



Unveiling the language processing of humans (and machines)

Byung-Doh Oh

April 25, 2025

Some material from Paula Buttery, Matthew Crocker, Roger Levy

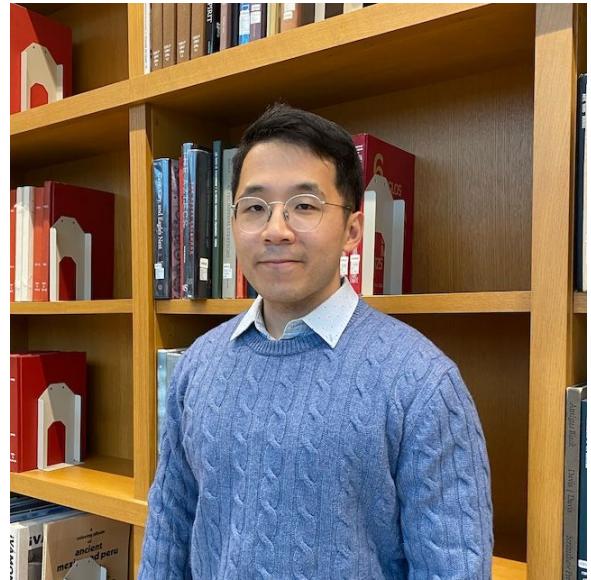
DS-GA 1012
NLU/CompSem

About me

I am a Faculty Fellow (\approx postdoc) in Data Science

- Research: You will hear about today
- Teaching: DS-GA 1015, Text as Data

I have email: oh.b@nyu.edu



PART 01

What is (computational) psycholinguistics?

I landed in JFK and took a ...

You (and your brain) were probably able to:

- Build some **mental representation** without seeing the end of sentence
- Do so **incrementally** without much conscious effort

Psycholinguistics

- Psycholinguistics is concerned with how we comprehend and produce language
- Psycholinguistics is concerned with how language is represented and processed in the mind

Propagation across levels

Acoustic Signal

Word Segmentation

Lexical Access

Syntactic Parsing

Semantic Interpretation

Meaning

Input over time

Example research questions

- Speech perception: How do infants segment acoustic input into words?
- Morphology: How are different inflected forms represented in the brain?
- Semantics: How is meaning represented in the brain?

The subfield I'm into (sentence processing)

Why are some words or sentences more difficult to process than others?

- The cat the dog licked ran away
- The cat the dog the rat chased licked ran away
- The fact that the employee who the manager hired stole office supplies worried the executive
- The executive who the fact that the employee stole office supplies worried hired the manager

What kind of data is collected and analyzed?

Usually some form of response to stimuli sentences in an experiment

- Measurement of **reaction time** to some linguistic task
- Measurement of **reading times**
- Measurement of **brain response**

Lexical-Decision Task

DOWT

When word

Z

Is the word shown on screen a
real word or is it made up?

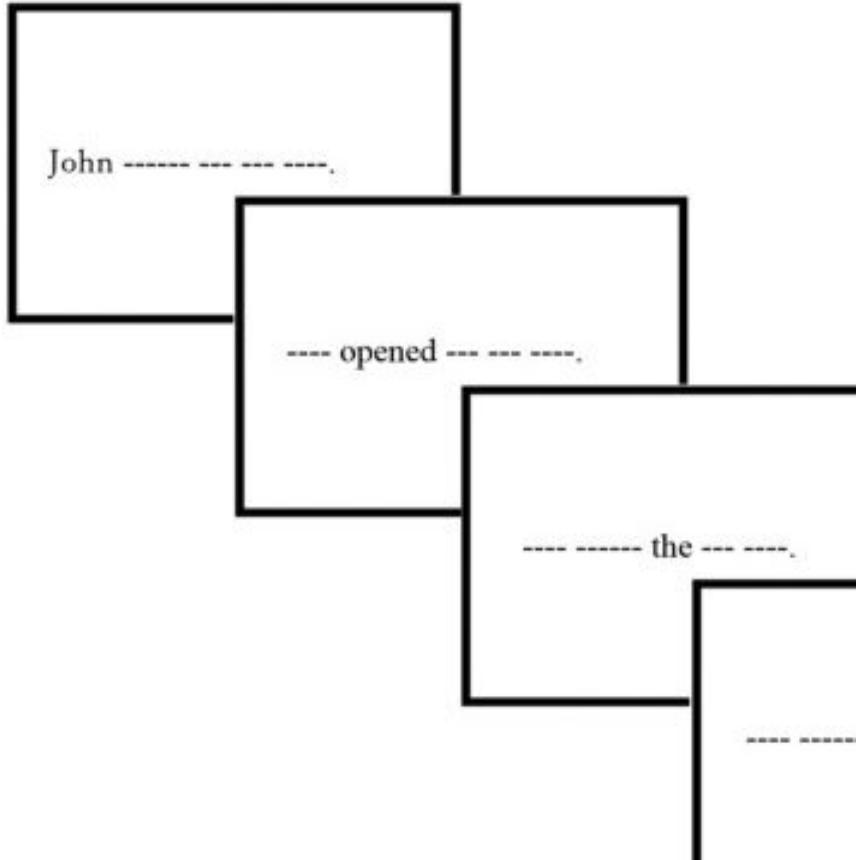
When Non-word

M

Lexical decision task:

Decide whether a word is
real or made up as
quickly as you can

The time taken to
respond (reaction time)
is measured



Self-paced reading (SPR):

Press a key on the keyboard to reveal the next word

Word-by-word reading times are measured as the time taken between keystrokes

Cannot return to earlier words

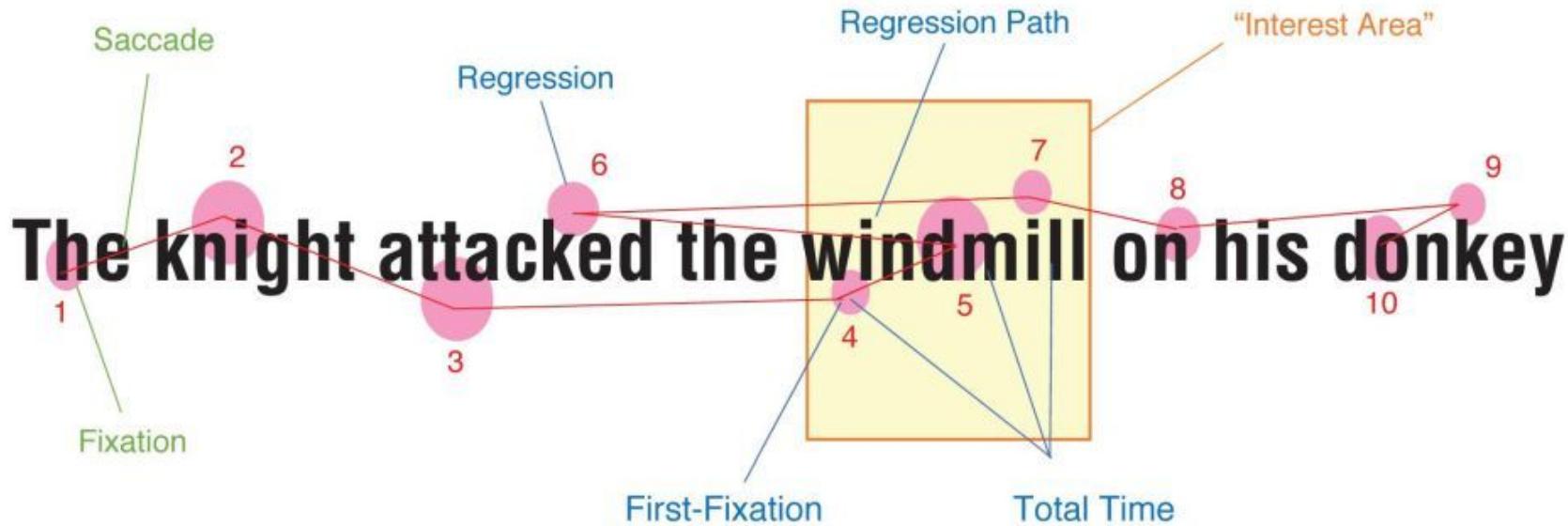


Eye-tracking (ET):

Wear an eye-tracker and read some text as you would naturally

Can return to earlier words

The eye movement data has to be post-processed to derive word-by-word reading times



- Some words are skipped during reading
- Eye-tracking allows re-reading (**regressions**; 6 in example)
- **First-pass duration:** Time taken between entering a word region for the first time from the left and leaving it to either left or right (4+5 in example)
- **Go-past duration:** Time taken between entering a word region for the first time from the left and leaving it to the right (4+5+6+7 in example)

Modality	Signal type	Temporal resolution	Spatial resolution	Method type	Portability
EEG	Electrical 	Approx. 0.05s	Approx. 10mm	Non-invasive 	Portable 
MEG	Magnetic 	Approx. 0.05s	Approx. 5mm	Non-invasive 	Non-portable 
ECoG	Electrical 	Approx. 0.03s	Approx. 1mm	Invasive 	Portable 
ICNR	Electrical 	Approx. 0.03s	Approx. 0.5mm(LFP)	Invasive 	Portable 
fMRI	Metabolic 	1s	Approx. 1mm	Non-invasive 	Non-portable 
NIRS	Metabolic 	1s	Approx. 5mm	Non-invasive 	Portable 

Brain responses:

Usually with a listening task,
difficult to collect

EEG, MEG, fMRI most common in
psycholinguistic studies

Complementary advantage in
temporal and spatial resolution

What kind of data is collected and analyzed?

The core assumptions of data analysis are:

- The time taken to react to a task or read the word reflects **processing difficulty** (difficult words take longer to read)
- Similarly, the change in brain measures also reflects processing difficulty (difficult words cause spike/dip in electrical activity, increase in blood flow)

So, what is computational psycholinguistics?

Computational psycholinguistics aims to develop **computational models** of the **cognitive mechanism underlying language comprehension**

The models should ideally:

- Generate concrete predictions for the phenomena of interest
- Accurately capture the trend in the experimental data
- Embody a **linking hypothesis** between the underlying mechanism and the observed behavior

So, what is computational psycholinguistics?

Computational psycholinguistics aims to develop **computational models** of the cognitive mechanism underlying language comprehension

The remainder of this lecture will be about evaluating Transformer language models (LMs) as computational models of predictive processing

- 1) The theory: How can LMs be viewed as a model of language processing?
- 2) The results: In what aspects are Transformer LMs ‘superhuman?’

PART 02

The theory: LMs as models of predictive processing

The shared principle of prediction (humans)

Prediction based on diverse contextual cues affect human language processing

Syntactic

- *Jamie was clearly intimidated...*

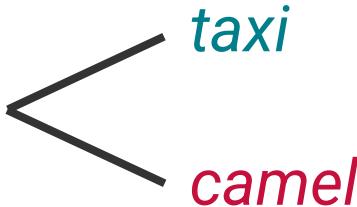
Phonological

- *Terry ate an...*
- *Terry ate a...*

Semantic & world knowledge

- *The children went outside to...*

The shared principle of prediction (humans)

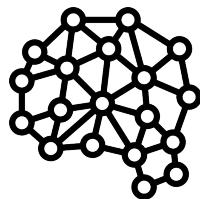
I landed in JFK and took a  *taxi*
camel

The more predictable *taxi* is easier to process than *camel*

Decades of psycholinguistics research have confirmed this

- Predictable words show distinct EEG profiles (Kutas & Hillyard, 1980)
- Predictable words are skipped more in eye-tracking data (Ehrlich & Rayner, 1981)

The shared principle of prediction (LMs)



I landed in JFK and took a

w_t	$P(w_t w_{1..t-1})$
<i>taxi</i>	0.1174
<i>flight</i>	0.0635
...	...
<i>camel</i>	3.8×10^{-5}

LMs learn nontrivial linguistic structure by simply predicting the next word
(Futrell & Mahowald, 2025; Linzen & Baroni, 2021; Mahowald et al., 2024)

The shared principle of prediction (LMs)

NLP models used to be limited in scope

Due to recent advances, a lot of NLP in 2025 is pushing $P(w_t | w_{1..t-1})$ to its limit

