The Inverse Scaling Effect of Pre-Trained Language Model Surprisal Is Not Due to Data Leakage

Byung-Doh Oh
Center for Data Science
New York University
oh.b@nyu.edu

Hongao Zhu
School of Foreign Languages
Shanghai Jiao Tong University
cisco_sfl_sjtu@sjtu.edu.cn

William Schuler
Department of Linguistics
The Ohio State University
schuler.77@osu.edu

Abstract

In psycholinguistic modeling, surprisal from larger pre-trained language models has been shown to be a poorer predictor of naturalistic human reading times. However, it has been speculated that this may be due to data leakage that caused language models to see the text stimuli during training. This paper presents two studies to address this concern at scale. The first study reveals relatively little leakage of five naturalistic reading time corpora in two pre-training datasets in terms of length and frequency of token n-gram overlap. The second study replicates the negative relationship between language model size and the fit of surprisal to reading times using models trained on 'leakage-free' data that overlaps only minimally with the reading time corpora. Taken together, this suggests that previous results using language models trained on these corpora are not driven by the effects of data leakage.

1 Introduction

Language models (LMs) based on neural networks, which are trained to predict upcoming words, have been shown to flexibly capture many linguistic regularities from raw text (Linzen and Baroni, 2021; Mahowald et al., 2024). This has sparked research at the intersection between language modeling and psycholinguistics that relates LM probabilities to human behavior. One line of such research focuses on evaluating LM surprisal (negative log probabilities; Shannon, 1948) against measures of processing difficulty such as word-by-word reading times, under an 'expectation-based' theoretical link that posits predictability as a key determinant of processing difficulty (Hale, 2001; Levy, 2008).

However, the source stimuli of the reading time datasets (e.g. Futrell et al., 2021; Luke and Christianson, 2018) used in such studies are often naturalistic text that are available online (e.g. news articles), which raises the concern that those texts may occur in the LMs' pre-training corpora. If the degree of such data leakage is severe, the LMs may assign artificially lower surprisal to the text in reading time datasets as a result of having 'memorized' it during training. As a consequence, this could bring into question the validity of previous results as well as the general practice of using pre-trained LMs in psycholinguistic modeling. For example, it has been speculated that the negative relationship between the size of an LM and the fit of its surprisal to human reading times observed on English data (e.g. Oh and Schuler, 2023b; Shain et al., 2024) may be due to such leakage (Wilcox et al., 2023a).

This work presents two studies to address this concern at scale. The first study assesses the leakage of five naturalistic reading time corpora in two pre-training datasets that were each used to train Pythia and GPT-2 LMs (Biderman et al., 2023; Radford et al., 2019) by identifying the longest overlapping token sequence and its frequency. The second study then uses the same methodology to curate training data that overlaps minimally with the reading time corpora, and trains LMs on it to examine whether the negative relationship between model size and the fit of LM surprisal is observed with 'leakage-free' training data. Additionally, data leakage is artificially introduced through fine-tuning to study how LM surprisal's fit to reading times would change in the face of severe leakage.

The results indicate that commonly used reading time corpora suffer little from data leakage, with most passages sharing only relatively short overlaps among the billions of tokens in the two pre-training corpora. Moreover, LMs trained on leakage-free data robustly replicate the negative relationship between model size and surprisal's fit to reading times, further indicating that this phenomenon is not simply due to leakage. However, results also show that actual severe leakage is likely to result in an overestimation of this negative relationship, which still warrants caution against the

Findings of the Association for Computational Linguistics: ACL 2025, pages 1820–1827

leakage of reading time corpora. Taken together, these results suggest that previous findings based on LMs trained on these corpora are not due to the effects of data leakage.

2 Study 1: Overlap Between Reading Time and Pre-Training Corpora

The first study assesses the leakage of naturalistic reading time corpora in LM pre-training datasets. To this end, Compacted Directed Acyclic Word Graphs (CDAWGs; Crochemore and Vérin, 1997; Inenaga et al., 2005) were built on two pre-training corpora, which allows the reading time corpora to be queried efficiently to identify the longest overlapping token sequence and its frequency.

