Summer School

Large Language Models for Digital Humanities Research

Track 3: Using LLMs for Psycholinguistic Research

Part 2: Text Generation and Text Analysis

Job Schepens

job.schepens@uni.koeln.de

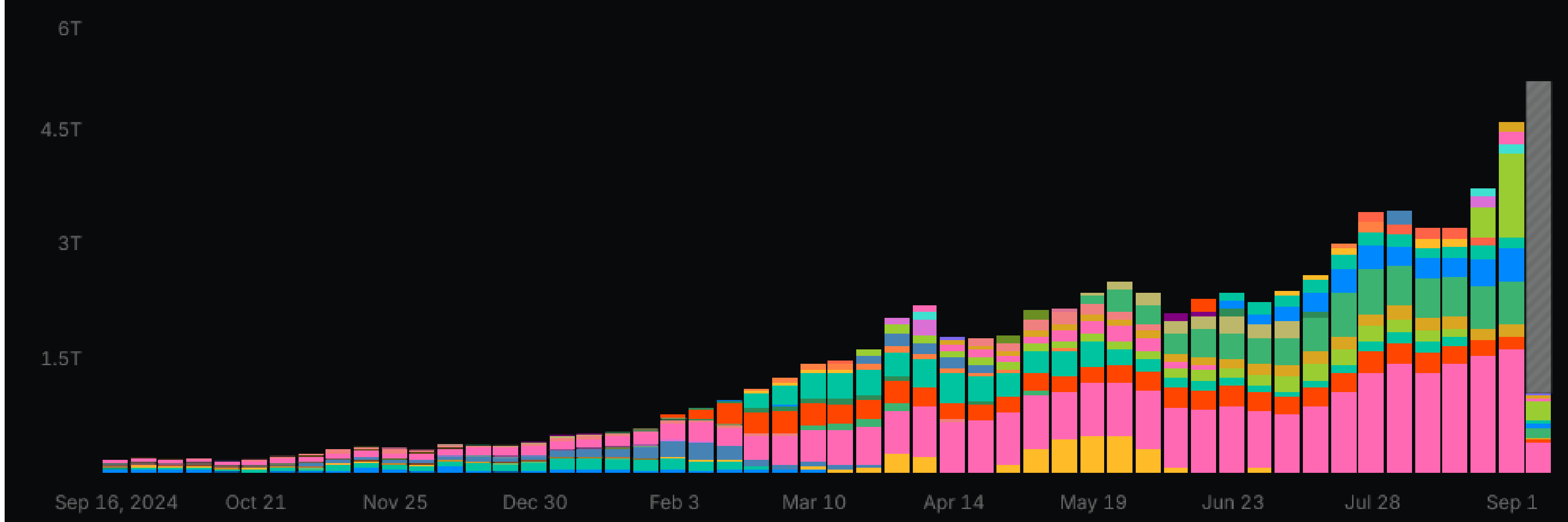
Contents

https://github.com/jobschepens/mlschool-text

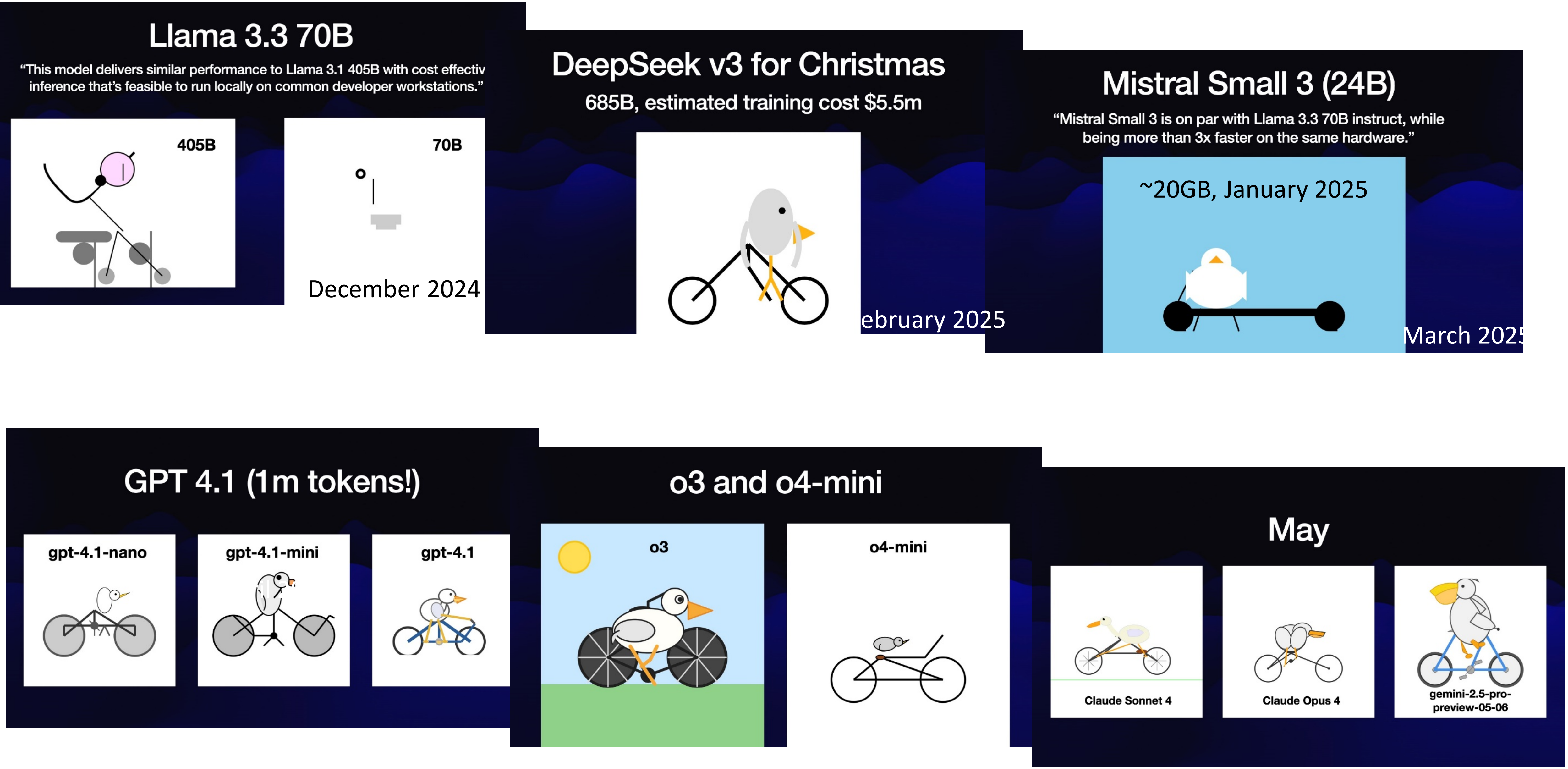
Leaderboard

Top this week

Token usage across models on OpenRouter



	Model	Tokens	Change
1.	Grok Code Fast 1 by x-ai	1.15T tokens	↑115%
2.	Claude Sonnet 4 by anthropic	564B tokens	↑3%
3.	Gemini 2.5 Flash by google	325B tokens	↓31%
4.	Gemini 2.0 Flash by google	181B tokens	↓11%
5.	GPT-4.1 Mini by openai	173B tokens	↑387%
6.	DeepSeek V3 0324 by deepseek	170B tokens	↑12%
7.	DeepSeek V3.1 (free) by deepseek	149B tokens	↑540%
8.	Gemini 2.5 Pro by google	148B tokens	↓18%
9.	Qwen3 30B A3B by qwen	113B tokens	↓4%
10.	GPT-5 by openai	85.4B tokens	↑15%