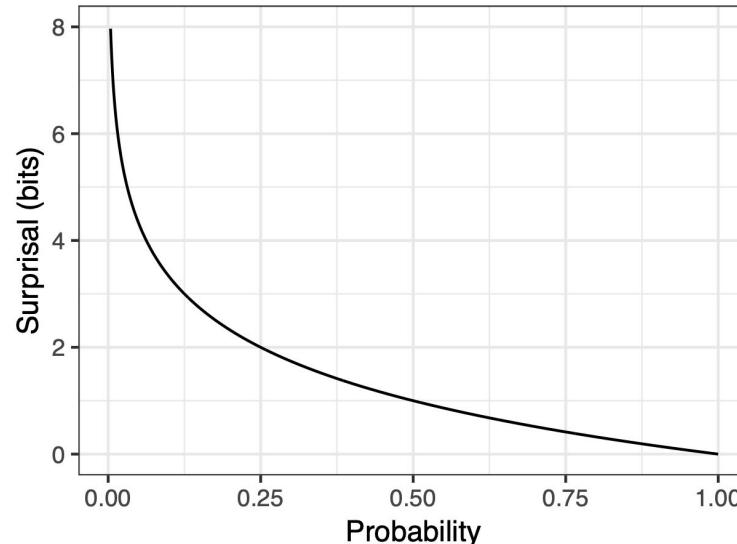
- Vast increase in model/data scale
- In-context learning/X-of-thought prompting
- Reinforcement learning from human feedback

Although with different goals, there is room to be creative with $P(w_t | w_{1..t-1})$ for computational psycholinguistics as well

Theoretical link between humans and LMs

Surprisal Theory (Hale, 2001; Levy, 2008): A word's difficulty is its **surprisal in context**

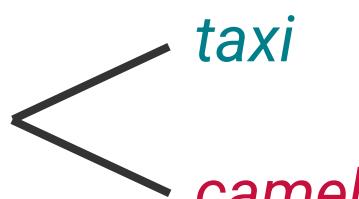
$$S(w_t) := -\log_2 P(w_t | \text{Context}) \approx -\log_2 P(w_t | w_{1\dots t-1})$$



Theoretical link between humans and LMs

Surprisal Theory (Hale, 2001; Levy, 2008): A word's difficulty is its **surprisal in context**

$$\text{Reading time of } w_t \propto -\log_2 P(w_t | w_{1..t-1})$$

I landed in JFK and took a 
taxi
camel

Reading time of *taxi* $\propto -\log_2 P(\text{i} \mid \text{I landed in JFK and took a})$

Reading time of *camel* $\propto -\log_2 P(\text{m} \mid \text{I landed in JFK and took a})$

Modeling methodology

w_t	<i>I</i>	<i>landed</i>	<i>in</i>	<i>JFK</i>
Reading time	709 ms	847 ms	766 ms	886 ms
$S_{LM1}(w_t)$	4.95	6.40	1.32	6.04
$S_{LM2}(w_t)$	3.53	5.73	0.69	4.14
$S_{LM3}(w_t)$	3.50	5.13	0.59	3.63

- Regression modeling conducted to fit surprisal (predictor) to RT (response)
- We can then evaluate which $S(w_t)$ fits reading times best

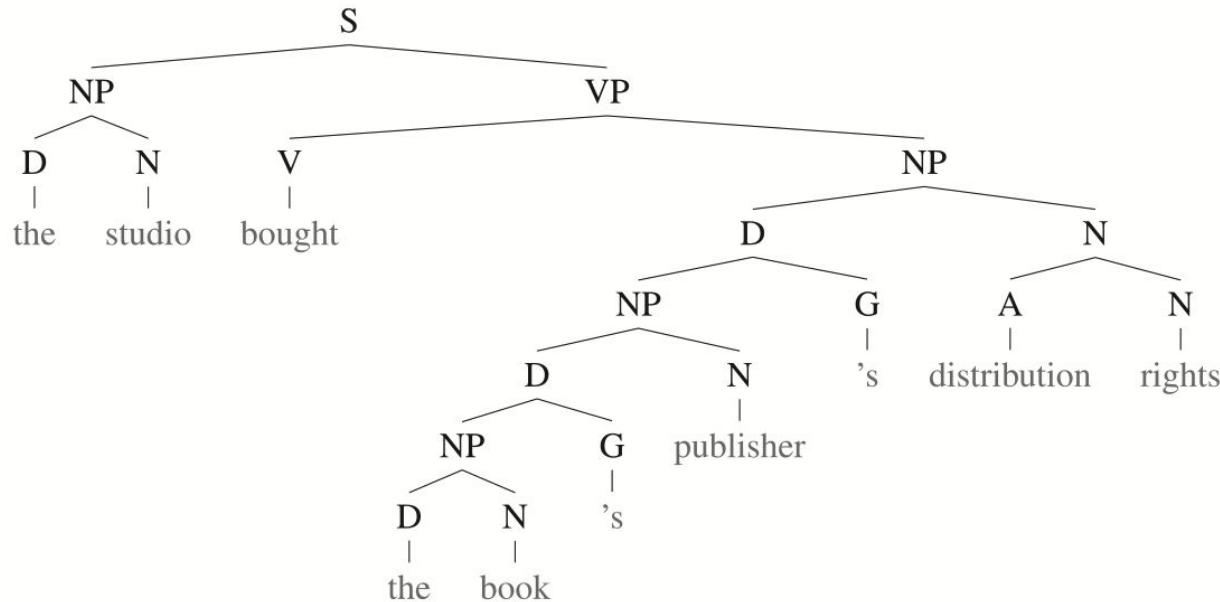
Some historical lore

A lot of sentence processing research has to do with syntactic structure

As such, **generative, incremental** syntactic parsers (joint distribution over words and parses) have been used to study the influence of **syntactic prediction**

- Marginalize over parses to derive surprisal
- If there's a big change in the likely parse as a result of observing the next word, its surprisal goes up

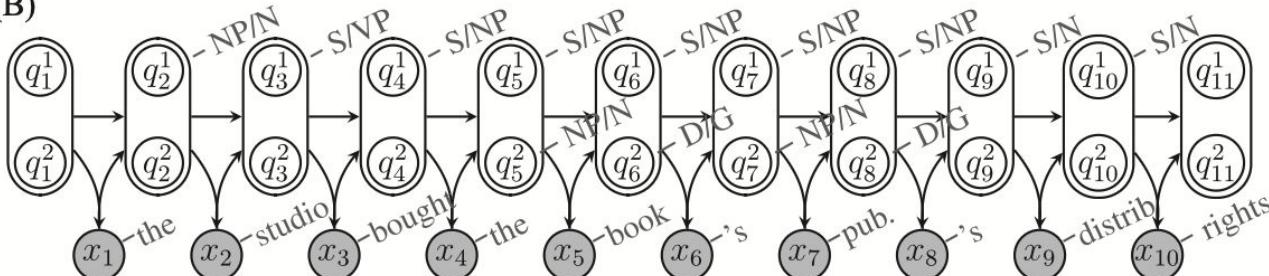
(A)



Example from [van Schijndel et al. \(2013\)](#)

What is shown in (B) is called left-corner parsing

(B)

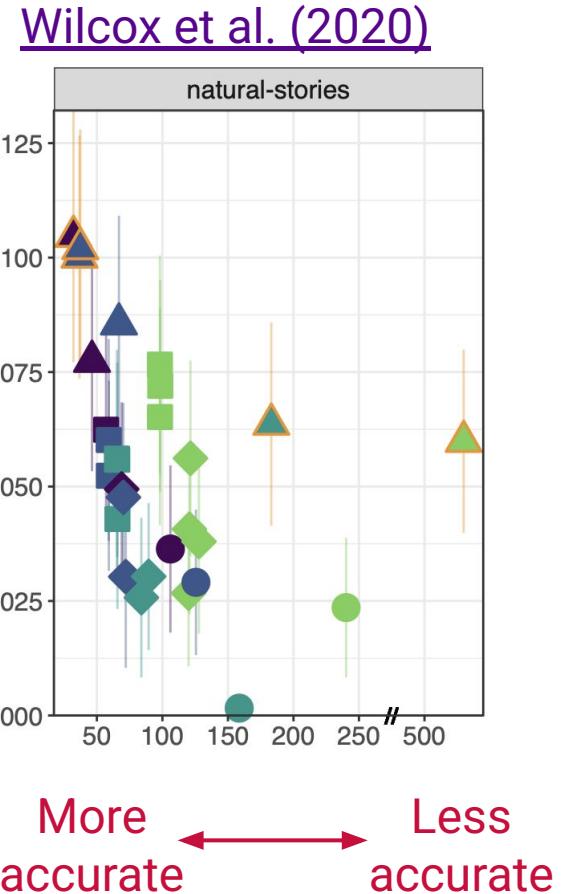
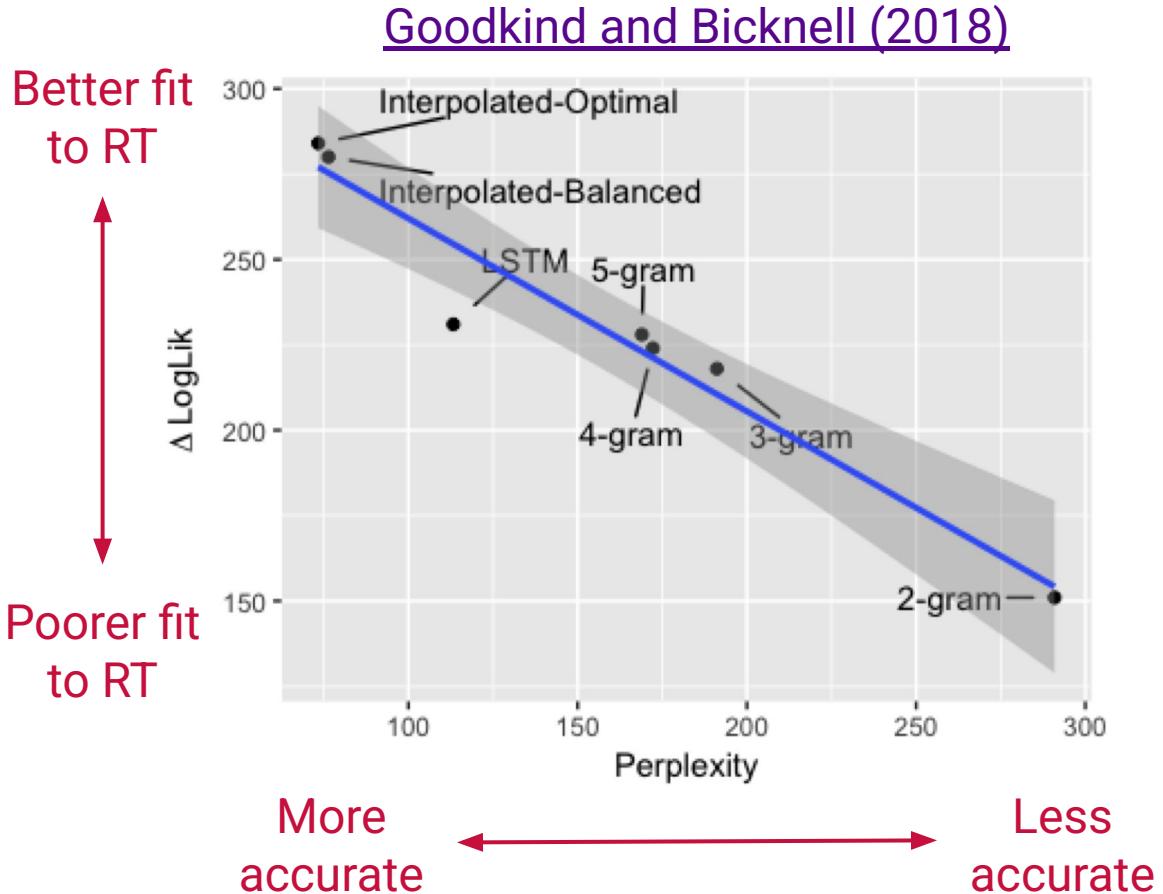


Maintain e.g. 2k parses on the beam, marginalize over them at each timestep to calculate surprisal

