Language Model Behavioral Phases are Consistent Across Architecture, Training Data, and Scale

James A. Michaelov[1], Roger P. Levy[1], Benjamin K. Bergen[2]

[1]Massachusetts Institute of Technology, [2]UC San Diego

Introduction

Apparently complex behaviors emerge in language models during pretraining, despite the relatively simple task of next-word prediction. We ask:

1. To what extent do language model predictions reflect simple heuristics?
2. Do training dynamics tend to follow the same path, and how consistent is it:
   - Across scale?
   - Across training dataset?
   - Across architecture?

Language Models

Pythia (Biderman et al., 2023)

- Trained on The Pile: 143k steps of ~2M tokens
- Models:
  - 10 seeds each: 14M, 31M, 70M, 160M, 410M parameters
  - 1 seed each: 1B, 1.4B, 2.7B, 6.9B, 12B

Open GPT-2 (Karamcheti et al., 2021)

- Trained on OpenWebText: 400k steps of ~0.5M tokens
- Models:
  - 4 seeds each: 117M, 345M

Parallel architecture (Parc) Language Models (ours)

- Trained on OpenWebText: 4k steps of ~0.5M tokens
- Models:
  - 6 seeds each: Pythia 160M, Mamba 130M RWKV 169M

Heuristics

N-gram Log-Probability

- Transformers learn n-gram probabilities of increasing order over the course of pretraining (Chang et al., 2024; Belrose et al., 2024).
- We calculate n-gram probabilities on both text corpora using infini-gram (Liu et al., 2024) word counts with the *Stupid Backoff* method (Brants et al., 2007), which is described in (1), with (2) describing the unigram case.

$$\hat{p}(w_i|w_{i-n+1}^{i-1}) = \begin{cases} \frac{c(w_{i-n+1}^{i})}{c(w_{i-n+1}^{i-1})}, & \text{if } c(w_{i-n+1}^{i}) > 0 \\ \alpha\hat{p}(w_i|w_{i-n+1}^{i-1}), & \text{otherwise} \end{cases} \quad (1) \qquad \hat{p}(w_i) = \frac{\max\{1, c(w_i)\}}{|C|} \quad (2)$$

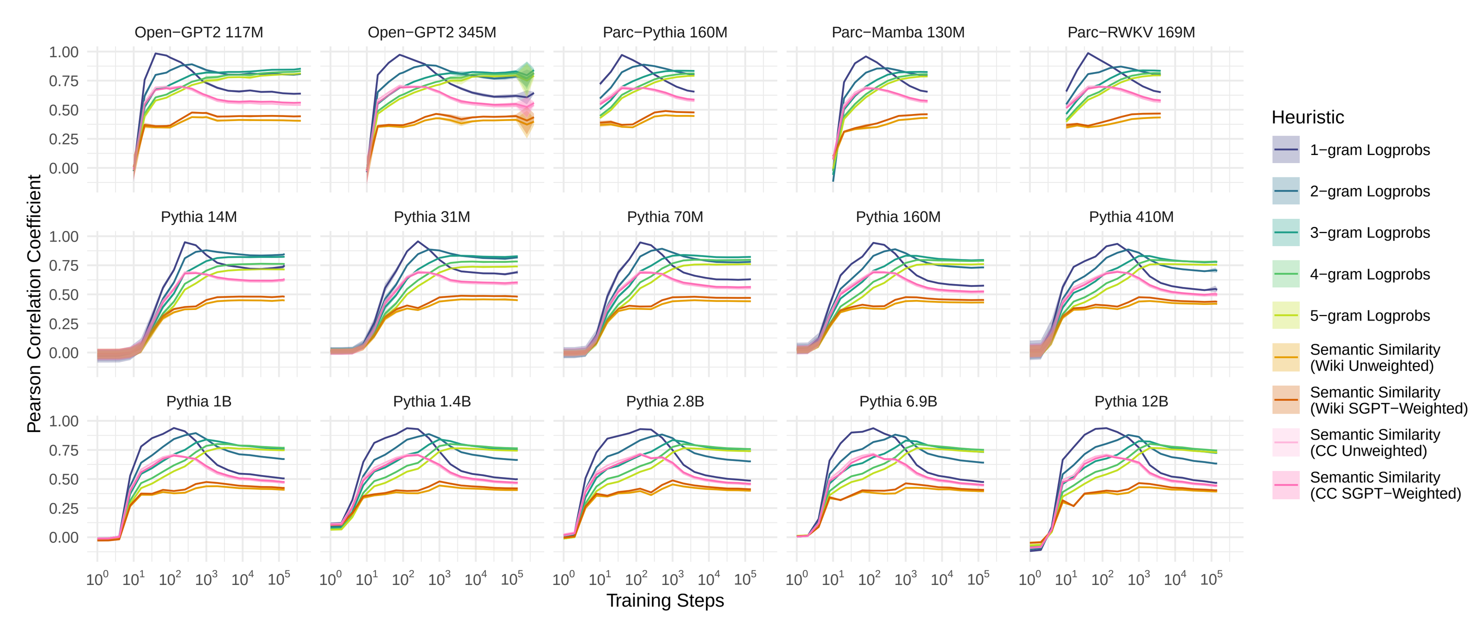
Contextual Semantic Similarity

- Previous work shows a correlation between LM log-probability and semantic relatedness to previous words (Michaelov et al., 2024).
- We calculate this as the cosine similarity between the *fastText* (Bojanowski et al., 2017) vector of a word w_i and its context c, as described in (3), using either uniform weighting (4) or SGPT (Muennighoff, 2022) weighting (5).