2.1 Methods

Pre-Training Corpora. The two English pre-training corpora that were analyzed in this study are the subset of the Pile (Gao et al., 2020) that was used to train the Pythia LMs (Biderman et al., 2023), and the OpenWebText Corpus (Gokaslan and Cohen, 2019), which is an open-source replication of the GPT-2 LMs' (Radford et al., 2019) training data. The training data of the Pythia LMs is provided as pre-tokenized examples of length 2,049, which are sequences sampled from a concatenated version of the Pile. A total of 143,000 batches that each contain 1,024 training examples were used to train the Pythia LMs, which amounts to a total of ~300B tokens. The OpenWebText Corpus consists of 8,013,769 documents, which is equivalent to ~8.7B tokens when tokenized with Pythia LM's subword tokenizer.

Reading Time Corpora. The five English reading time corpora that served as queries are:

1. Dundee (Kennedy et al., 2003): 67 newspaper editorials from *The Independent*.
2. Brown (Smith and Levy, 2013): 13 passages from the Brown Corpus (Kučera and Francis, 1967).
3. GECO (Cop et al., 2017): 13 chapters from the novel *The Mysterious Affair at Styles* (Christie, 1920).
4. Provo (Luke and Christianson, 2018): 55 passages of news articles, science magazine excerpts, and fictional work.
5. Natural Stories (Futrell et al., 2021): 10 passages of narrative and expository text.

With the exception of the Dundee Corpus, most of the source text in these corpora are available online, which makes them susceptible to leakage in pre-training corpora. Additionally, while Natural Stories has been manually edited to include challenging syntactic constructions, there is still likely to be substantial overlap if the pre-training corpora contain the original source text.

CDAWG Construction and Querying. A CDAWG is a finite-state machine that is specialized for indexing sequences, which allows the length of the longest occurring suffix of a query to be returned efficiently. We use Merrill et al.'s (2024) implementation[1] to build CDAWGs on the two pre-training corpora after normalizing their line breaks and whitespaces and tokenizing them with Pythia LM's subword tokenizer. Training examples from the Pile were additionally split at <|endoftext|> tokens in order to treat text from different documents as separate sequences.

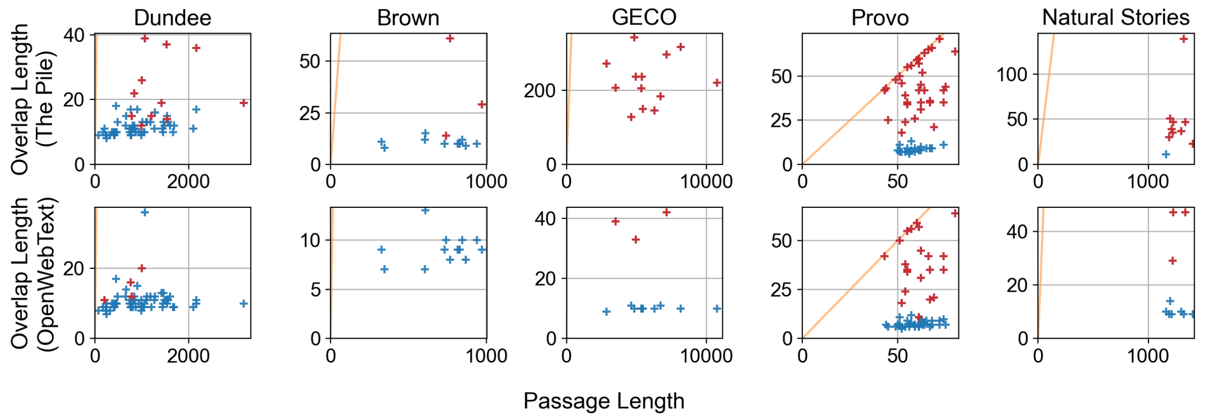
Subsequently, each passage of the five reading time corpora was tokenized using the same tokenizer and queried against the two CDAWGs. The length of the longest occurring suffix was then calculated at every token position to retrieve the globally longest overlapping token sequence between each passage and the pre-training corpora.[2] The frequency of this sequence in the pre-training corpora was also retrieved to further gauge the severity of this overlap. Additionally, the joint probability of this sequence was calculated using a 5-gram LM with backoff using the KenLM toolkit (Heafield et al., 2013) with parameters estimated on the Gigaword 4 corpus (Parker et al., 2009). This allows us to identify sequences that are likely to appear in a corpus of a given size at roughly chance level.[3]