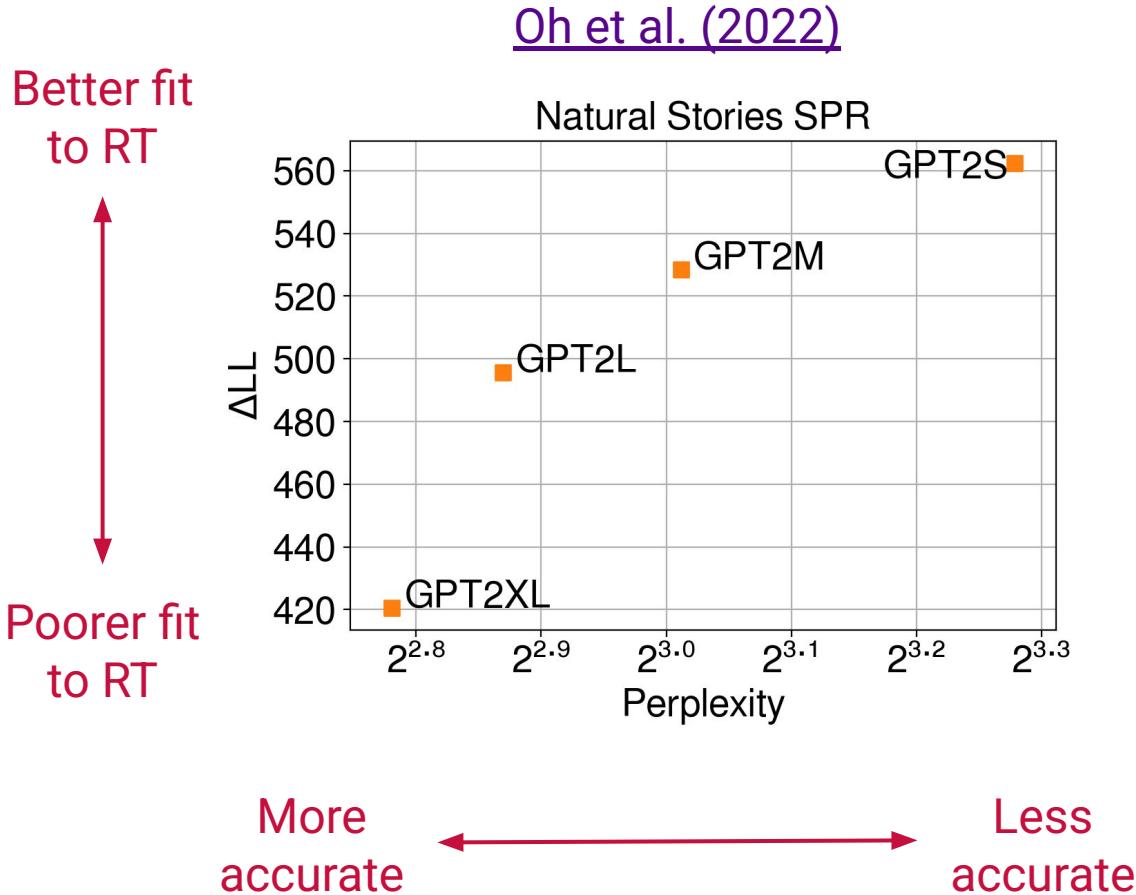
PART 02

The results: Where LMs show ‘superhuman’ language processing

People thought that language processing is driven by accurate prediction



This relationship completely breaks down with more contemporary* LMs



1. Does this trend replicate with other Transformer LMs?
2. If so, are there linguistic factors that drive this trend?

Why does surprisal from larger Transformer-based language models provide a poorer fit to human reading times?

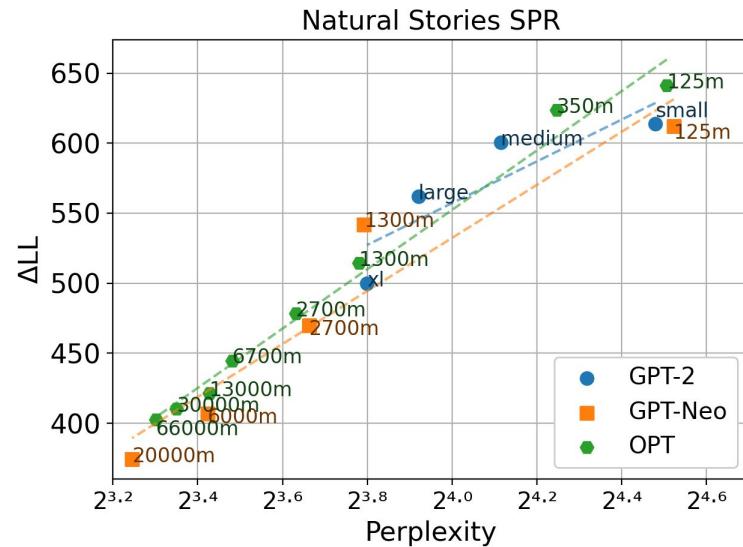
Methods: Replication with more LMs

- Regression models fit to **Natural Stories** and **Dundee** datasets
- Baseline predictors: word length/position, **saccade length**, **previous word fixated**
- Predictors of interest: LM surprisal
- Evaluation: Δ log-likelihood (Δ LL)

Model	Model size (#Parameters)
GPT-2 Small	~117M
GPT-2 Medium	~345M
GPT-2 Large	~774M
GPT-2 XL	~1.6B
GPT-Neo 125M	~125M
GPT-Neo 1.3B	~1.3B
GPT-Neo 2.7B	~2.7B
GPT-J 6B	~6B
GPT-NeoX 20B	~20B
OPT 125M	~125M
OPT 350M	~350M
OPT 1.3B	~1.3B
OPT 2.7B	~2.7B
OPT 6.7B	~6.7B
OPT 13B	~13B
OPT 30B	~30B
OPT 66B	~66B

Larger LMs provide poorer predictors of reading times

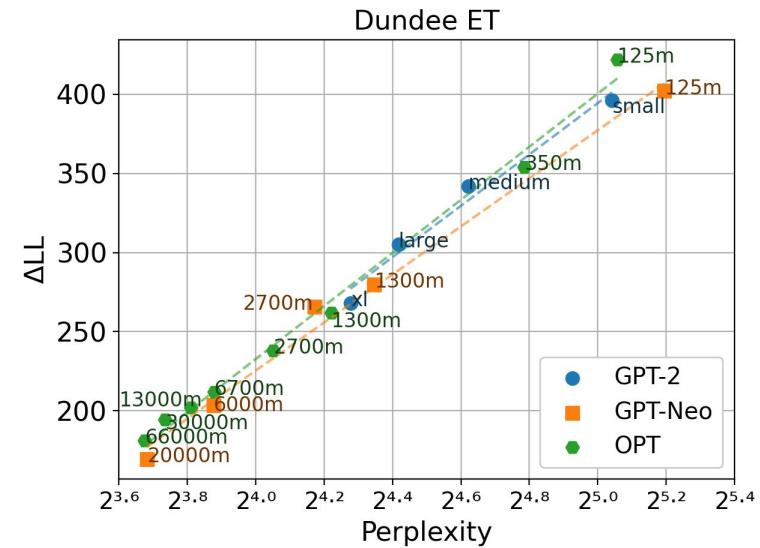
Better fit
to RT



Poorer fit
to RT

More accurate,
larger

Less accurate,
smaller



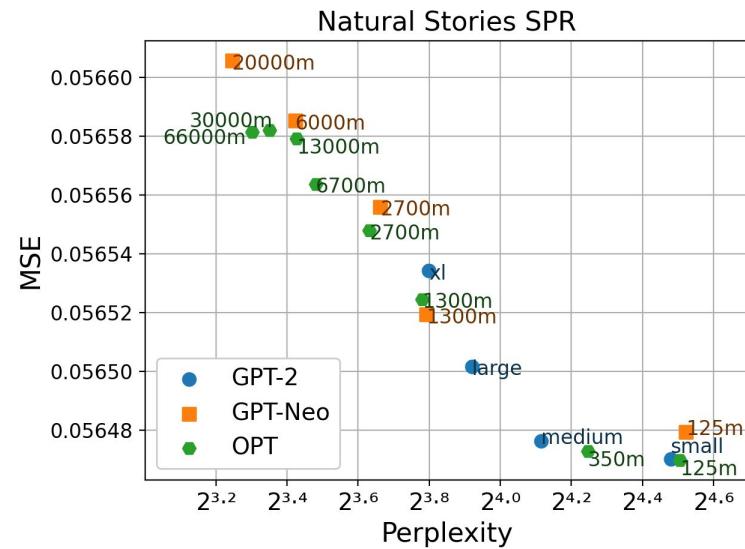
More accurate,
larger

Less accurate,
smaller

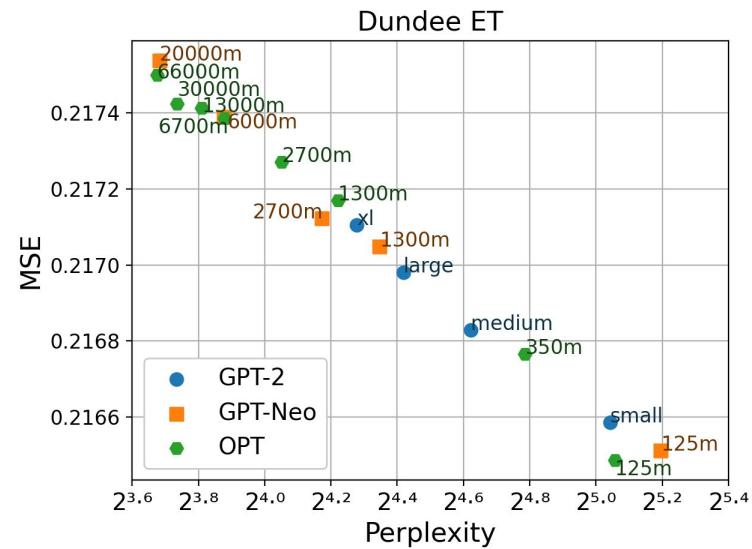
What linguistic factors drive this trend?

- Text annotated with word-level and syntactic properties
- Top 5 subsets with the largest difference in MSE between models identified

Poorer fit
to RT

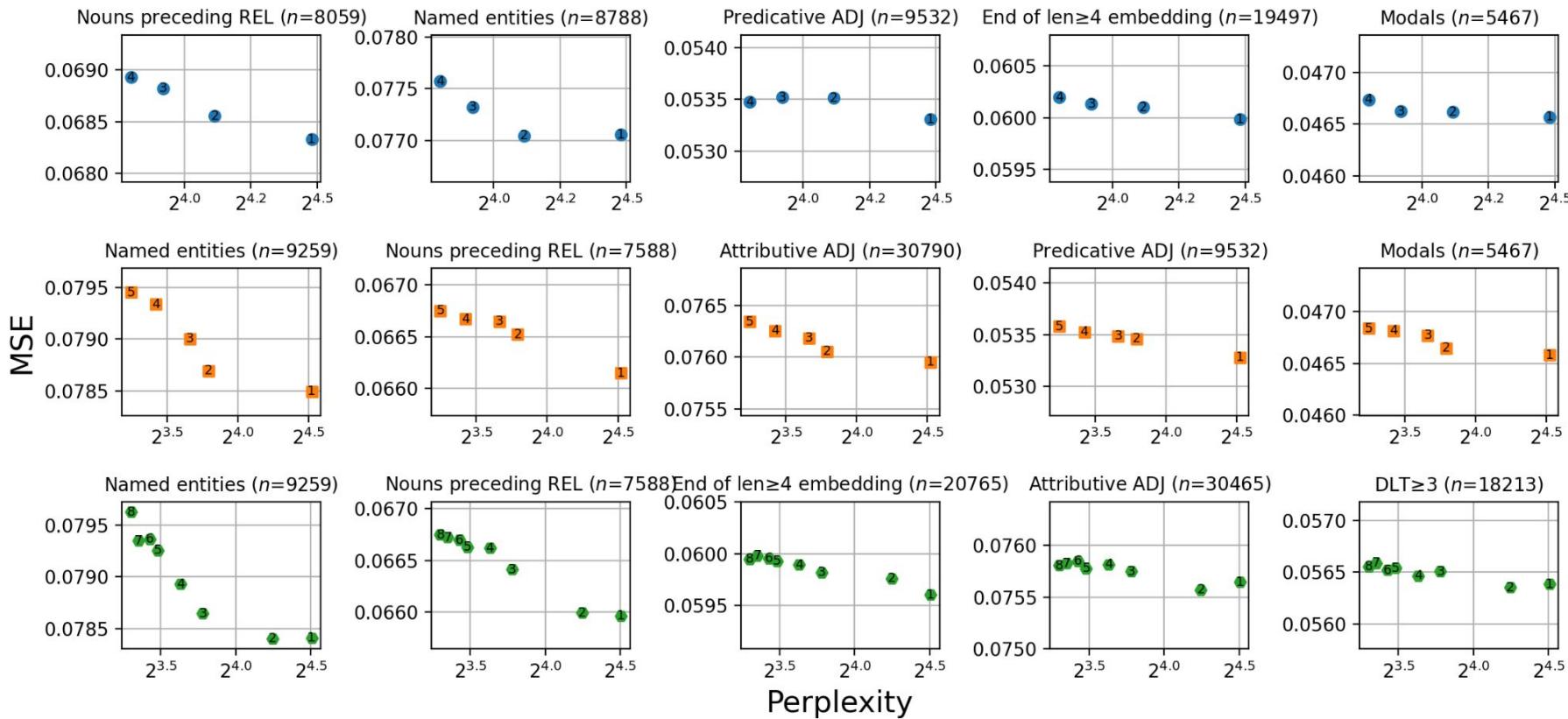


Better fit
to RT



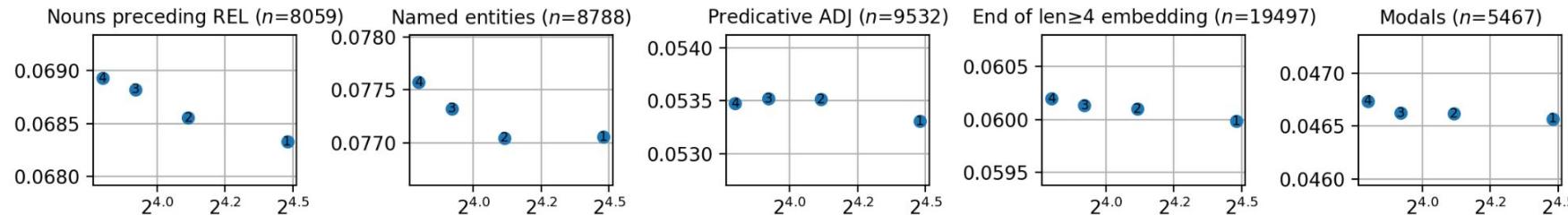
The effect of LM size seems to be driven by open-class words like named entities