Simon Willison (2025), AI Engineer World's Fair

Current developments in NLP using LLMs

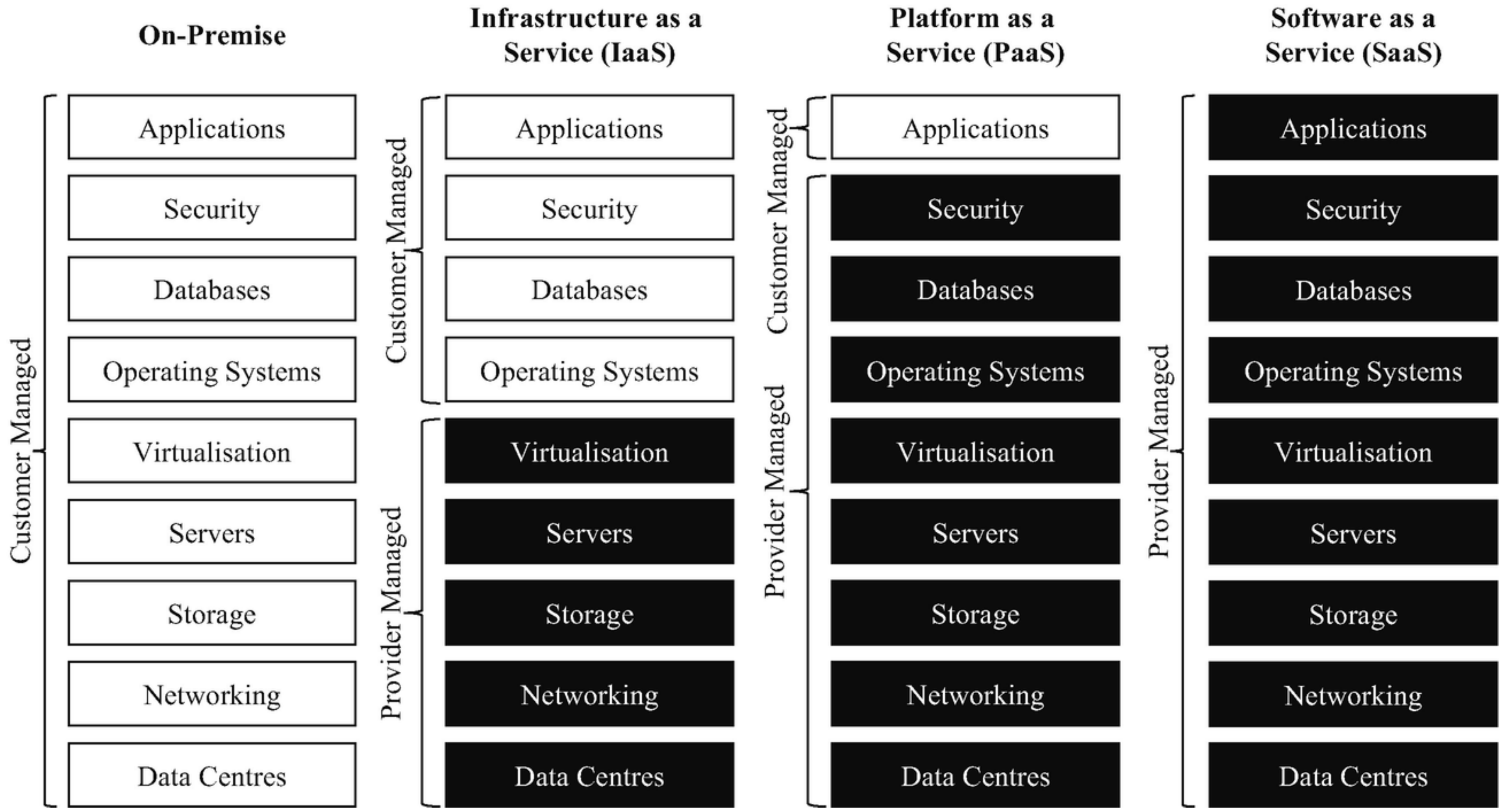
- Many current developments…
  - Reasoning, agentic capabilities, larger context windows, multimodal integration, open weight models
- Many (immature) possibilities
  - As (safe?) **tools**: e.g. stimulus generation, automatic annotation, word frequency estimation
  - As (robust?) **models** (of what?): e.g. next word prediction / surprisal process and reading times / cloze probability
- But also many responsibilities (e.g. keynote by Elen le Foll)
  - Ethics, code of conduct, standards, best practices (transparency, human verification, etc.)
- LLMs in Cologne Language
  - SFB1252 Research Data & Methods, SFB1252 Brown Bag Lunches, Reproducibilitea, DH Colloquium, UzK Data Steward Network, Informal discussion groups, etc.
  - Ongoing research: upcoming "LLMs for linguistic analysis" workshop in Cologne 24-25 November (more info soon)

Ollama
llama.cpp
expensive
hardware

AWS EC2, Google
compute engine
can be expensive

Only for small models:
e.g. Gradio app with
LLaMA-7B on Google Colab,
Binder, Github codespaces

Hugging Face endpoints
together.ai
openrouter.com
chat.openai.com
claude.com



https://en.wikipedia.org/wiki/Software_as_a_service#/media/File:Comparison_of_on-premise,_IaaS,_PaaS,_and_SaaS.png

Vibe coded with Claude Sonnet 4

LLMs for Experimental Stimulus Generation: Example 1

Generating sentences, controlling for syntax, frequency, and meaning:

- Syntactic complexity: Prompt engineering to generate sentences with specific syntactic structures (e.g., "Generate simple **SVO sentences**" vs. "Generate sentences with **embedded relative clauses**")
- Word frequency: Fine-tuning on frequency-controlled corpora to **maintain target word frequency ranges**
- Semantic content manipulation: Using **LLMs or embeddings**
- Multi-constraint generation: Simultaneous control of **multiple variables** (e.g., "Generate high-frequency words in complex syntactic structures about cooking")

```python
# Precise, topic-aware prompt engineering
prompts = {
    ('SVO', 'high', 'cooking'):
        "Write a simple sentence about cooking using common words:",
    ('embedded', 'low', 'AI research'):
        "Generate a sentence about artificial intelligence research with academic
    ('direct_object', 'medium', 'social'):
        "Create a sentence about social interaction with moderate vocabulary that
}

# spaCy-based syntactic analysis for validation
def analyze_syntax(self, sentence: str) -> Dict:
    doc = self.nlp(sentence)

    # Detect embedded relative clauses
    relative_clauses = [token for token in doc if token.dep_ == 'relcl']

    # Count direct objects
    direct_objects = [token for token in doc if token.dep_ == 'dobj']
```

```python
def generate_multiple_candidates(self, prompt: str, num_candidates: i
    candidates = []
    inputs = self.tokenizer.encode(prompt, return_tensors='pt')

    for _ in range(num_candidates):
        outputs = self.model.generate(
            inputs,
            max_length=len(inputs[0]) + 25,
            temperature=0.9,          # Controlled randomness
            top_p=0.9,               # Nucleus sampling
            repetition_penalty=1.1,   # Avoid repetition
            do_sample=True
        )

        # Extract and clean first sentence
        generated = self.tokenizer.decode(outputs[0], skip_special_tc
        sentence = generated[len(prompt):].split('.')[0] + '.'
        candidates.append(sentence)
```

Generated Sentence	Syntactic Target	Frequency Target	Topic	Syntax Match	Frequency Match	Topic Score	All Constraints Met	Attempts Needed	Dependency Depth	Length
SVO + HIGH FREQUENCY + COOKING										
The time is coming.	SVO	HIGH	cooking and foo...	✓ YES	✓ YES	0.722	🏆 PERFECT	96	2	5
A person's name is your number.	SVO	HIGH	cooking and foo...	✓ YES	✓ YES	0.714	🏆 PERFECT	139	3	8
The chicken is cooked very well.	SVO	HIGH	cooking and foo...	✓ YES	✗ NO	0.795	⚠ PARTIAL	117	2	7
The recipe includes vegetables.	SVO	HIGH	cooking and foo...	✓ YES	✗ NO	0.792	⚠ PARTIAL	78	2	5
a chicken curry or something.	SVO	HIGH	cooking and foo...	✓ YES	✗ NO	0.786	⚠ PARTIAL	68	1	7
EMBEDDED RELATIVE + LOW FREQUENCY + AI RESEARCH										
Machine Learning (MLA) is the field where computer science becomes more interesting as it progresses through each area of study.	EMBEDDED_RELATIVE	LOW	artificial inte...	✓ YES	✓ YES	0.875	🏆 PERFECT	11	7	23
say in the example where AI may be used as well.	EMBEDDED_RELATIVE	LOW	artificial inte...	✓ YES	✓ YES	0.795	🏆 PERFECT	24	5	12
It is conceivable to design something from the data it can learn.	EMBEDDED_RELATIVE	LOW	artificial inte...	✓ YES	✓ YES	0.810	🏆 PERFECT	29	6	13
The next thing I want to do is talk about the future.	EMBEDDED_RELATIVE	LOW	artificial inte...	✓ YES	✓ YES	0.780	🏆 PERFECT	32	4	13
In order to learn how the human mind works we need an environment that can understand our emotional state.	EMBEDDED_RELATIVE	LOW	artificial inte...	✓ YES	✓ YES	0.816	🏆 PERFECT	33	6	20
DIRECT OBJECTS + MEDIUM FREQUENCY + SOCIAL										
I think this is why we love each other, or the relationship I live in will change.	DIRECT_OBJECTS	MEDIUM	social interact...	✓ YES	✓ YES	0.747	🏆 PERFECT	3	5	21
The word emotional is often used to describe the process by which people feel themselves felt.	DIRECT_OBJECTS	MEDIUM	social interact...	✓ YES	✓ YES	0.774	🏆 PERFECT	5	5	19
The most successful people are those who have an interest in what they're doing.	DIRECT_OBJECTS	MEDIUM	social interact...	✓ YES	✓ YES	0.778	🏆 PERFECT	7	6	16
We want to write as many words per minute in an hour or two of time.	DIRECT_OBJECTS	MEDIUM	social interact...	✓ YES	✓ YES	0.723	🏆 PERFECT	9	6	17
I have an opinion on this person, He is very nice or simply You are extremely polite.	DIRECT_OBJECTS	MEDIUM	social interact...	✓ YES	✓ YES	0.695	🏆 PERFECT	11	4	24