While our method of detecting data leakage based on token sequence overlap is not robust against minor variations in surface form (e.g. paraphrases), using a 'softer' match criterion such as the similarity between sequence-level embeddings is computationally infeasible at the scale of the pre-

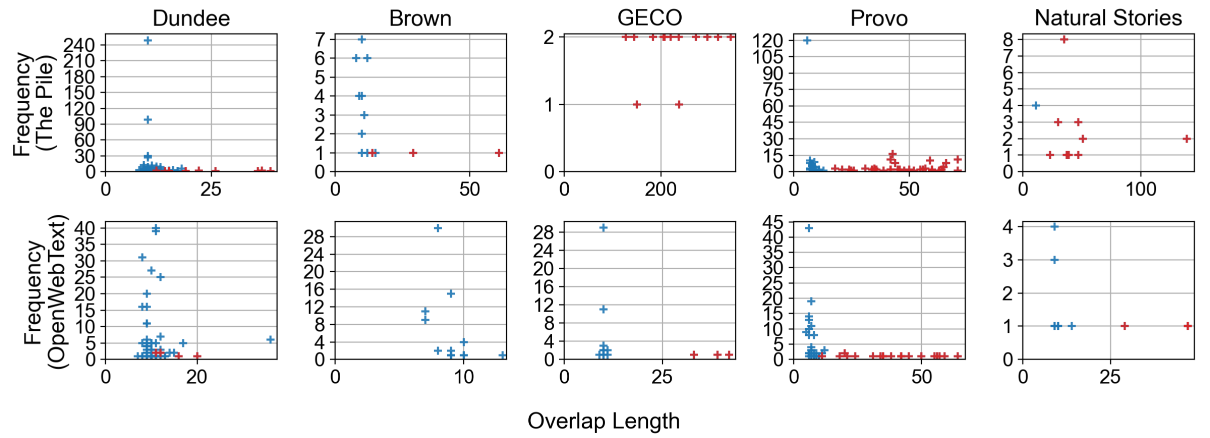
[1] `https://github.com/viking-sudo-rm/rusty-dawg`

[2] Borrowing the example of Merrill et al. (2024), querying `l l o y d` against the reference `h e l l o w o r l d` returns the lengths <1, 2, 3, 0, 1> at each token position, which allows the longest overlapping sequence `l l o` to be identified at token position 3.

[3] The probability of the sequence appearing at least once was estimated as $1 - (1 - p)^n$, where p is the probability of the sequence and n is the number of whitespace words in each corpus. The p that sets the probability of this event to some threshold can then be calculated for each pre-training corpus.



(a) The length of each passage in the reading time corpora, and the length of its longest overlapping sequence with the Pile (top) and the OpenWebText Corpus (bottom), both measured in the number of subword tokens. The orange line denotes the $y = x$ line that indicates complete overlap.



(b) The length and frequency of the longest overlapping sequence between each passage of the reading time corpora and the Pile (top) and the OpenWebText Corpus (bottom). When there are multiple overlaps with the same maximum length, the highest frequency is reported.

Figure 1: Overlap between the five reading time corpora and the two pre-training corpora. Each '+' represents one passage, and the red '+'s denote longest overlapping sequences that have a probability lower than 0.05 to appear at least once in each corpus by chance (i.e. sequences with 5-gram log probabilities lower than −28.87 and −25.33 for the Pile and OpenWebText respectively).

training datasets we study, and may yield unreliable results depending on the quality of the embeddings. Additionally, our method allows complete overlaps to be detected reliably, which similarity-based methods usually cannot allow when the lengths of the two sequences are different.