Natural Stories SPR



The effect of LM size seems to be driven by open-class words like named entities

Natural Stories SPR

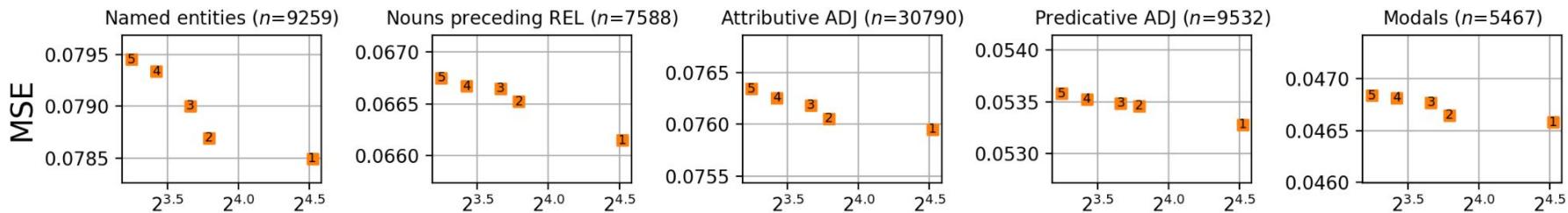


MSE

Perplexity

The effect of LM size seems to be driven by open-class words like named entities

Natural Stories SPR

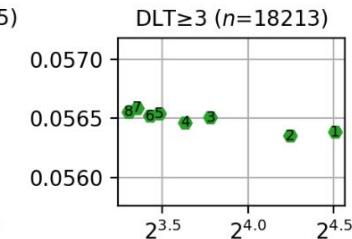
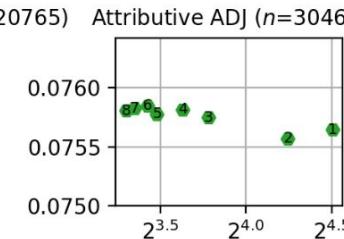
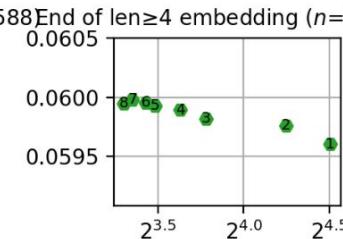
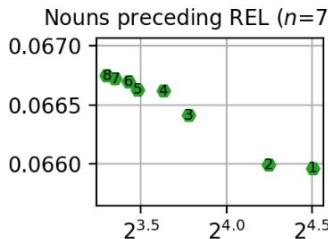
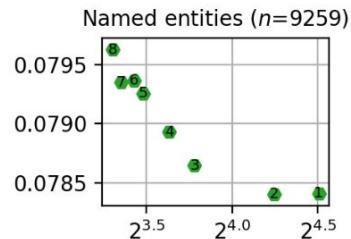


Perplexity

The effect of LM size seems to be driven by open-class words like named entities

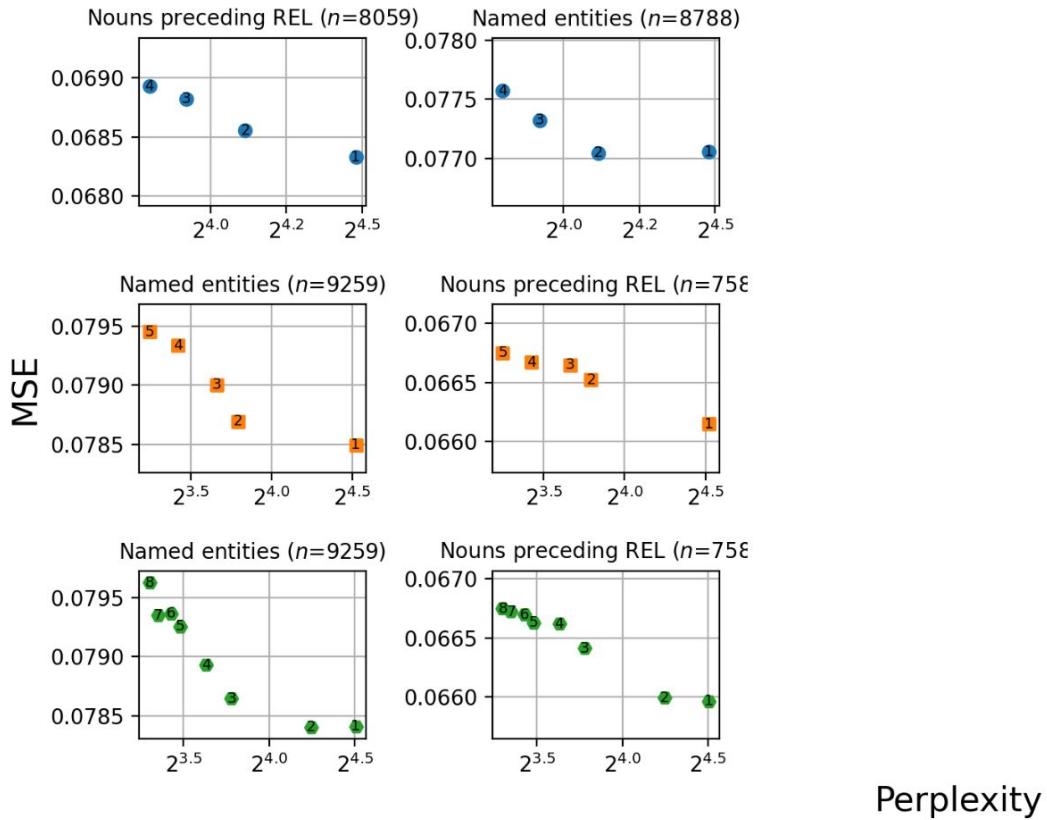
Natural Stories SPR

MSE



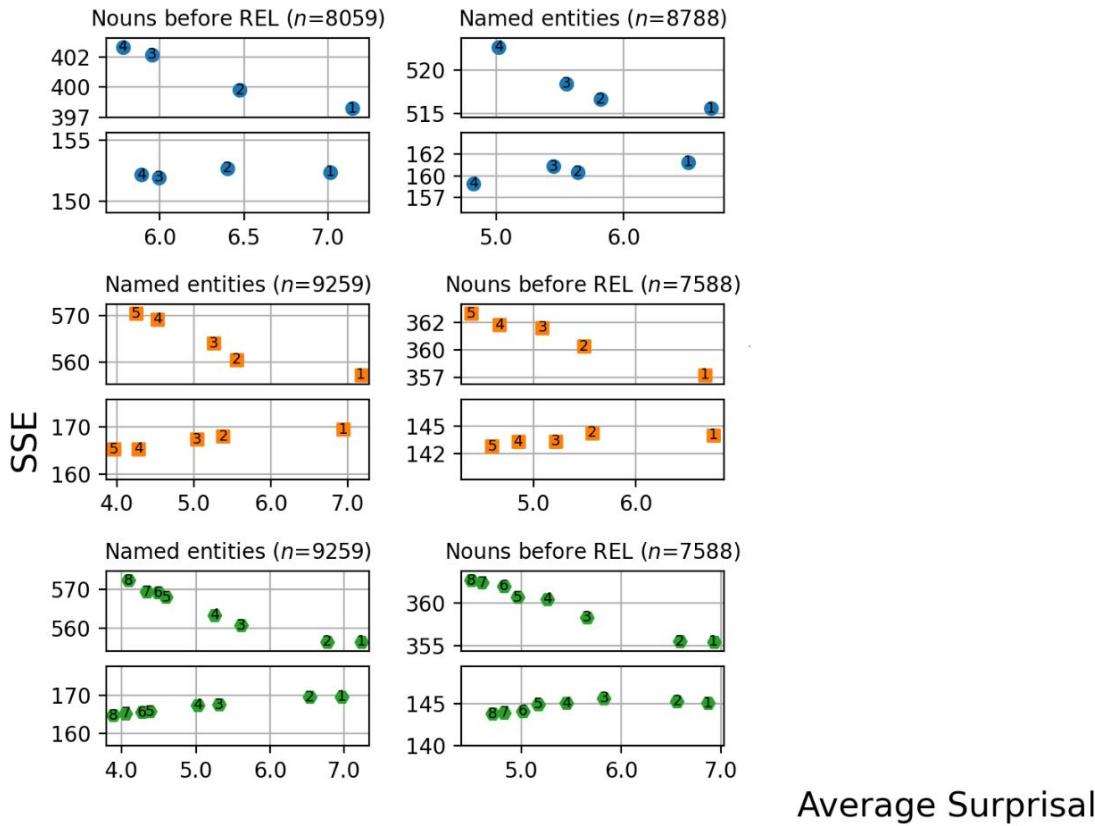
The effect of LM size seems to be driven by open-class words like named entities

Natural Stories SPR



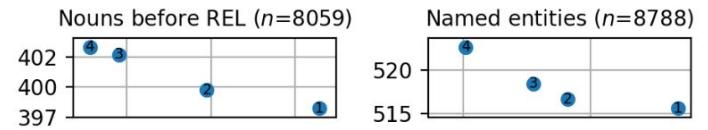
Larger LMs underpredict reading times of named entity terms

Natural Stories SPR

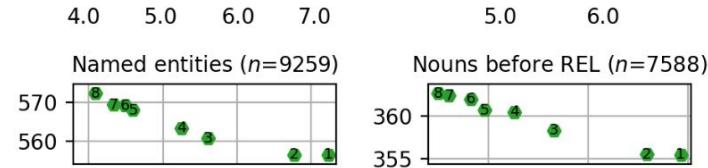
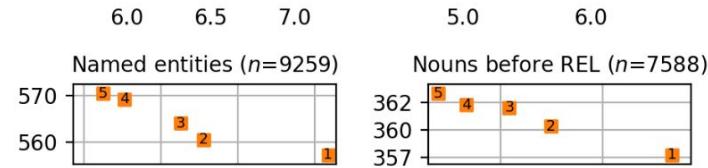


Larger LMs underpredict reading times of named entity terms

Natural Stories SPR



SSE



Average Surprisal

Some examples

Large LMs can predict the following words with very high probability:

(In a passage about the Roswell UFO incident)

... In January nineteen ninety-seven, Karl _____, one of the more prominent pro-UFO researchers, ...

