coding and bayesian inference in the brain. *Current opinion in neurobiology*, 46:219–227.

Yoshua Bengio, Yann LeCun, and Geoffrey E. Hinton. 2021. Deep learning for AI. *Commun. ACM*, 64(7):58–65.

Nathan Cashdollar, Philipp Ruhnau, Nathan Weisz, and Uri Hasson. 2017. The role of working memory in the probabilistic inference of future sensory events. *Cerebral cortex*, 27(5):2955–2969.

Marisa Casillas and Michael Frank. 2013. The development of predictive processes in children's discourse understanding. In *Proceedings of the annual meeting of the Cognitive Science Society*, volume 35.

Nathan Cornille, Marie-Francine Moens, and Florian Mai. 2024. Learning to plan for language modeling from unlabeled data. In *First Conference on Language Modeling*.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114.

Jonathan St BT Evans. 1984. Heuristic and analytic processes in reasoning. *British Journal of Psychology*, 75(4):451–468.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Macmillan.

Ronald T Kellogg. 2013. A model of working memory in writing. In *The science of writing*, pages 57–71. Routledge.

Omid Mallahi. 2019. The role of working memory (wm) in fluency, accuracy and complexity of argumentative texts produced by iranian efl learners. *Iranian Journal of Learning & Memory*, 2(5):55–65.

Maxwell I. Nye, Michael Henry Tessler, Joshua B. Tenenbaum, and Brenden M. Lake. 2021. Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning. In *NeurIPS*, pages 25192–25204.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nat.*, 588(7839):604–609.

Cory Shain, Idan Asher Blank, Marten van Schijndel, William Schuler, and Evelina Fedorenko. 2020. fmri reveals language-specific predictive coding during naturalistic sentence comprehension. *Neuropsychologia*, 138:107307.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv preprint*. ArXiv:1712.01815 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Sari Ylinen, Alexis Bosseler, Katja Junttila, and Minna Huotilainen. 2017. Predictive coding accelerates word recognition and learning in the early stages of language development. *Developmental science*, 20(6):e12472.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023. LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention. *arXiv preprint*. ArXiv:2303.16199 [cs].