2.2 Results

Figure 1a shows that except for the Provo Corpus, no passage in the reading time corpora is observed entirely in both pre-training datasets. That is, the length of each passage's longest overlapping sequence is relatively short compared to the full passage length. While the Provo passages that are observed in their entirety or the longer overlapping sequences in the Pile that exceed 100 tokens may especially be concerning, Figure 1b shows that such instances are very infrequent. Most of the highly overlapping Provo passages each occur under 10 times in both pre-training corpora, and the longer overlapping sequences exceeding 100 tokens occur at most twice. Therefore, we interpret these results as indicating that the reading time corpora suffer little from data leakage.[4]

3 Study 2: The Influence of Leakage on Fit to Reading Times

The previous study shows that most passages of the reading time corpora have not been leaked in the two pre-training corpora, which suggests that

[4]We publicly release the longest overlapping sequences and their frequencies at https://github.com/byungdoh/rt-leakage.

Model	#L	#H	d_{model}	#Parameters
Small	3	4	256	28,125,440
Medium	6	8	512	70,426,624
Large	12	12	768	162,322,944

Table 1: Hyperparameters of LMs that were trained in this study. #L: number of layers; #H: number of attention heads per layer; d_{model}: embedding size.

Corpus/Measure	Fit	Exploratory
Brown SPR	59,292	29,671
Natural Stories SPR	384,984	192,826
Dundee FP	98,115	48,598
GECO FP	144,850	72,468
Provo FP	52,959	26,539

Table 2: Number of data points in the fit and exploratory partitions of each reading time dataset.

leakage is unlikely to be a possible explanation for the adverse effect of model size on LM surprisal's fit to reading times (Oh and Schuler, 2023b; Shain et al., 2024). The second study causally verifies this by training LMs of different sizes on training examples that overlap minimally with the reading time corpora. Additionally, to examine how surprisal's fit to reading times would change in the face of severe leakage, these LMs are fine-tuned on examples created from the reading time corpora.

3.1 Methods

LM Training on Leakage-Free Data. We used the methodology of the previous study to identify training examples from the Pile that overlap minimally with the reading time corpora. Specifically, CDAWGs were built separately on 143 chunks of 1,000 training batches. A total of 18 chunks were found to have no more than 11 continuous tokens of overlap with any passage in the five reading time corpora, which filters out all overlaps improbable enough to meet our threshold in Figure 1a. Among these, we sampled 10 chunks (i.e. 10,000 training batches of 1,024 examples; ~20.9B tokens) as the training data. One epoch of this 'leakage-free' data was used to train Pythia-like Transformer LMs of three different sizes (Table 1) using the GPT-NeoX library (Andonian et al., 2021).[5]

LM Fine-Tuning on Reading Time Data. After the LMs were trained, leakage was artificially introduced by fine-tuning them on examples created from the reading time corpora. The construction procedure of the fine-tuning examples closely followed that of the Pythia training data. First, the passages of the five reading time corpora were shuffled and concatenated with <|endoftext|> tokens inserted at passage boundaries to create one long sequence consisting of 165,643 tokens. Subsequently, this sequence was split into contiguous sequences of length 2,048 to create one fine-tuning batch of 80 examples. This procedure was repeated to generate additional batches, each containing the five reading time corpora, albeit in different order. These batches were used to fine-tune each LM using the AdamW optimizer (Loshchilov and Hutter, 2019) with a constant learning rate of 0.0001. Results are reported after 5 and 10 fine-tuning steps.[6]

Surprisal Calculation and Reading Time Modeling. The three LMs were used to calculate word-by-word surprisal on the five reading time corpora, after both initial training and subsequent fine-tuning. When a passage did not fit into a context window of 2,048 tokens, the second half of the previous context window was used to condition the surprisal of the remaining tokens. As the Pythia LM's subword tokens contain leading whitespaces, word probabilities were calculated with trailing whitespaces to ensure their consistency (Oh and Schuler, 2024; Pimentel and Meister, 2024).[7]

Subsequently, for each LM, linear mixed-effects (LME; Bates et al., 2015) models that contain LM surprisal and standard baseline predictors were fit to about 50% of the data points in each reading time dataset (fit partition; Table 2).[8] The goodness-of-fit of each regression model was then evaluated by calculating the log-likelihood on about 25% of held-out data points (exploratory partition). This was compared against the log-likelihood of the baseline regression model that does not contain LM surprisal to evaluate the contribution of LM surprisal (ΔLogLik). All LME models incorporate maximal

[5]See Appendix A for additional training details.