(In a passage about the Tulip mania)

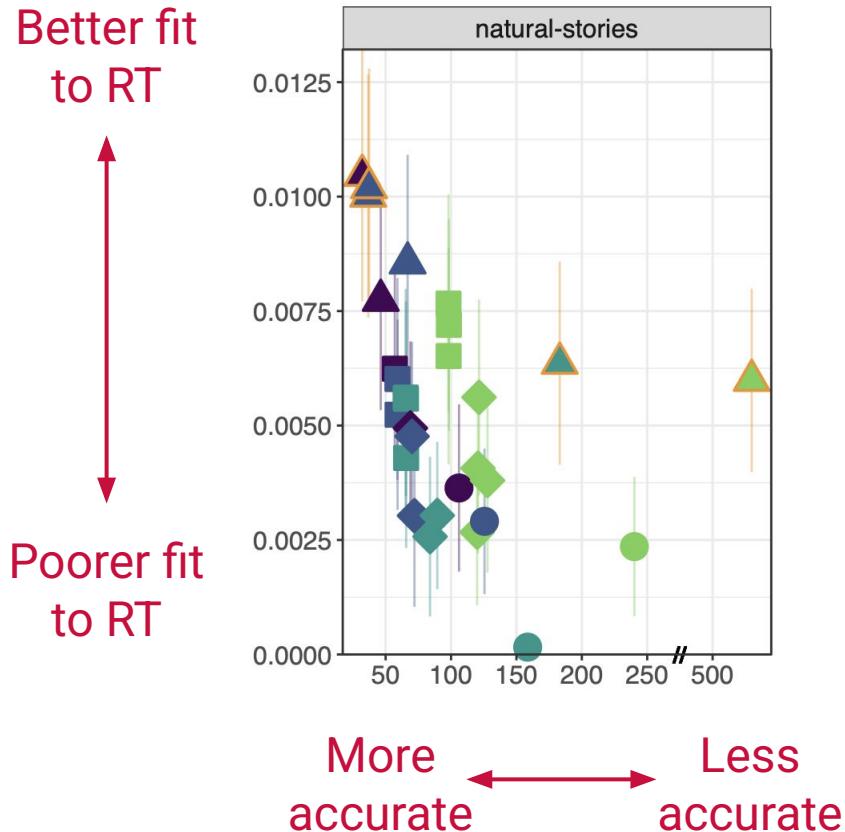
... At one point twelve acres of land were offered for a Semper _____ bulb. ...

Recap (1)

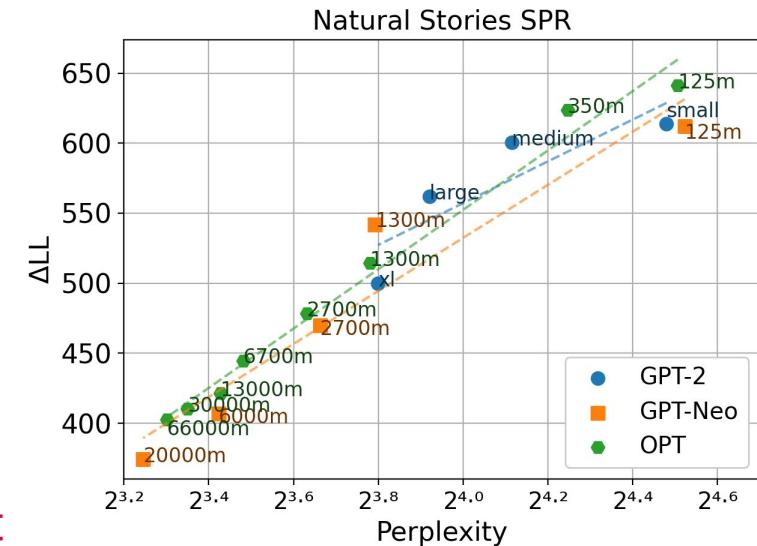
- Surprisal from larger LMs show strictly poorer fits to human reading times
- Effect mostly driven by underpredictions of reading times by LM surprisal
- In NLP terms, LMs seem to hold much more “parametric knowledge” compared to an average reader
- This is desirable for NLP applications, but does make them ‘superhuman’

Still conflicting results about next-word prediction accuracy

Wilcox et al. (2020)



Oh and Schuler (2023)



More accurate, larger Less accurate, smaller

The difference in models across studies

Big difference in terms of both LM size and training data amount

- Model size: probably small vs. 66B parameters
- Training data: 42M tokens vs. 8.7B tokens

The relationship probably reverses somewhere in between, but there is a really big middle ground to cover

☞ Pythia: Interpreting Transformers Across Time and Scale

This repository is for EleutherAI's project *Pythia* which combines interpretability analysis and scaling laws to understand how knowledge develops and evolves during training in autoregressive transformers. For detailed info on the models, their training, and their properties, please see our paper [Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling](#).

The Pythia suite was developed with the explicit purpose of enabling research in interpretability, learning dynamics, and ethics and transparency for which existing model suites were inadequate. The key features of the Pythia suite are:

1. All models, data, and code used in the paper are publicly released, enabling full reproducibility of results. All results in our paper have been independently verified by at least one other lab.
2. All models feature 154 checkpoints saved throughout training, enabling the study of learning dynamics of LLMs.
3. All models were trained on the same data in the same order, enabling researchers to explore causal interventions on the training process.

Transformer-based language model surprisal predicts human reading times best with about two billion training tokens

Covering the middle ground (training data)

- Regression models fit to Natural Stories and Dundee datasets, ΔLL calculated
- Predictors of interest: LM surprisal
- Trained on identical batches of 1024×2048 (~2M) tokens
- Intermediate checkpoints evaluated

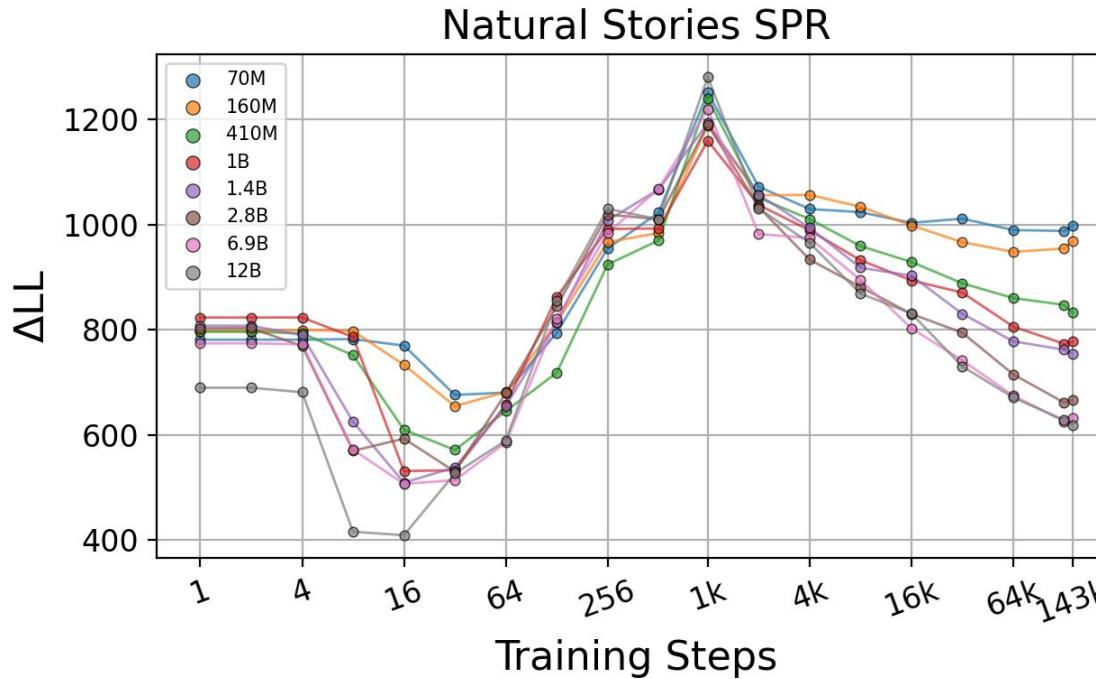
Model	Model size (#Parameters)
Pythia 70M	~70M
Pythia 160M	~160M
Pythia 410M	~410M
Pythia 1B	~1B
Pythia 1.4B	~1.4B
Pythia 2.8B	~2.8B
Pythia 6.9B	~6.9B
Pythia 12B	~12B