Vibe coded Code with Claude Sonnet 4
Sentences generated using GPT-2

Vibe coded with Claude Sonnet 4

LLMs for Experimental Stimulus Generation: Example 2

Generating **minimal pairs**

- Phonological: Generating words using an **encoder model (e.g. BERT)** to control semantic plausibility and filter for similarity in onset, length, stress, tone, vowel quality, voicing, aspiration, …
- Morphological: Generation of **inflectional and derivational minimal pairs** (e.g., walk/walked, happy/happiness)
- Syntactic: Generating sentences that differ only in **target syntactic structures**
- Cross-modal minimal pairs: Generating **text-image pairs** with controlled linguistic-visual correspondences

Pipeline:

1. **Template Creation**

```
"The [MASK] was parked outside"
```

2. **BERT Prediction**

```
candidates = model.predict(template)
# → car, van, truck, bmw, cab...
```

3. **Phonological Filtering**

```
filter_by_contrast_type(
  target="car",
  candidates=candidates,
  type="coda"
)
```

Controls:

- Edit distance = 1
- Position-specific contrasts
- Semantic plausibility
- Frequency matching

Context 1: *"The red [MASK] was parked outside"*

- car → cab (prob: 0.018)
- car → cat (prob: 0.003)

Context 2: *"She drove her [MASK] to work today"*

- car → cat (prob: 0.001)

Context 3: *"The [MASK] needs new tires"*

- car → cat (prob: 0.002)
- car → cart (prob: 0.001)
- car → cab (prob: 0.001)

Generated: 6 minimal pairs across 3 contexts

Interim Summary: LLMs for stimulus generation

Strengths

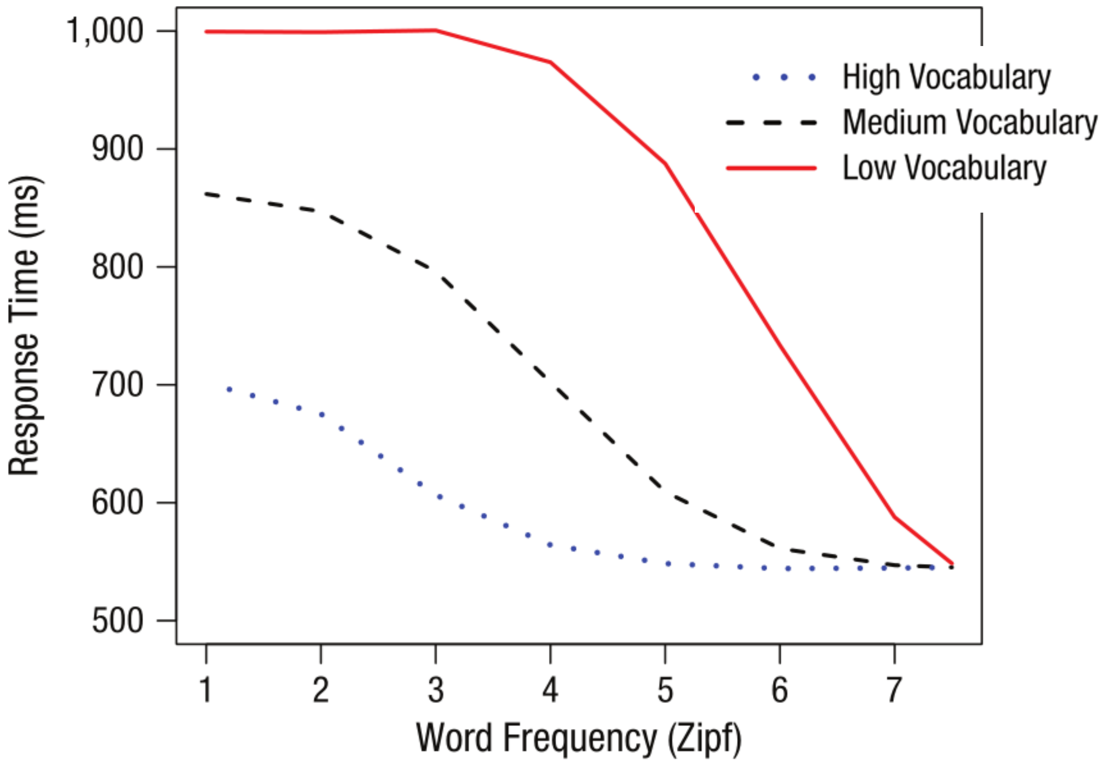
- **Scalability**: Generate many stimuli
- **Consistency and transparency**: Formalized criteria
- **Flexibility**: Easy to regenerate with different criteria
- **Reproducibility**: Documented and version controlled implementation

Weaknesses

- **Quality** : Biases due to LLM training data, fine-tuning, architecture, etc.
- **Lack of Domain Expertise**: Not trained on reasoning about specific linguistic issues
- **Reproducibility**: Due to LLM's stochastic nature, re-generating the code likely results in different stimuli
- **Ethical Concerns**: Possibly generating harmful content
- **Verification:** Human-in-the-loop
- **Black Box:** No mechanistic interpretation possible

Building linguistic corpora for word frequency estimation

- Word frequency is a strong behavioral correlate in visual word recognition paradigms
- Word frequency: counting word occurrences in a corpus
- Corpora used for estimating word frequency are based on text from:
  - Books, Newspapers DWDS, Heisters et al., 2011
  - Subtitles SUBTLEX, Brysbaert et al., 2011
  - German children's books childLex, Schroeder et al. 2015
  - Text generated by LLMs? Schepens et al., PsyArXiv



The Word Frequency Effect in Word Processing: An Updated Review

Current Directions in Psychological Science
1–6

DOI: 10.1177/0963721417727521
www.psychologicalscience.org/CDPS
SAGE

Marc Brysbaert[1], Paweł Mandera[1], and Emmanuel Keuleers[2]
[1]Department of Experimental Psychology, Ghent University, and [2]Department of Communication and Information Sciences, Tilburg University

Two aims

Aim 1: **Compare word frequency** based on text generated by LLMs vs. text written by humans.
→ Human text: ~10 million words in existing corpus of children's books (**ChildLEX**; Schroeder et al., 2015)
→ Measures: correlation, number and percentage of shared words, lexical richness, Zipfs law, etc.

Aim 2: **Compare the estimated word frequency effect** on response times for LLM vs childLex word frequency
→ Lexical decision response times for grade 1-6 and young and old adults (**DeveL**; Schröter & Schroeder, 2017)
→ Measures: Improvement in model fit (AIC) of linear regression models, control for AoA, OLD20, word length

Generating 9 corpora (“conditions”)

GPT, DeepSeek, LLama

“Kinder” vs. “Erwachsene”

```
prompt=[
        {
         "role": 'system",
         "content": "4000 Wörter zu "
         + titel
         + " auf Deutsch geschrieben"
         + " für Kinder im Alter "
         + age_range
        }
    ]
```

```
openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=prompt,
        temperature=0.5,
        max_tokens=4000,
        n=4,
        stop=None,
        frequency_penalty=0,
        presence_penalty=0
```

Continue until 4000 words

0.5 vs. 0.7

Generating 9 corpora

1. 1 corpus: GPT 3.5
2. 2x2 corpora: 2 temperatures (low, high) and 2 target audiences (child-directed, adult-directed prompt)
3. 2x2 corpora: 2 open weight models (DeepSeek V1, Llama 3.3 70B) and 2 text lengths (short, long)

GPT, DeepSeek, LLama

“Kinder” vs. “Erwachsene”

```
prompt=[
        {
         "role": "system",
         "content": "4000 Wörter zu "
         + titel
         + " auf Deutsch geschrieben"
         + " für Kinder im Alter "
         + age_range
        }
    ]
```

```
openai.ChatCompletion.create(
          model="gpt-3.5-turbo",
          messages=prompt,
          temperature=0.5,
          max_tokens=4000,
          n=4,
          stop=None,
          frequency_penalty=0,
          presence_penalty=0
```

Continue until 4000 words

0.5 vs. 0.7

Example text (GPT 3.5)

Generated text: Das rote Phantom

Es war einmal ein kleiner **Junge namens Max**. Max war sieben Jahre alt und lebte in einer kleinen Stadt am Rande des Waldes. Er liebte es, draußen zu spielen und **Abenteuer** zu erleben.