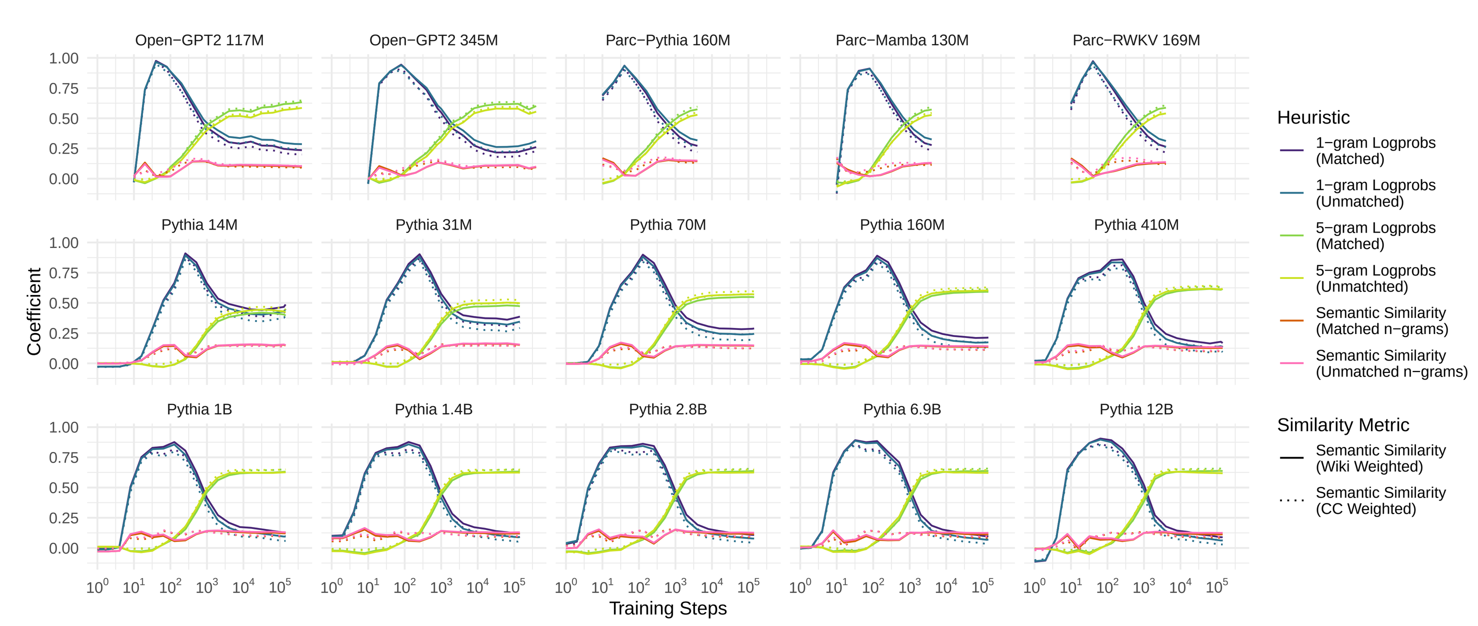
$$\vec{c} = \sum_{j=1}^{j=i-1} \beta_j \vec{w_j} \quad (3) \qquad \beta = \frac{1}{i-1} \quad (4) \qquad \beta = \frac{j}{\sum_{k=1}^{k=i-1} k} \quad (5)$$