Sweet spot at around two billion tokens

Better fit
to RT



Poorer fit
to RT



Less
data



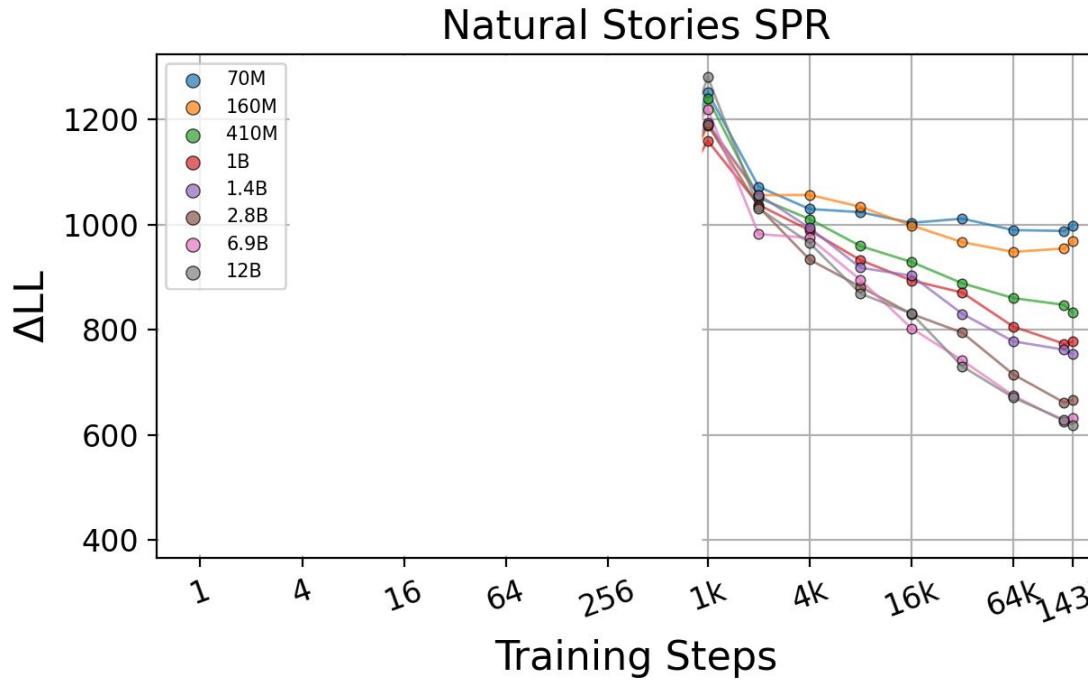
More
data

Sweet spot at around two billion tokens

Better fit
to RT



Poorer fit
to RT



Less
data



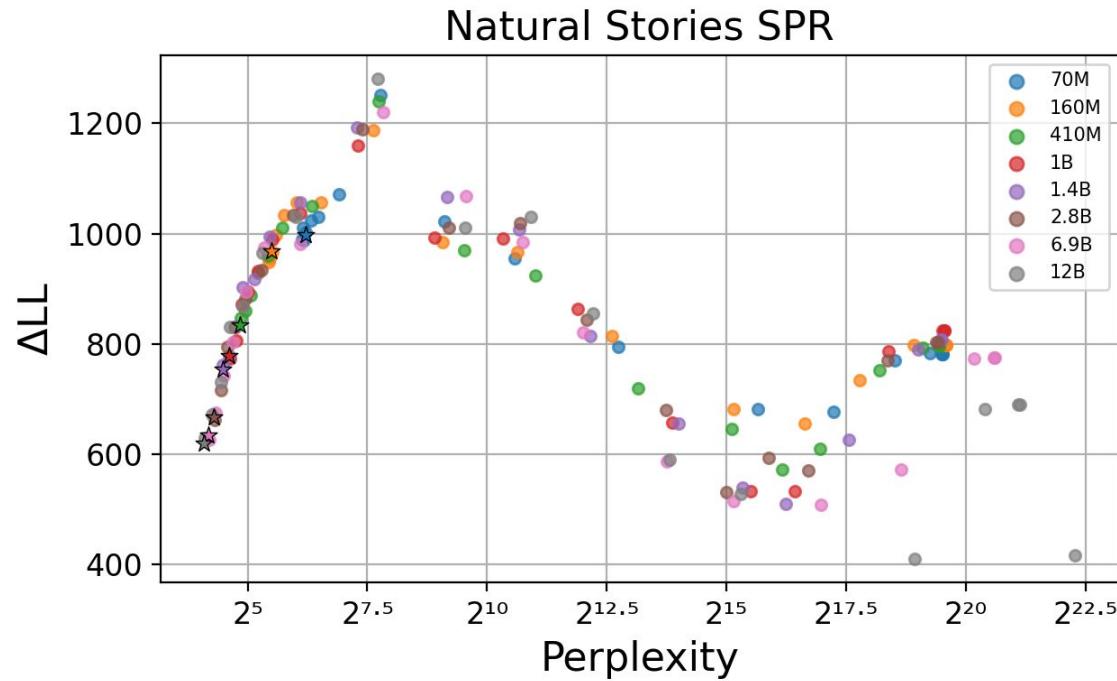
More
data

The two studies captured two different regimes

Better fit
to RT



Poorer fit
to RT



More
accurate



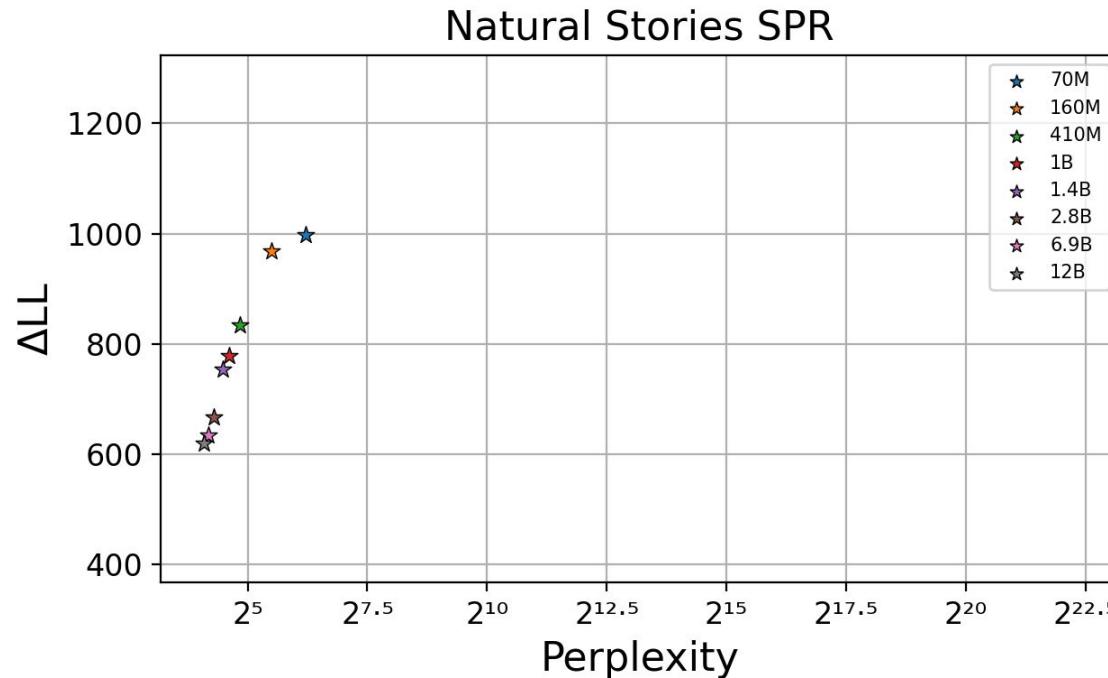
Less
accurate

The two studies captured two different regimes

Better fit
to RT



Poorer fit
to RT



More
accurate



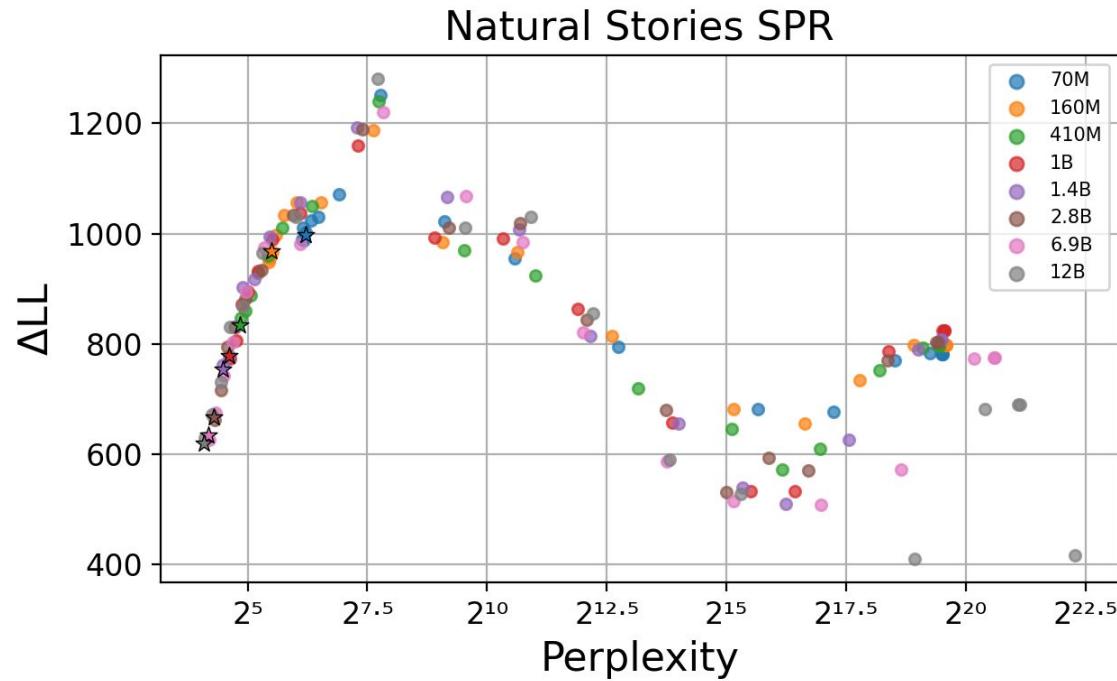
Less
accurate

The two studies captured two different regimes

Better fit
to RT



Poorer fit
to RT



More
accurate



Less
accurate

Covering the middle ground (model size)

- (Much) smaller LMs trained following the procedures of Pythia LMs
- LMs evaluated after $\{1, 2, 4, \dots, 512, 1000, 1500, \dots, 10000\}$ training steps

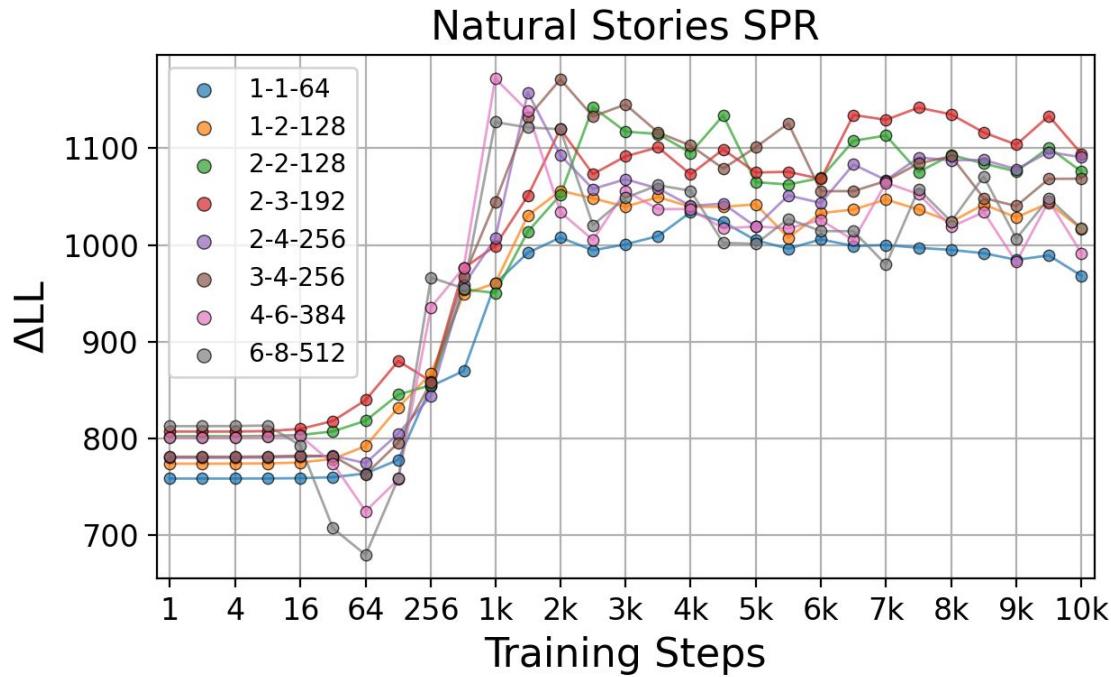
Model	Model size (#Parameters)
Repro 1-1-64	~6M
Repro 1-2-128	~13M
Repro 2-2-128	~13M
Repro 2-3-192	~20M
Repro 2-4-256	~27M
Repro 3-4-256	~28M
Repro 4-6-384	~46M
Repro 6-8-512	~70M