[6]We expect each fine-tuning step to serve as an upper bound for the effect of data leakage due to observing the same data during pre-training, given the recent and repeated nature of exposure during fine-tuning.

[7]For example, without this correction, if both P(␣car | I ␣sold ␣the) and P(pet | I ␣sold ␣the ␣car) are very high, the combined probabilities of "␣car" and "␣car pet" given the context "I ␣sold ␣the" can exceed one.

[8]Self-paced reading times for Brown and Natural Stories, and first-pass durations for Dundee, GECO, and Provo. See Appendix B for the full LME modeling details.

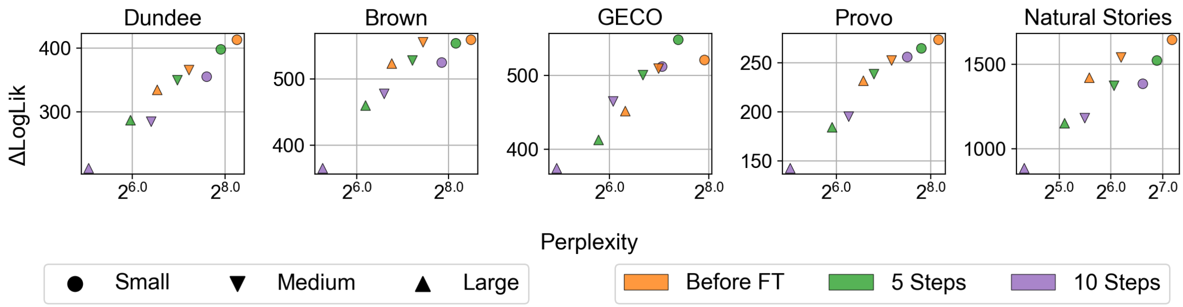


Figure 2: ΔLogLik due to surprisal on held-out data and corpus-level perplexity from LMs trained leakage-free (orange) and after 5 and 10 fine-tuning (FT) steps on reading time data (green and purple respectively).

by-subject random effects (Barr et al., 2013) and assume a linear relationship between surprisal and reading times (Wilcox et al., 2023b; Xu et al., 2023; Shain et al., 2024) and a lingering influence of surprisal from the previous word (Rayner et al., 1983). The LMs' perplexity on each reading time corpus is also reported.

3.2 Results

Figure 2 shows that LMs trained on leakage-free data filtered according to this very strict criterion still demonstrate a negative relationship between model size and fit to reading times on all five datasets. Together with the results of the previous study, this indicates that similar previous findings using Pythia and GPT-2 LMs (Oh and Schuler, 2023a; Shain et al., 2024) are not simply due to leakage. The results from the LMs fine-tuned on reading time data in Figure 2 also show that if severe leakage were to exist, this would result in an overestimation of the strength of this negative relationship. When examples from the reading time corpora are added to training, larger models seem to be able to predict certain words more accurately given the same number of fine-tuning updates, resulting in larger decreases in both perplexity and ΔLogLik. This suggests that smaller LMs are generally less susceptible to the influence of leakage, and that model-centered methods for diagnosing memorization (e.g. evaluating an LM's generated text given the prefix; Carlini et al., 2023) may be effective for assessing leakage in very large LMs.

4 Conclusion

This study examines whether commonly used naturalistic reading time corpora have leaked into large-scale datasets on which LMs are trained. In terms of sequence overlap, the leakage of most naturalistic reading time passages is found to be benign in two pre-training corpora. While setting a criterion for what constitutes severe data leakage is difficult and requires some judgment, we hope that the overlapping sequences between commonly used reading time corpora and pre-training corpora identified in this work provide a resource for those that wish to be more careful with psycholinguistic modeling.

The subsequent regression experiment replicates the negative relationship between model size and surprisal's fit to reading times using LMs trained on leakage-free data. Taken together, these results suggest that previously reported findings using LMs trained on these corpora are not driven by the effects of data leakage. In contrast to Wilcox et al. (2023a), who analyzed the potential influence of data leakage using smaller LMs trained from scratch, these results provide more direct evidence that generalizes to trends observed from larger pre-trained LMs like Pythia and GPT-2.