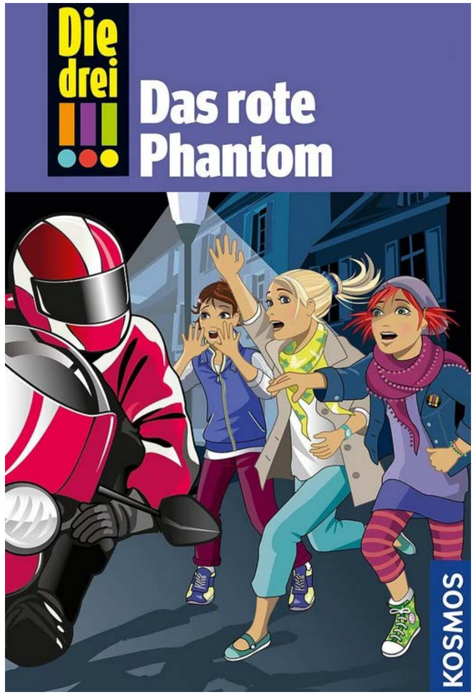
Eines Tages, als Max im Wald spielte, hörte er ein seltsames Geräusch. Es war ein leises Knurren, das immer lauter wurde. Max drehte sich um und sah ein rotes Phantom auf ihn zukommen. Es hatte riesige Augen und scharfe Krallen.

Max war zuerst erschrocken, aber dann erkannte er, dass das rote Phantom nur ein kleiner Hund war. Der Hund war verängstigt und hungrig. Max beschloss, ihm zu helfen.

Klappentext

Kim, Franzi und Marie sind Die drei !!!. Mutig und clever ermitteln die drei Freundinnen und sind jedem Fall gewachsen.

Ein Phantom geht um! Nachts stellt ein **vermummter Motorradfahrer** auf seiner roten Maschine Verbrecher und vereitelt ihre Straftaten. Als ein dreister Dieb Franzis Rucksack klaut, ist das rote Phantom sofort zu Stelle. Der Ehrgeiz der drei Detektivinnen ist geweckt – wer ist das rote Phantom? Bei einer wilden Verfolgungsjagd durch die Nacht kommen Kim, Franzi und Marie hinter das Geheimnis und erleben eine erstaunliche Überraschung.



Example text (Llama 3.3 70B)

Generated text: Mein erster Schultag und der Eisbär-Schreck

Es war ein sonniger Morgen im September. Ich, Tim, war so aufgeregt, dass ich heute meinen ersten Schultag hatte. Meine Mutter half mir, mein neues Schulranzen zu packen, und mein Vater machte mir ein leckeres Frühstück.

Als wir zur Schule kamen, sah ich viele neue Gesichter. Ich fühlte mich ein bisschen schüchtern, aber meine Mutter gab mir einen großen Kuss und sagte: "Alles wird gut, Tim. Du wirst viele neue Freunde finden."

Ich ging in meine Klasse und sah, dass alle anderen Kinder auch ein bisschen nervös waren. Unser Lehrer, Herr Müller, stellte sich vor und erzählte uns, dass wir heute viele spannende Dinge lernen würden.

Klappentext

Ein fröhliches Mut-mach-Buch für den ersten Schultag

Ben freut sich riesig: Endlich kommt er in die Schule! Aber er hat auch ganz schön Lampenfieber, schließlich wird nach den Ferien alles anders sein: Er muss Plüscheisbärin Sardine zum ersten Mal alleine lassen! Und was wird aus seiner Schultüte – schließlich kann Mama doch gar nicht basteln! Wie gut, dass er Florence kennenlernt, die Bens beste Freundin wird. Denn wenn man zusammen in die Schule kommt, kann gar nicht mehr viel schiefgehen ...

• Mit farbenfrohen Illustrationen von Heike Wiechmann

• Einfühlsame Schilderung der aufregenden Zeit vor dem ersten Schultag

• Ein Vorlesebuch für alle, die auf den ersten Schultag warten

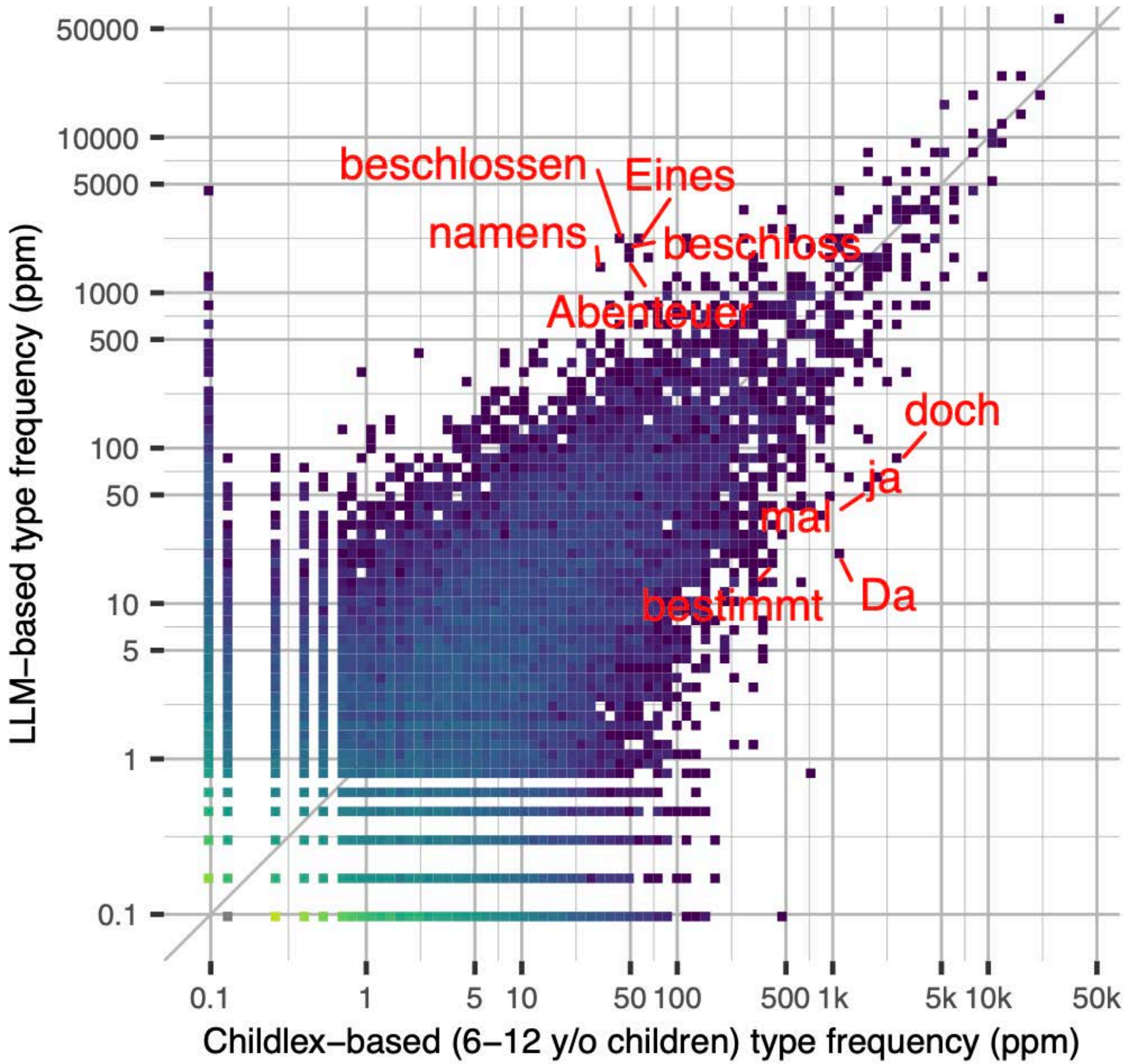
Example text (DeepSeek V3 – long-form text prompt – “ending”)

Mögen wir niemals vergessen, was er uns gelehrt hat: dass die Welt voller Geheimnisse ist, die es zu entdecken gilt, und dass die wahre Stärke oft in den unerwartetsten Gestalten zu finden ist. King-Kong lebt weiter, nicht nur auf der Leinwand, sondern in unseren Herzen und in den Geschichten, die wir von Generation zu Generation weitergeben. Er bleibt ein **Symbol für die ungezähmte Kraft der Natur**, für die Schönheit des Unbekannten und für die unerschütterliche Entschlossenheit, die in jedem von uns schlummert. **King-Kong ist mehr als nur ein Monster oder eine Kreatur** – er ist ein Spiegel unserer eigenen Ängste, Träume und Sehnsüchte.