Smaller LMs converge earlier

Better fit
to RT



Poorer fit
to RT



Less
data



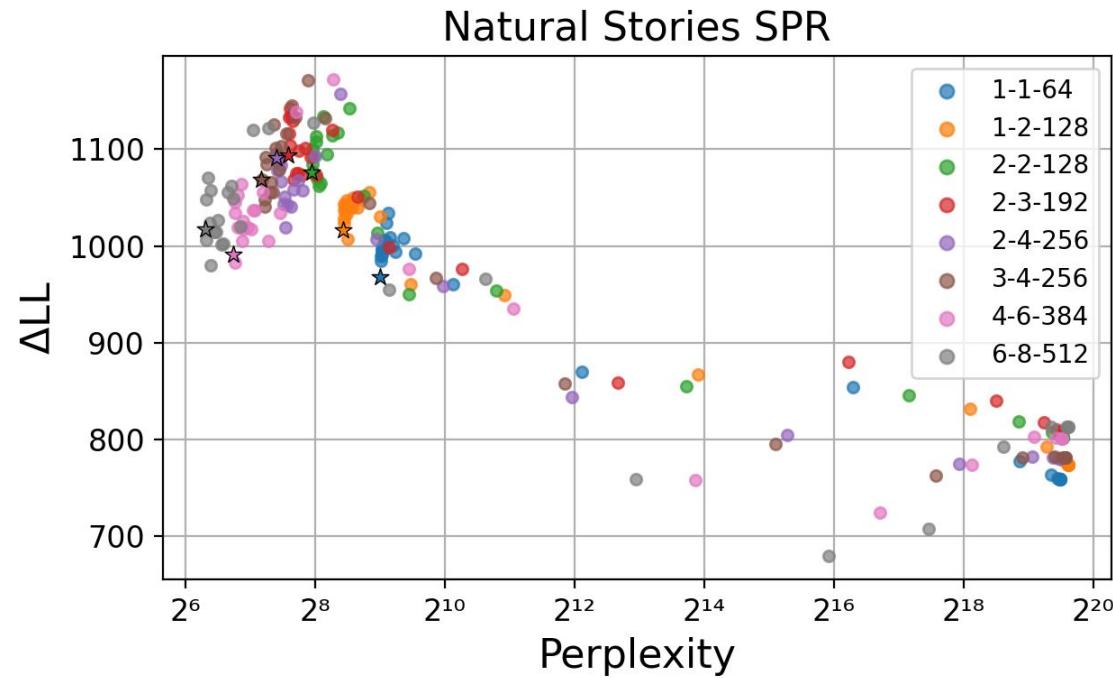
More
data

The two different regimes, again

Better fit
to RT



Poorer fit
to RT



More
accurate



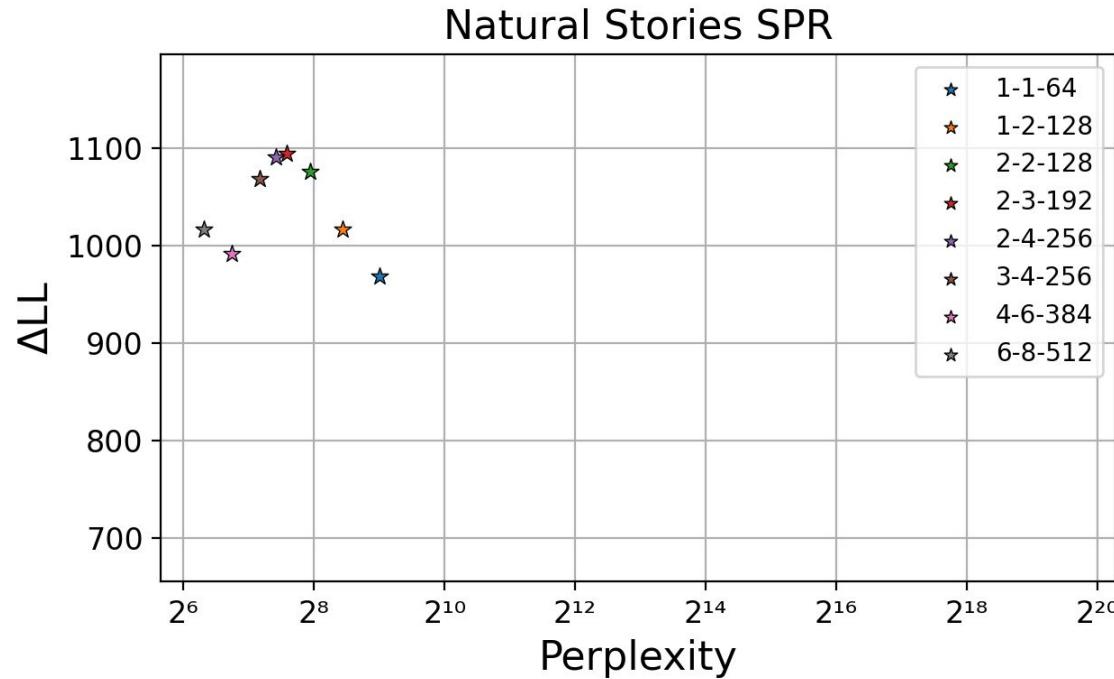
Less
accurate

The two different regimes, again

Better fit
to RT



Poorer fit
to RT



More
accurate



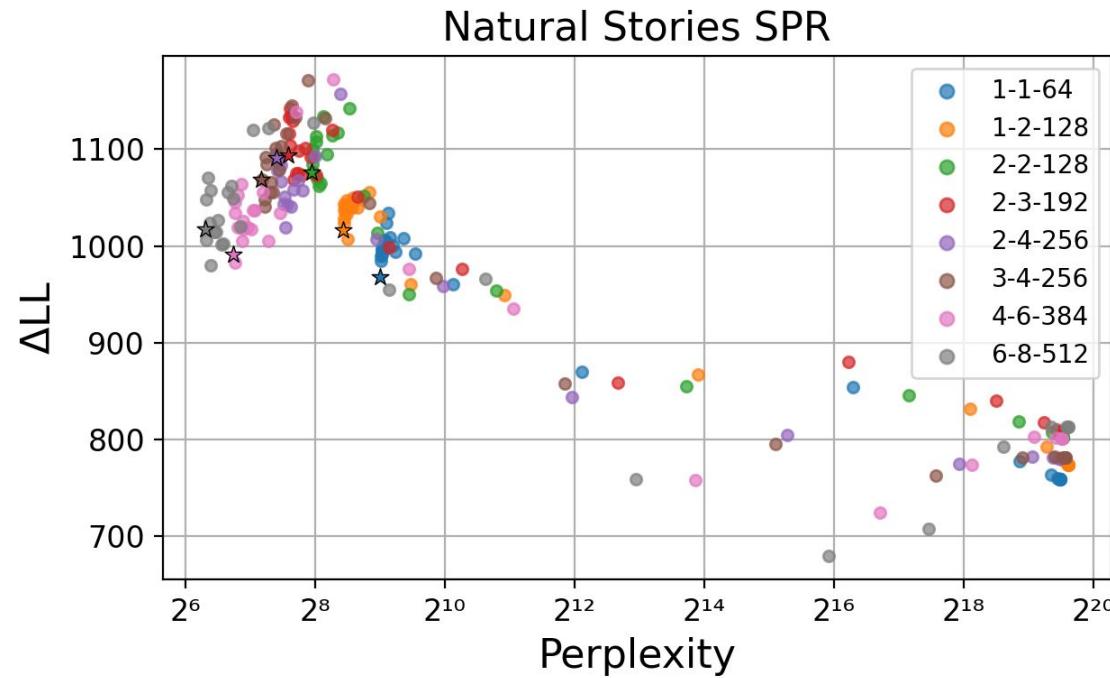
Less
accurate

The two different regimes, again

Better fit
to RT



Poorer fit
to RT



More
accurate



Less
accurate

Recap (2)

- Fit to reading times starts to degrade after about 2B tokens of training data
- Strong interaction between LM size and training data amount after the peak
- Consolidates conflicting results about LM perplexity and fit to reading times

More recently (1)

'If you were to journey'

Finer granularity, more character-like ($|V| = 256$)

I f y o u _w er e _to _j our n e y

Coarser granularity, more word-like ($|V| = 128000$)

If you were to journey

Figure 1: Smaller subword vocabulary sizes result in longer sequences of finer-granularity tokens that are more character-like (top), and larger vocabulary sizes result in shorter sequences of coarser-granularity tokens that are more word-like (bottom).

Reading times are measured in words, but LMs predict over subword tokens

Manipulated the tokenizer and trained Mamba-2 models with different vocabulary sizes

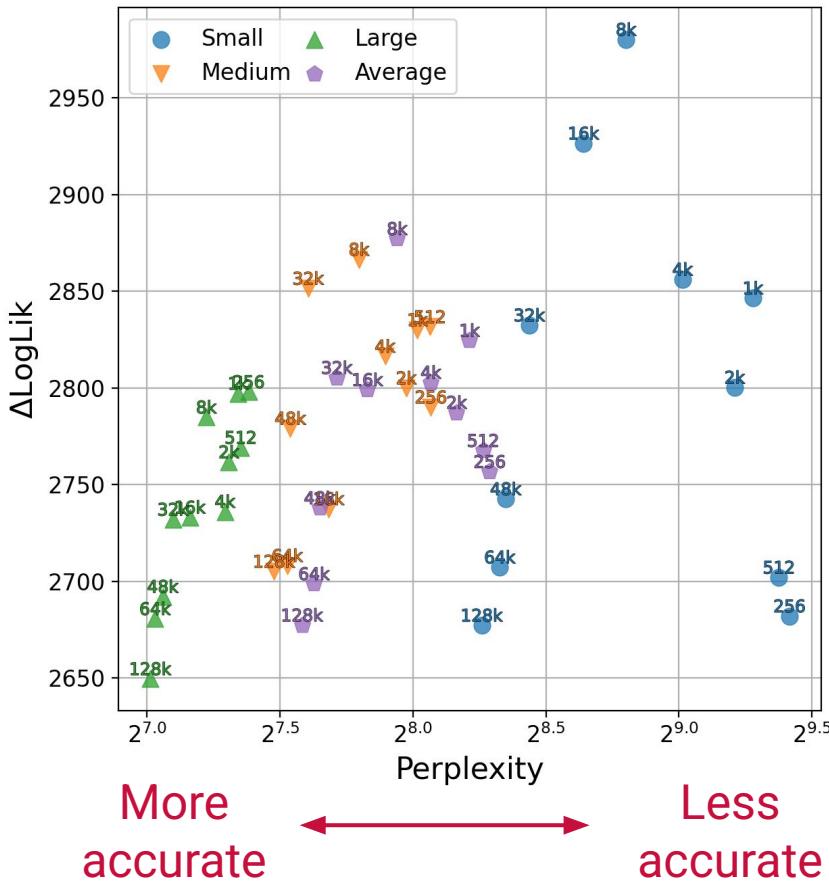
The impact of token granularity on the predictive power of language model surprisal

Again, some sweet spot in the middle?

Better fit
to RT



Poorer fit
to RT



The influence of the tokenizer is more notable for smaller LMs

A vocabulary size of 8k is a lot smaller than what is often used

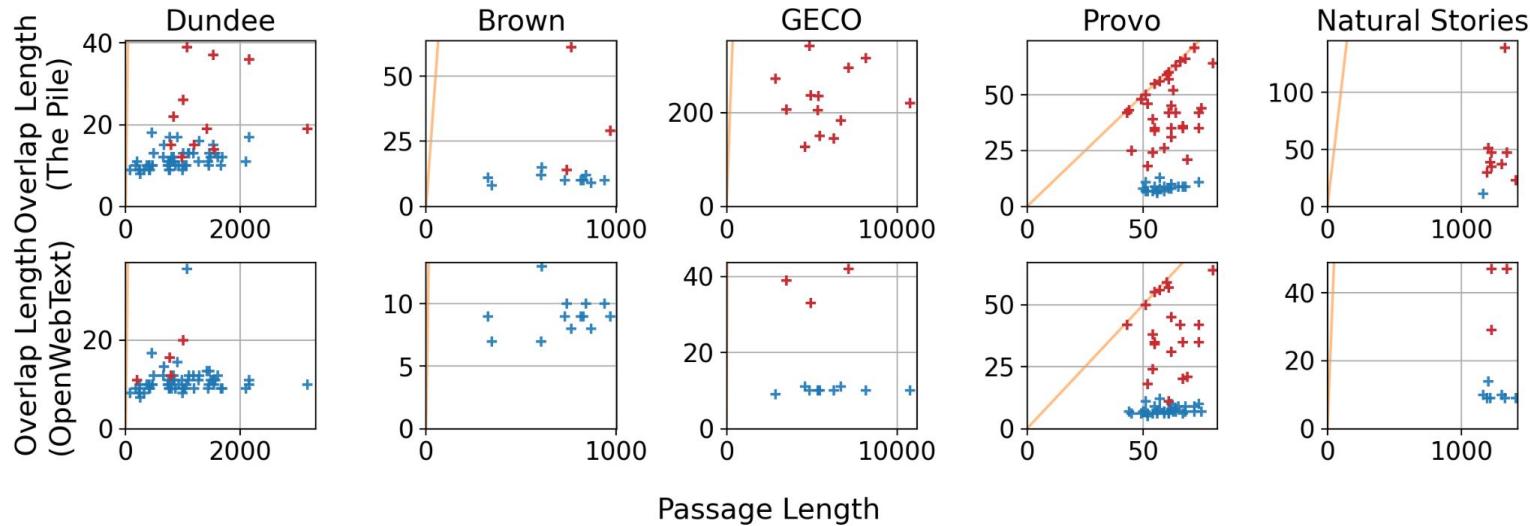
More recently (2)

- Have the LMs already seen the text corpora used to collect reading times from human subjects?
- Does this explain why larger models make more ‘superhuman’ predictions?
- Leakage detection using longest token n -gram overlap

Assessing the leakage of naturalistic reading time corpora in language model pre-training datasets (not available online yet)

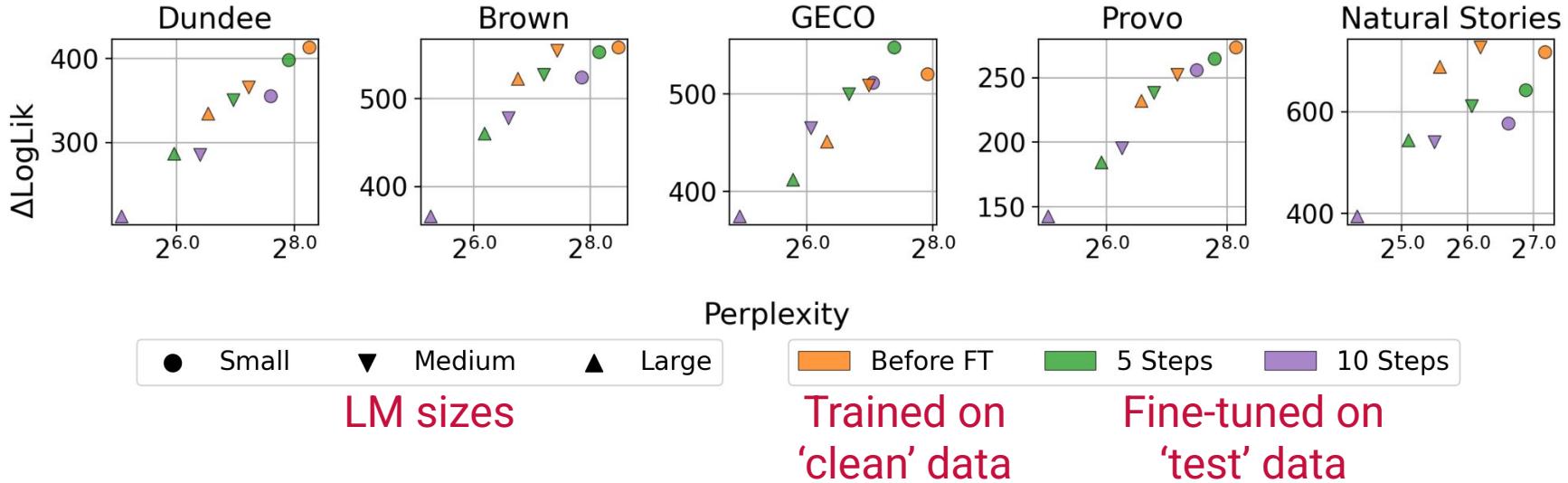
Pythia training
data (300B)

GPT-2 training
data (8.7B)