Previous studies have shown that more accurate predictions of named entities and other low-frequency words primarily drive this inverse scaling trend of pre-trained LM surprisal (Oh and Schuler, 2023b; Oh et al., 2024). This study additionally suggests that the ability of larger models to predict such words more accurately is not due to 'memorizing' exact sequences, but rather due to more complex word-to-word associations learned during training.

Acknowledgments

We thank the ARR reviewers and the area chair for their helpful comments, and William Merrill for his help with CDAWG construction. This work was supported by the National Science Foundation (NSF) grant #1816891. All views expressed are those of the authors. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

Limitations

In this work, the potential leakage of naturalistic text stimuli is evaluated through studies using English corpora, language models trained on English text, and reading time data from native speakers of English. Therefore, replication studies are necessary to further assess the leakage of text stimuli in other languages. Additionally, data leakage in this work is diagnosed mainly through token n-gram overlaps, which is insensitive to minor variations in form. Moreover, as the OpenWebText Corpus is an open-source effort to replicate GPT-2's undisclosed training data, the corpus statistics of the actual training data may differ. Finally, this work is concerned with the use of language models as cognitive models of human sentence processing, and therefore does not relate to their use in natural language processing applications.

Ethics Statement

This work used publicly available text corpora (Gokaslan and Cohen, 2019; Gao et al., 2020) and human subject data collected as part of previously published research (Futrell et al., 2021; Smith and Levy, 2013; Cop et al., 2017; Kennedy et al., 2003; Luke and Christianson, 2018). Readers are referred to the respective publications for more information about the data collection and validation procedures. As this work studies the connection between language models and human sentence processing, its potential negative impacts on society appear to be minimal.

References

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Shivanshu Purohit, Tri Songz, Phil Wang, and Samuel Weinbach. 2021. GPT-NeoX: Large scale autoregressive language modeling in PyTorch.

Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68:255–278.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 2397–2430.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Agatha Christie. 1920. *The Mysterious Affair at Styles*. John Lane. Retrieved from Project Gutenberg.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior Research Methods*, 49(2):602–615.

Maxime Crochemore and Renaud Vérin. 1997. On compact directed acyclic word graphs. In Jan Mycielski, Grzegorz Rozenberg, and Arto Salomaa, editors, *Structures in Logic and Computer Science: A Selection of Essays in Honor of A. Ehrenfeucht*, pages 192–211. Springer Berlin Heidelberg.

Richard Futrell, Edward Gibson, Harry J. Tily, Idan Blank, Anastasia Vishnevetsky, Steven Piantadosi, and Evelina Fedorenko. 2021. The Natural Stories corpus: A reading-time corpus of English texts containing rare syntactic constructions. *Language Resources and Evaluation*, 55:63–77.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint*, arXiv:2101.00027.

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText Corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–8.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696.

Shunsuke Inenaga, Hiromasa Hoshino, Ayumi Shinohara, Masayuki Takeda, Setsuo Arikawa, Giancarlo Mauri, and Giulio Pavesi. 2005. On-line construction of compact directed acyclic word graphs. *Discrete Applied Mathematics*, 146(2):156–179.

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The Dundee Corpus. In *Proceedings of the 12th European Conference on Eye Movement*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Henry Kučera and W. Nelson Francis. 1967. *Computational analysis of present-day American English*. Brown University Press, Providence, RI.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7:195–212.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*.

Steven G. Luke and Kiel Christianson. 2018. The Provo Corpus: A large eye-tracking corpus with predictability norms. *Behavior Research Methods*, 50(2):826–833.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540.

William Merrill, Noah A. Smith, and Yanai Elazar. 2024. Evaluating n-gram novelty of language models using Rusty-DAWG. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14459–14473.

Byung-Doh Oh and William Schuler. 2023a. Transformer-based language model surprisal predicts human reading times best with about two billion training tokens. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1915–1921.

Byung-Doh Oh and William Schuler. 2023b. Why does surprisal from larger Transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Byung-Doh Oh and William Schuler. 2024. Leading whitespaces of language models' subword vocabulary pose a confound for calculating word probabilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3464–3472.

Byung-Doh Oh, Shisen Yue, and William Schuler. 2024. Frequency explains the inverse correlation of large language models' size, training data amount, and surprisal's fit to reading times. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2644–2663.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2009. English Gigaword LDC2009T13.