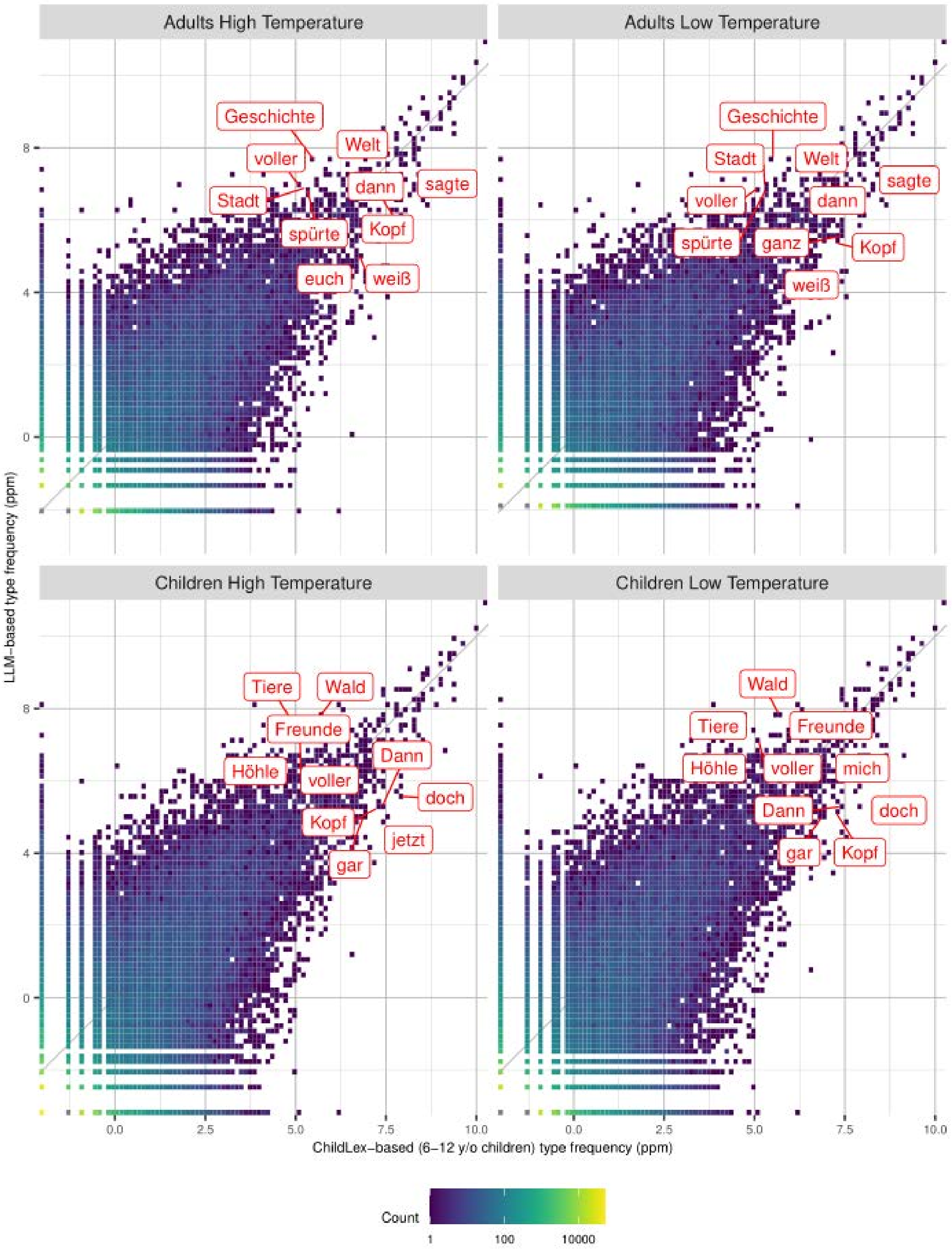
In einer Welt, die oft von Rationalität und Technologie dominiert wird, erinnert er uns daran, dass es noch Raum für das Mystische, das Unerklärliche und das Wunderbare gibt. Er zeigt uns, dass selbst in der Konfrontation mit dem Unbekannten Respekt und Mitgefühl die mächtigsten Werkzeuge sind.

Mögen wir King-Kongs Erbe ehren, indem wir mutig in die unbekannten Welten unserer eigenen Leben vordringen, die Geheimnisse der Natur schützen und niemals aufhören, an die Magie zu glauben, die in jedem Winkel dieser Erde verborgen liegt. Denn solange wir uns an ihn erinnern, wird King-Kong immer bei uns sein – ein stummer Wächter, ein Freund in der Dunkelheit und ein Zeichen dafür, dass die größten Abenteuer oft dort beginnen, wo wir es am wenigsten erwarten.

Word frequency comparison (childLex vs. GPT 3.5)



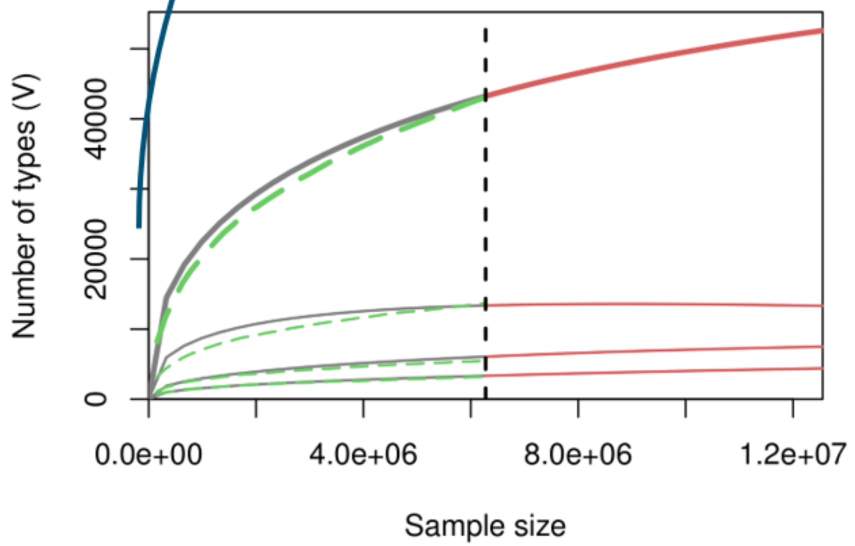
r = .88


Adults High Temperature
Adults Low Temperature
Children High Temperature
Children Low Temperature
Geschichte
Welt
voller
dann
sagte
Stadt
spürte
Kopf
euch
weiß
ganz
Tiere
Wald
Freunde
Dann
Höhle
voller
doch
Kopf
jetzt
gar
mich
LLM-based type frequency (ppm)
ChildLex-based (6–12 y/o children) type frequency (ppm)
0.0
2.5
5.0
7.5
10.0
0
4
8
Count
1
100
10000

Lexical richness comparison (childLex vs. GPT-3.5)

Measure	childLex	LLM-corpus
n Books	500	500
Tokens	9,850,786	6,252,808
Types	182,454	46,409
Lemmas	117,952	34,519

Low lexical richness of LLM corpus



All corpora: Corpus comparison

Measure	childLex	LLM-corpus
n Books	500	500
Tokens	9,850,786	6,252,808
Types	182,454	46,409
Lemmas	117,952	34,519

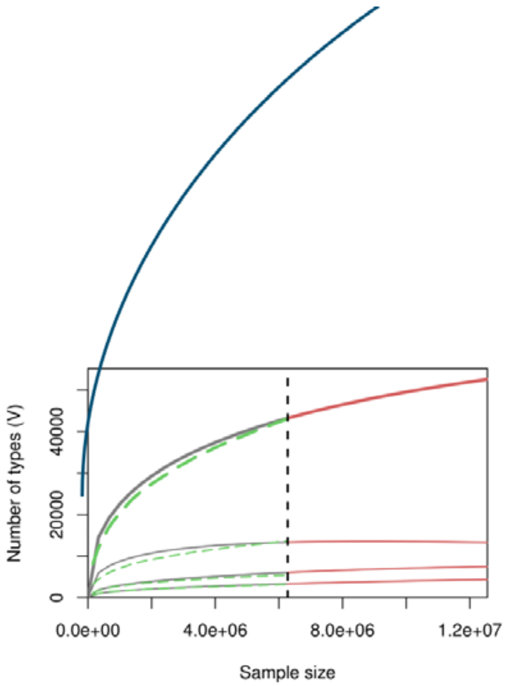
		Adult Corpus		Child Corpus	
Measure	childLex	Low Temp.	High Temp.	Low Temp	High Temp
n Books	500	500	500	500	500
Tokens	9,850,786	7,191,531	7,368,921	23,320,466	23,887,118
Types	182,454	71,423	83,921	84,978	110,603
Lemmas	117,952	52,528	61,318	63,552	82,126

Measure	childLex	Llama long	Llama short	DS-V3 long	DS-V3 short
n Books	500	500	500	500	500
Tokens	9,850,786	10,332,850	7,215,565	9,763,062	7,162,685
Types	182,454	51,320	40,660	239,830	95,321
Lemmas	117,952	39,272	39,272	191,309	74,695