If the passage is completely leaked in training data, the overlap length should be the same as the passage length (yellow line)

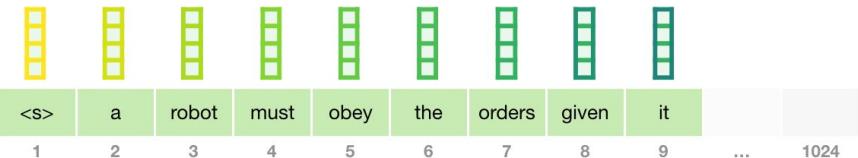
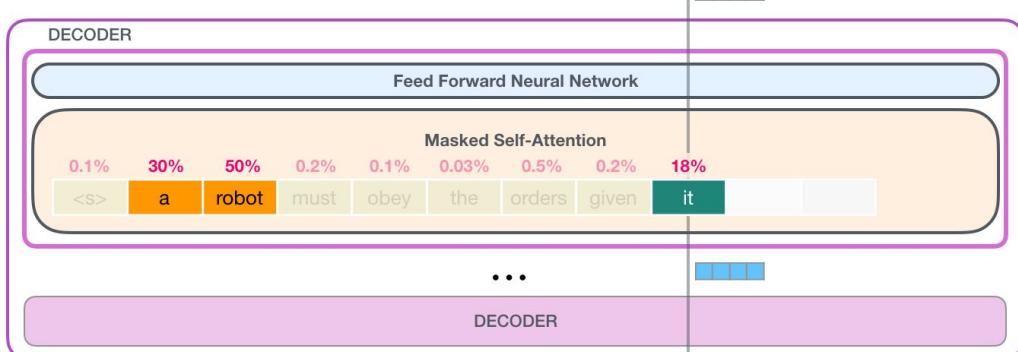
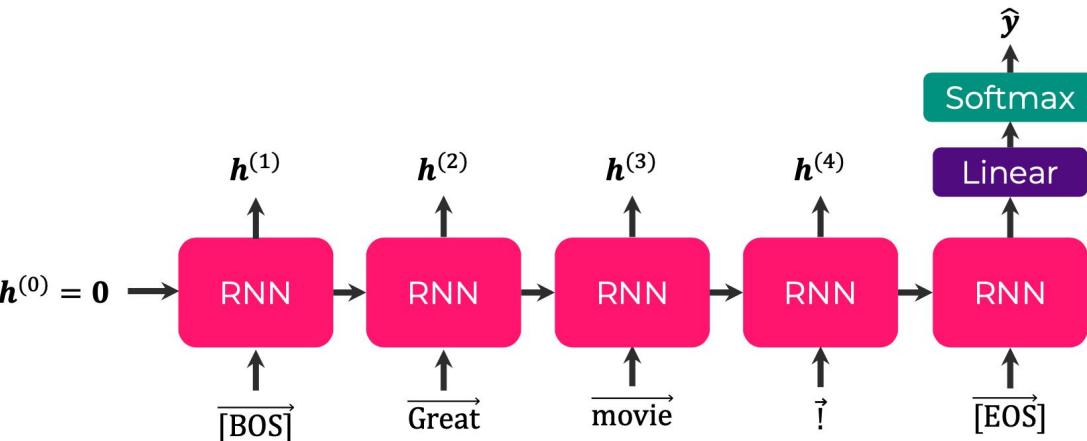
Inevitably some overlap, but benign on most datasets



The effects of model size hold even when trained on very ‘clean’ data

But larger models degrade quicker if directly trained on the reading time corpora

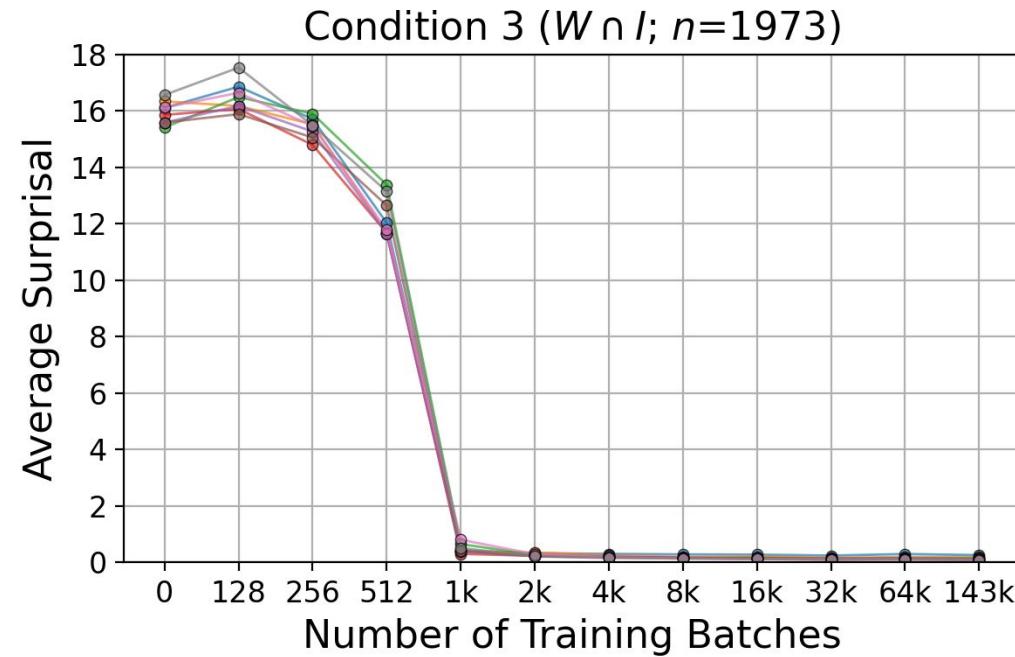
More recently (3)



What are some differences between the two neural networks?

One that's relevant for language processing: Transformers have lossless access to previous tokens

More recently (3)



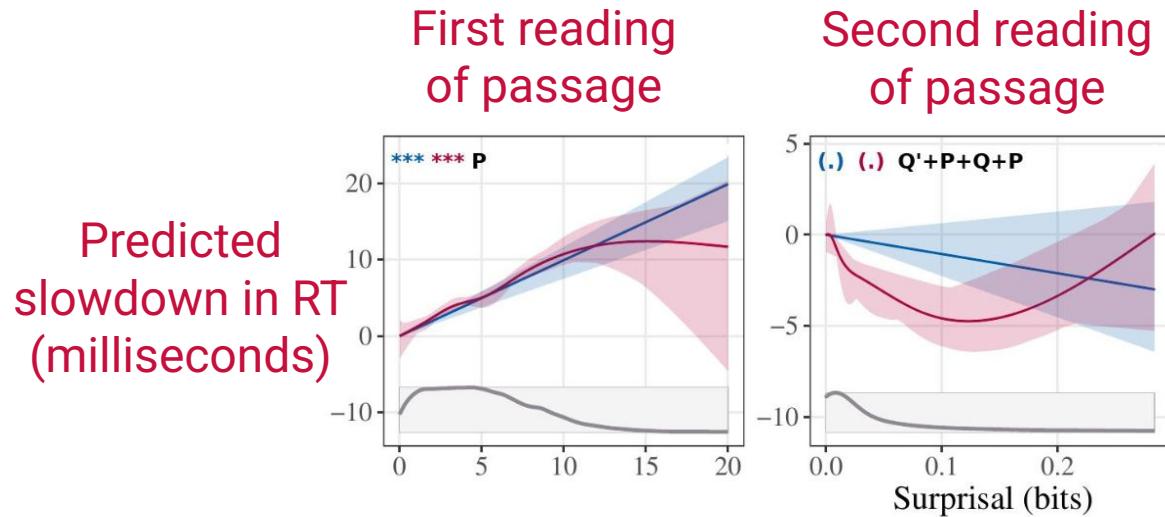
You can see this through how well Transformers are able to 'copy' earlier tokens of the input sequence

Figure shows surprisal at
A B [some intervening tokens] A → B

All models learn to copy by relying on bigram patterns early in training

This is thought to be achieved through induction heads ([Olsson et al. 2022](#))

More recently (3)



Transformer LMs (even Pythia 70M!) predict repeated text with near-zero surprisal

“Surprisal collapse” ([Gruteke Klein et al. 2024](#)): surprisal is unable to capture reading times during repeated reading

More recently (3)

$$\alpha \cdot \begin{bmatrix} 1 & & \\ e^{-\lambda} & 1 & \\ & e^{-\lambda} & 1 \end{bmatrix} + \frac{(1 - \alpha)}{\sqrt{d}} \cdot \begin{bmatrix} q_1 k_1 & & \\ q_2 k_1 & q_2 k_2 & \\ q_3 k_1 & q_3 k_2 & q_3 k_3 \end{bmatrix}$$

(a) de Varda and Marelli (2024)

$$m \cdot \begin{bmatrix} 0 & & \\ -1 & 0 & \\ -2 & -1 & 0 \end{bmatrix} + \frac{1}{\sqrt{d}} \cdot \begin{bmatrix} q_1 k_1 & & \\ q_2 k_1 & q_2 k_2 & \\ q_3 k_1 & q_3 k_2 & q_3 k_3 \end{bmatrix}$$

(b) Press et al. (2022)

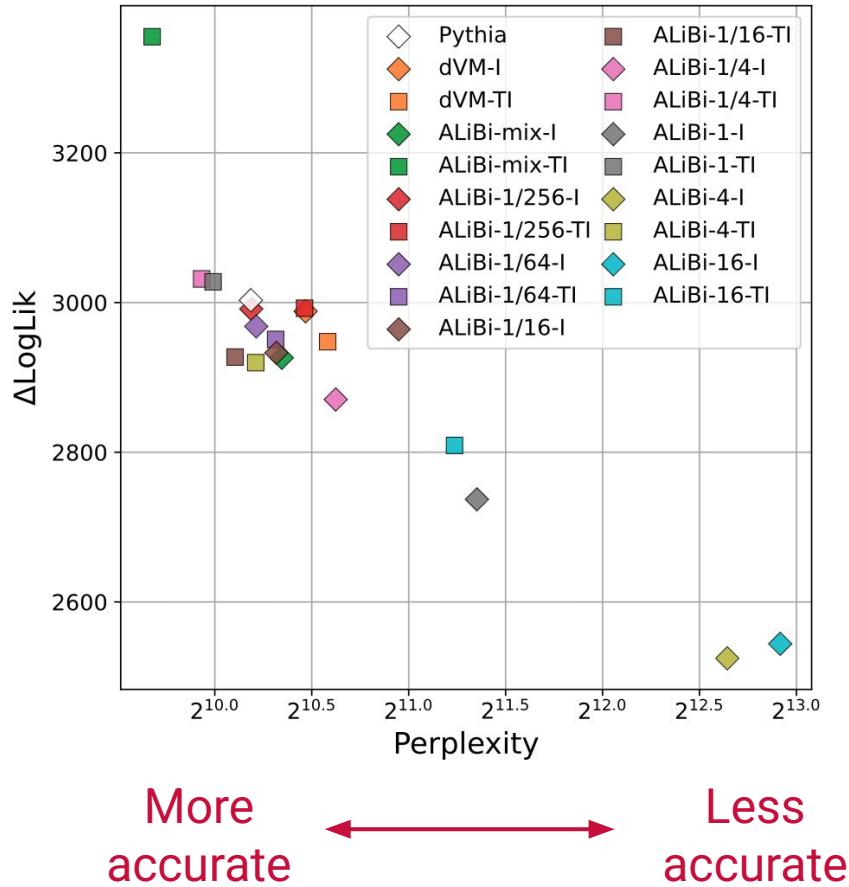
We can bake in a **recency bias** by intervening on the attention weights

This has the effect of **downweighting earlier tokens** in the input context

Linear recency bias during training improves Transformers' fit to reading times

Recency bias helps, under specific circumstances

Better fit
to RT



Recency bias needs to be incorporated both during training and inference

Different heads need to have different decay rates

- Seems to help track different dependencies

TL;DR

Computational psycholinguistics aims to develop computational models of the cognitive mechanism underlying language comprehension

LMs can be evaluated as models of predictive processing under surprisal theory by using surprisal to model reading times

Modern Transformer LMs seem to be ‘superhuman’ for two reasons

- During training: Models learn too much parametric knowledge
- During inference: Transformers tend to copy earlier input tokens

In the works

Applying “model editing” techniques for the completely opposite purpose

Using LM surprisal to model reading times in Chinese

- People have to perform word segmentation implicitly

Tweaking state space models to implement human-like limitations in memory

Thanks for listening!

oh.b@nyu.edu