Results

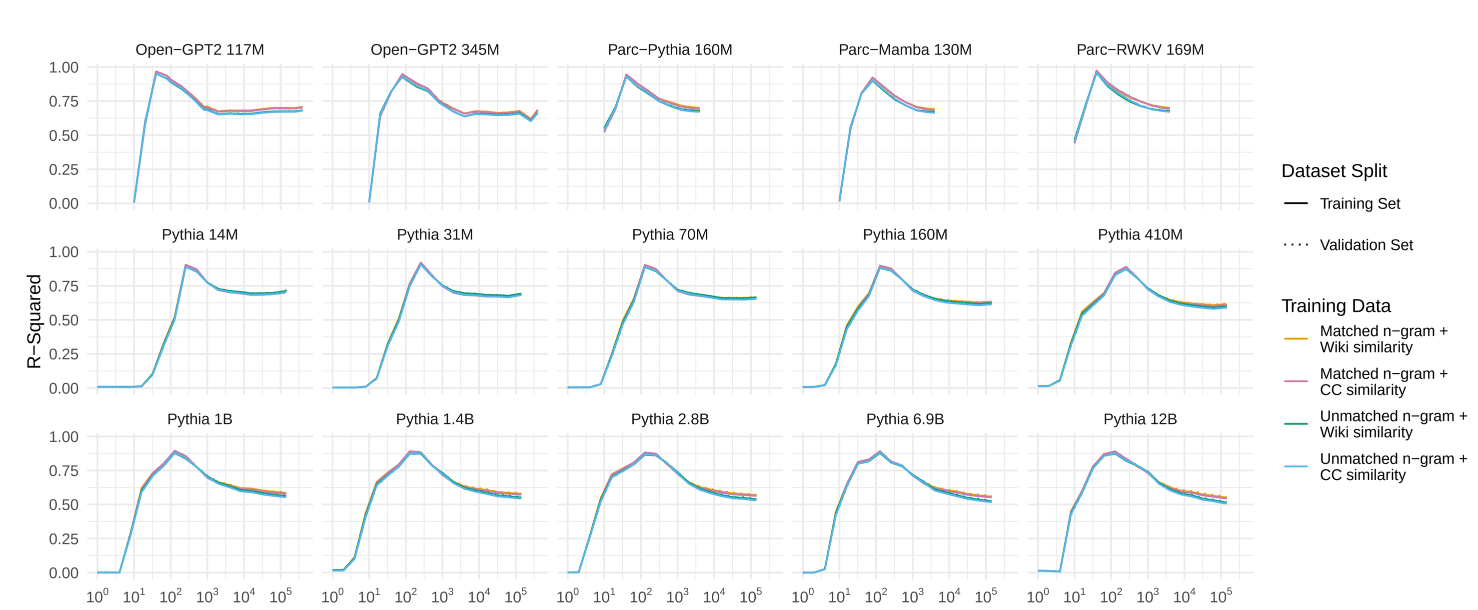
Pearson correlation (r) between language model log-probability, n-gram log-probability, and contextual semantic similarity



Coefficients of linear regressions predicting language model log-probability with unigram log-probability, 5-gram log-probability, and contextual semantic similarity



Variance explained (R^2) by linear regressions predicting language model log-probability with unigram log-probability, 5-gram log-probability, and contextual semantic similarity



Test Dataset: NaWoCo

We created the Natural Words in Context dataset:

- Single-token words (for Parc, Pythia, Open GPT2) in sentence contexts with 4+ preceding words sampled from FineWeb (Penedo et al., 2024).
- All sentences were classified as non-toxic, began with a capitalized letter, contained no other capitalized words, and did not occur in The Pile or OpenWebText (to avoid contamination).
- Training set: 77,999 words; Validation set: 39,474; Test set: 40,980.

Summary of Results

- Consistent pattern across all architectures: LM log-probability correlates most with unigram log-probability, then bigram, trigram, etc.
- Consistent pattern in coefficients: initial rapid increase in unigram log-probability coefficient, which then decreases as 5-gram coefficient increases (but not to zero).
- Variable but consistently-present coefficient of contextual similarity.
- Little difference between n-gram corpora, between contextual similarity weighting method, or between the fit to the training and validation set.
- R^2 rises with correlation to (and coefficient of) unigram log-probability up to 98%; and remains above 50% for all model checkpoints.

Discussion and Future Work

- To what extent do these heuristics explain more 'complex' LM behaviors?
- All language models tested follow the same trajectory, but must they?
  - Benchmark performance mostly increases after 5-grams are learned. To what extent is this a necessary step?
  - Representing in-context n-grams is important for in-context learning, and may be related learning 'global' n-grams (Bietti et al., 2023).
- Even models trained on much more data still show a susceptibility to over-predicting common sequences and words that are related to their context—can this approach be used to predict this propensity in a given model?

References

- Belrose et al. (2023). 'Neural Networks Learn Statistics of Increasing Complexity' ICML
- Biderman et al. (2023). 'Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling' ICML
- Bietti et al. (2023). 'Birth of a Transformer: A Memory Viewpoint' NeurIPS
- Bojanowski et al. (2017). 'Enriching Word Vectors with Subword Information' TACL
- Brants et al. (2007). 'Large Language Models in Machine Translation' EMNLP
- Chang et al. (2024). 'Characterizing Learning Curves During Language Model Pre-Training: Learning, Forgetting, and Stability' TACL
- Gao et al. (2020). 'The Pile: An 800GB Dataset of Diverse Text for Language Modeling' arXiv
- Gokaslan and Cohen (2019). 'OpenWebText corpus' http://Skylion007.github.io/OpenWebTextCorpus
- Karamcheti et al. (2021). 'Mistral — a journey towards reproducible language model training'. https://crfm.stanford.edu/2021/08/26/mistral.html
- Liu et al. (2024). 'Infini-gram: Scaling Unbounded n-gram Language Models to a Trillion Tokens' COLM
- Michaelov et al. (2024). 'Strong Prediction: Language Model Surprisal Explains Multiple N400 Effects' Neurobiology of Language
- Penedo et al. (2024). 'The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale' NeurIPS Datasets & Benchmarks