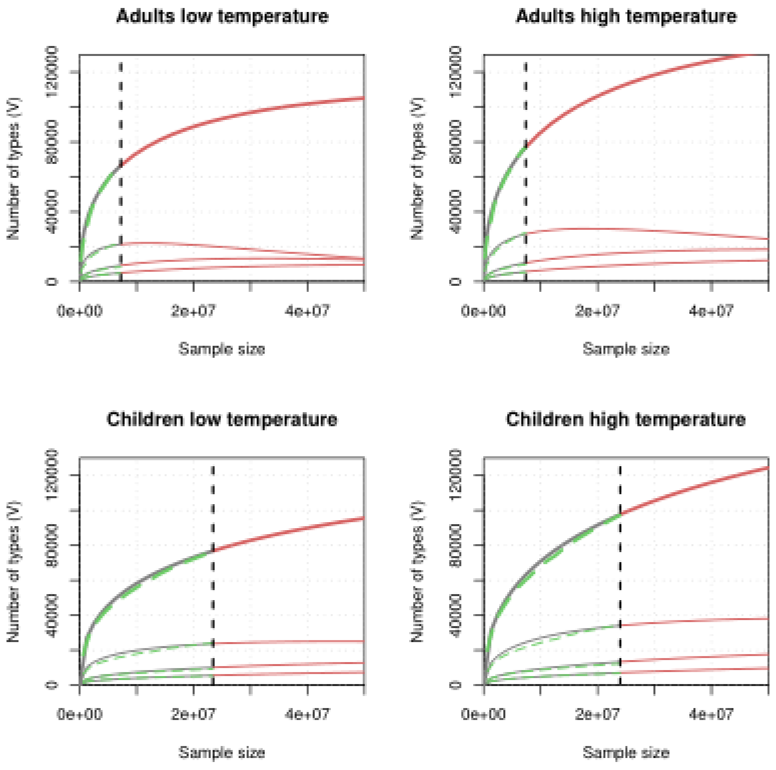
Increase in lexical richness with higher temperature and DeepSeek