Tiago Pimentel and Clara Meister. 2024. How to compute the probability of a word. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18358–18375.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. ZeRO: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16.

Keith Rayner, Marcia Carlson, and Lyn Frazier. 1983. The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior*, 22(3):358–374.

Cory Shain, Clara Meister, Tiago Pimentel, Ryan Cotterell, and Roger Levy. 2024. Large-scale evidence for logarithmic effects of word predictability on reading time. *Proceedings of the National Academy of Sciences*, 121(10):e2307876121.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128:302–319.

Ethan Gotlieb Wilcox, Clara Meister, Ryan Cotterell, and Tiago Pimentel. 2023a. Language model quality correlates with psychometric predictive power in multiple languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7503–7511.

Ethan Gotlieb Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023b. Testing the predictions of surprisal theory in 11 languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

Weijie Xu, Jason Chon, Tianran Liu, and Richard Futrell. 2023. The linearity of the effect of surprisal on reading times across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15711–15721.

A LM Training Details

The LMs in Study 2 were trained closely following the training procedures of the Pythia LMs.

Datasets	LME Formula
Brown Natural Stories	`RT ~ LMsurp + LMsurp_prev + Unisurp + length + index + (LMsurp + LMsurp_prev + length + index + 1 | subject)`
Dundee GECO Provo	`RT ~ LMsurp + LMsurp_prev + Unisurp + length + index + pfix + (LMsurp + index + 1 | subject)`

Table 3: Formulae of LME models fit in Study 2. `LMsurp`: LM surprisal, `LMsurp_prev`: LM surprisal of previous word, `Unisurp`: unigram surprisal, `length`: word length, `index`: position of the word within the sentence, `pfix`: whether the previous word was fixated. The baseline regression models were fit with these formulae without the `LMsurp` and `LMsurp_prev` predictors. All predictors were z-transformed.

Like many other Transformer LMs, Pythia LMs are based on decoder-only layers, but they parallelize the computations of the attention mechanism and those of the feedforward neural network, and do not use tied parameters for the input embedding and final projection matrices. The three LMs were trained using the Zero Redundancy Optimizer (ZeRO; Rajbhandari et al., 2020) implementation of Adam (Kingma and Ba, 2015) with a maximum learning rate of 0.001. This learning rate was warmed up linearly over the first 1% of training steps (i.e. 100 steps) and was subsequently lowered to a minimum of 0.0001 following a cosine annealing schedule over the remainder of the 10,000 training steps. Gradients were clipped to a maximum norm of 1 prior to each update to stabilize training. All training took place in half-precision on 48GB Nvidia RTX 8000 GPUs.

B LME Modeling Details

Data Preprocessing and Partitioning. For the Brown and Natural Stories datasets, reading times of words at sentence boundaries and those shorter than 100 ms or longer than 3,000 ms were excluded. Data from subjects who answered four or fewer comprehension questions correctly were also removed from the Natural Stories data. The Dundee, GECO, and Provo datasets were filtered to exclude reading times of unfixated words, words following saccades longer than four words, and words at sentence and document boundaries. Reading times of words at line and screen boundaries were also removed from the Dundee data that provides annotations of line/screen locations.

After data preprocessing, each dataset was partitioned into fit and exploratory partitions that comprise roughly of 50% and 25% of the data respectively (Table 2). This partitioning was based on the sum of the subject ID and the sentence ID, which keeps all data from a particular subject-sentence combination intact in one partition. Each fit partition was used to fit the regression models, and the exploratory partition was used to calculate regression model likelihood. The remaining ~25% of the data is reserved for statistical significance testing and was not used in this work.

LME Model Specifications. The baseline predictors included in all LME models are word length in characters, index of word position within the sentence, unigram surprisal (all datasets), and whether the previous word was fixated (Dundee, GECO, Provo only). Unigram surprisal was calculated using the KenLM toolkit (Heafield et al., 2013) with probabilities estimated on the OpenWebText Corpus (Gokaslan and Cohen, 2019). The by-subject random effects structures of the LME models were determined by starting with maximal random effects and removing the least predictive random effect until all LME models converged. The resulting LME formulae are outlined in Table 3.