All corpora: Corpus comparison

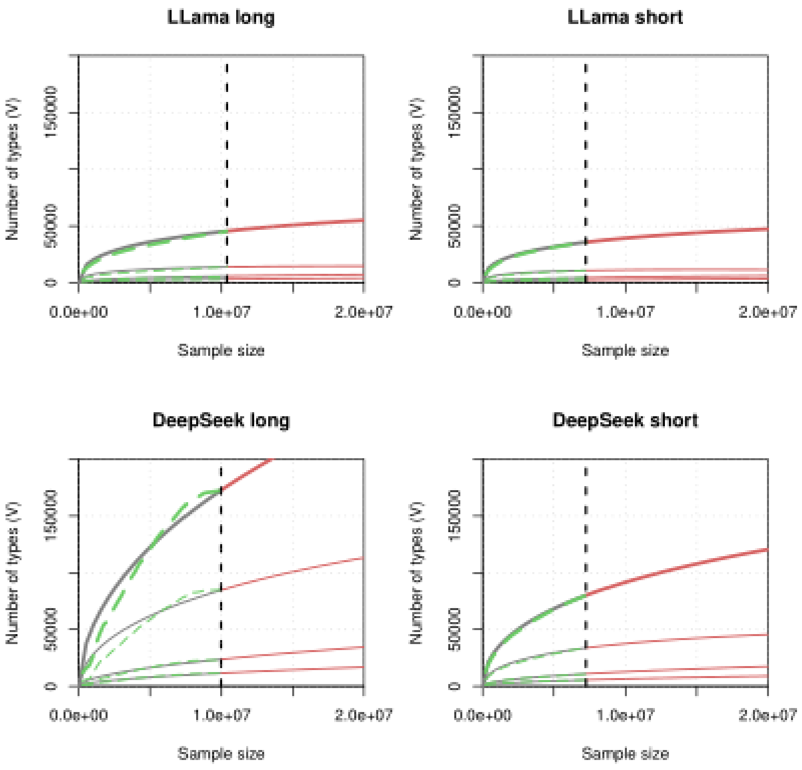
Exp 1



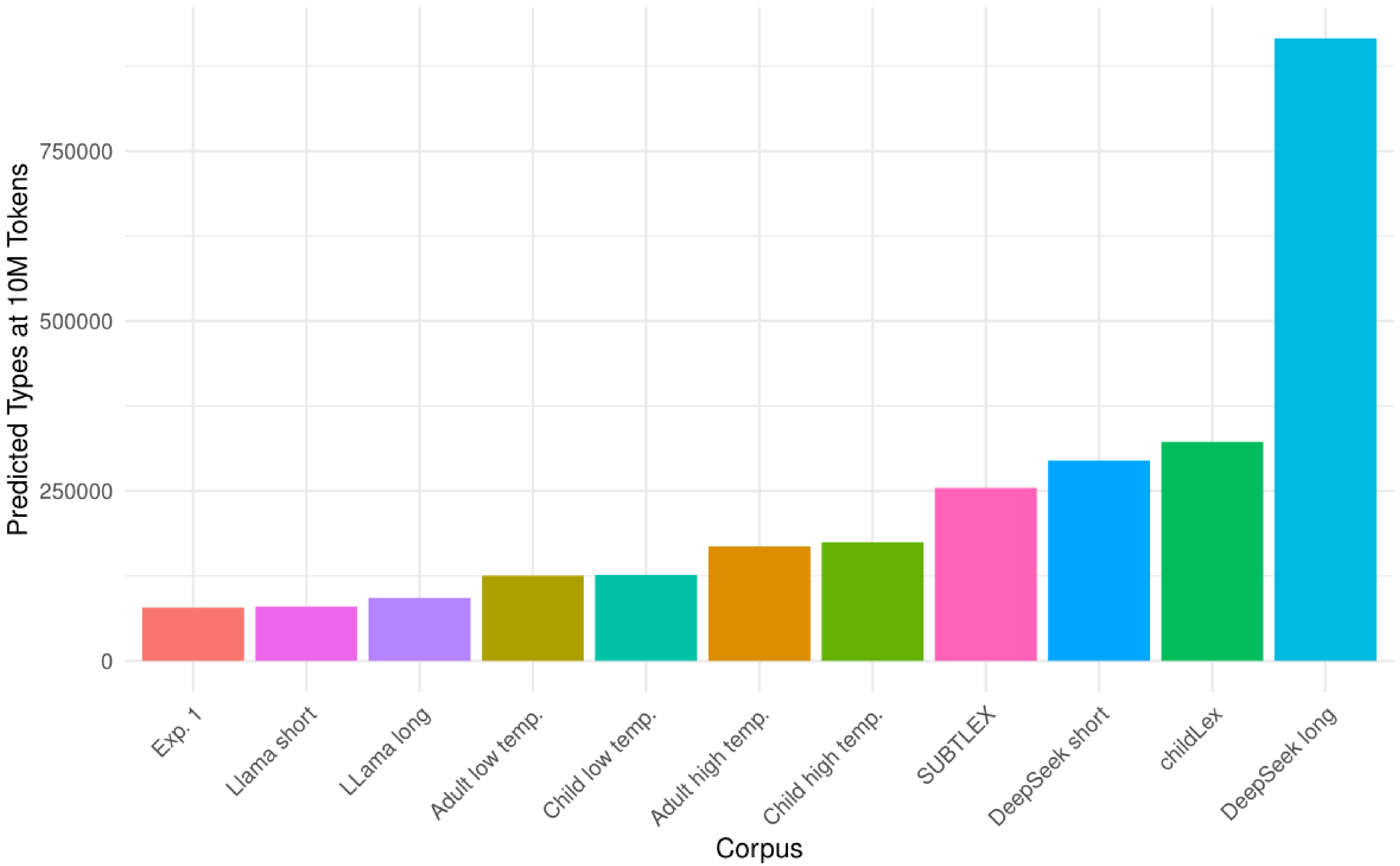
Exp 2

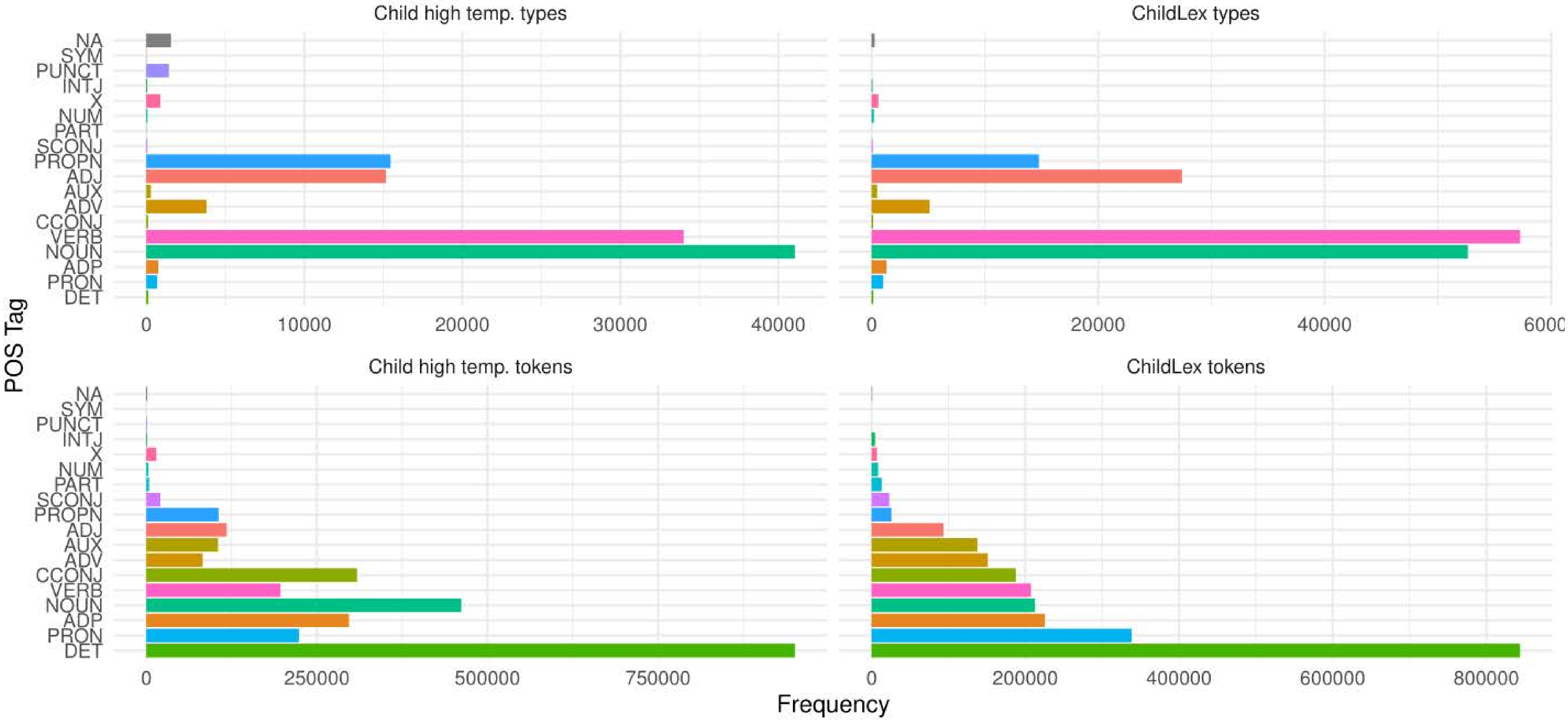


Exp 3

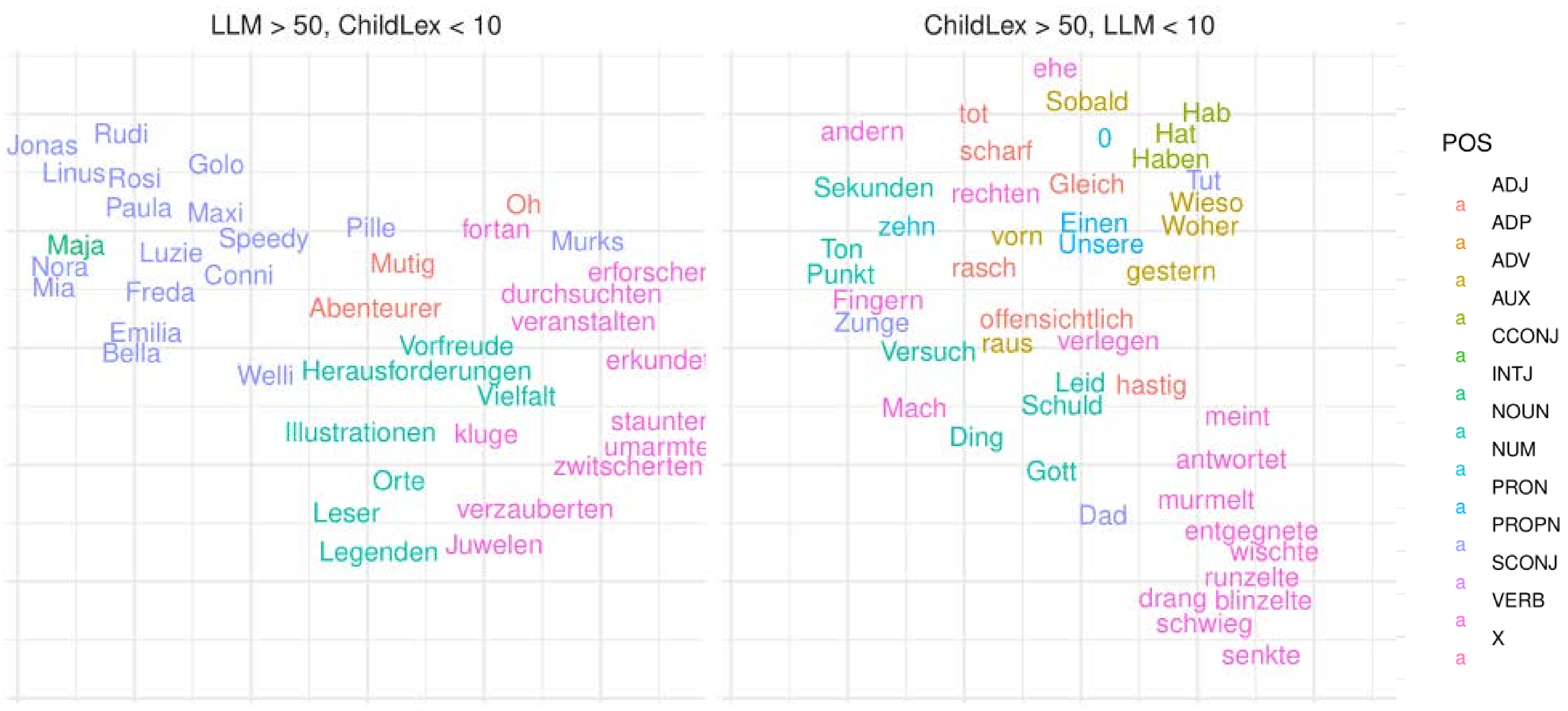


Increase in lexical richness with higher temperature and DeepSeek


Predicted Types at 10M Tokens
750000
500000
250000
0
Exp. 1
Llama short
LLama long
Adult low temp.
Child low temp.
Adult high temp.
Child high temp.
SUBTLEX
DeepSeek short
childLex
DeepSeek long
Corpus



Child high temp. types
ChildLex types
Child high temp. tokens
ChildLex tokens
POS Tag
Frequency
NA
SYM
PUNCT
INTJ
X
NUM
PART
SCONJ
PROPN
ADJ
AUX
ADV
CCONJ
VERB
NOUN
ADP
PRON
DET
0
10000
20000
30000
40000
0
20000
40000
6000
0
250000
500000
750000
0
200000
400000
600000
800000



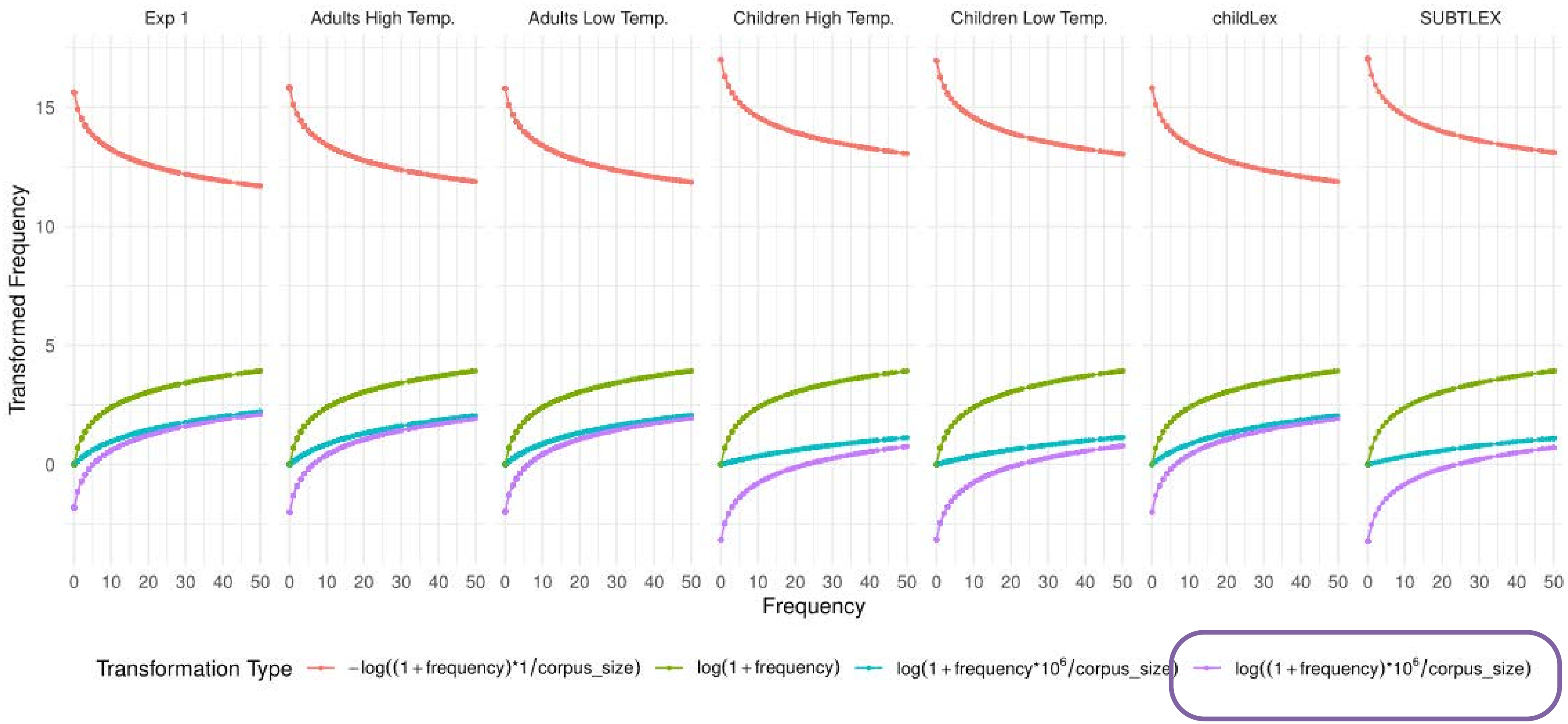
Distributions of word embeddings using dimensionality reduction with UMAP to word vectors from FastText-German. Embeddings were reduced from 300-dimensional vectors. Color coding corresponds to different parts-of-speech (POS) tags

- The top frequent words that occur the least often in the other corpus, for all word lengths, and for words with more than 10 characters.

childLex	F	childLex >10	F	ChLT	F	ChLT >10	F
daß	6.1	Olchi-Kinder	3.5	Max	8.3	nahegelegenen	5.0
1	5.1	Wohnungstür	3.1	Mia	7.5	Schulvampire	4.5
jedenfalls	4.8	einigermaßen	3.0	Lina	7.0	unvergessliche	4.2
guckte	4.3	Viertelstunde	3.0	Lena	6.7	Sternenfohlen	4.1
mußte	4.2	Mottenflügel	2.9	Emma	6.6	einzustehen	4.1
Jedenfalls	4.2	stirnrunzelnd	2.9	Tim	6.5	Inselschüler	4.0
kriegt	4.2	entgeistert	2.8	Felix	6.4	Schafgääång	3.9
muß	4.1	Wolkenpfote	2.8	Paul	6.3	KuchenMonster	3.9
0	4.1	Unglaublich	2.8	Lilli	6.2	SkaterBande	3.7
wär	4.1	ausnahmsweise	2.8	Finn	6.1	abenteuerlustiger	3.5

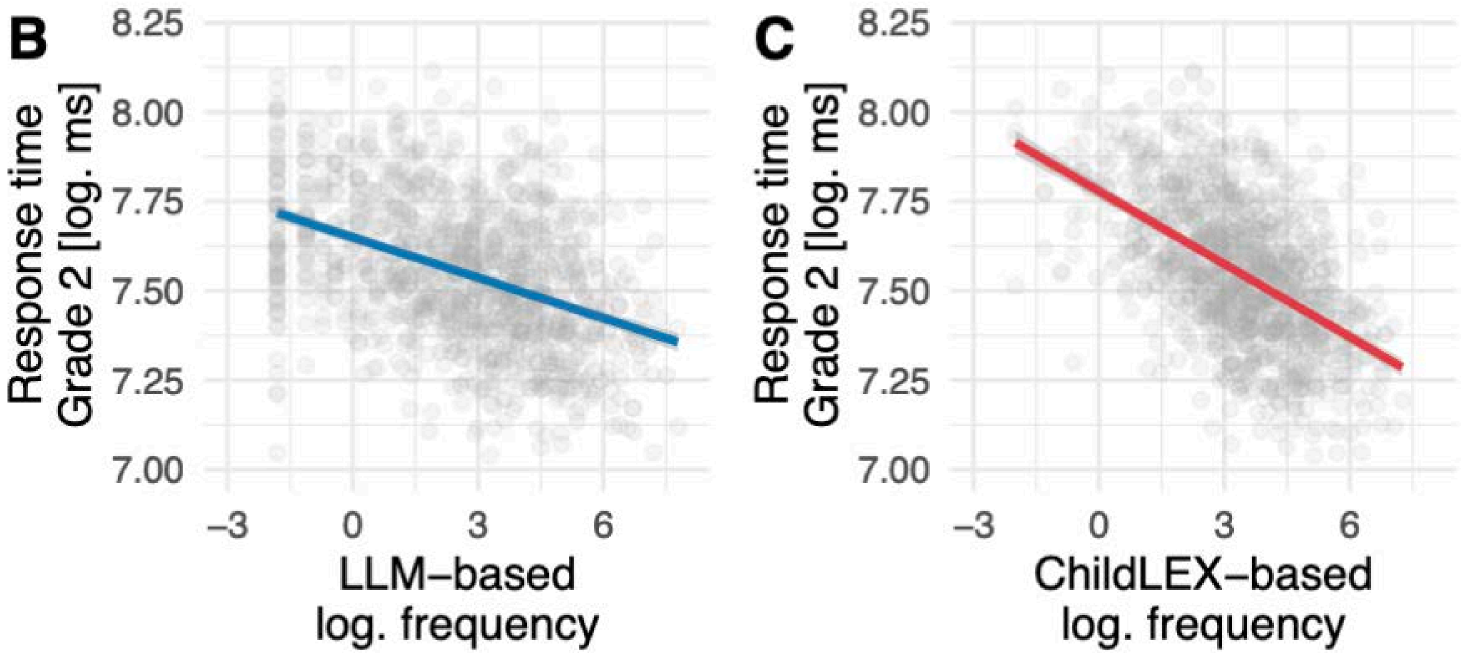
Setup

- Log-transformed normalized word frequency and log-transformed child RTs $\log\left(\frac{(1+\text{frequency})\times 10^6}{\text{corpus\_size}}\right)$



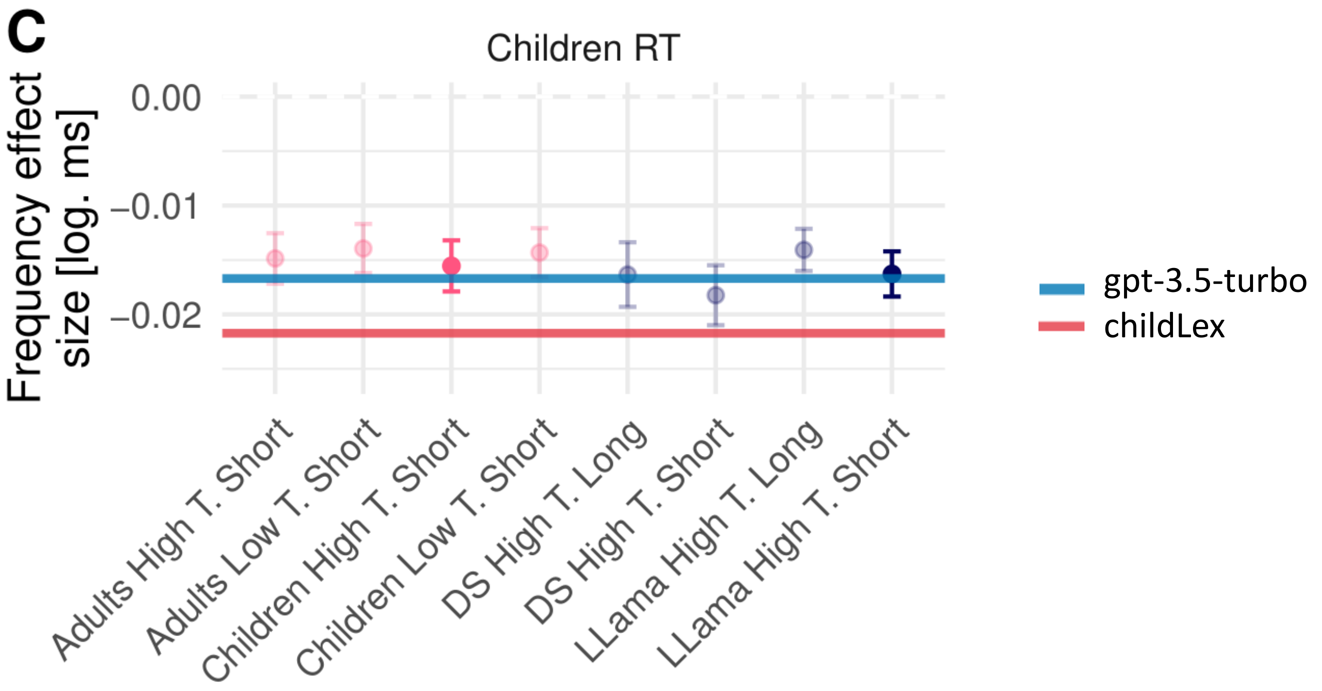
Effect size comparison (childLex vs. GPT-3.5)

RT ~ old20 + aoa + letter.cnt + unigram
+ bigram + trigram + Word Frequency



Effect size is lower

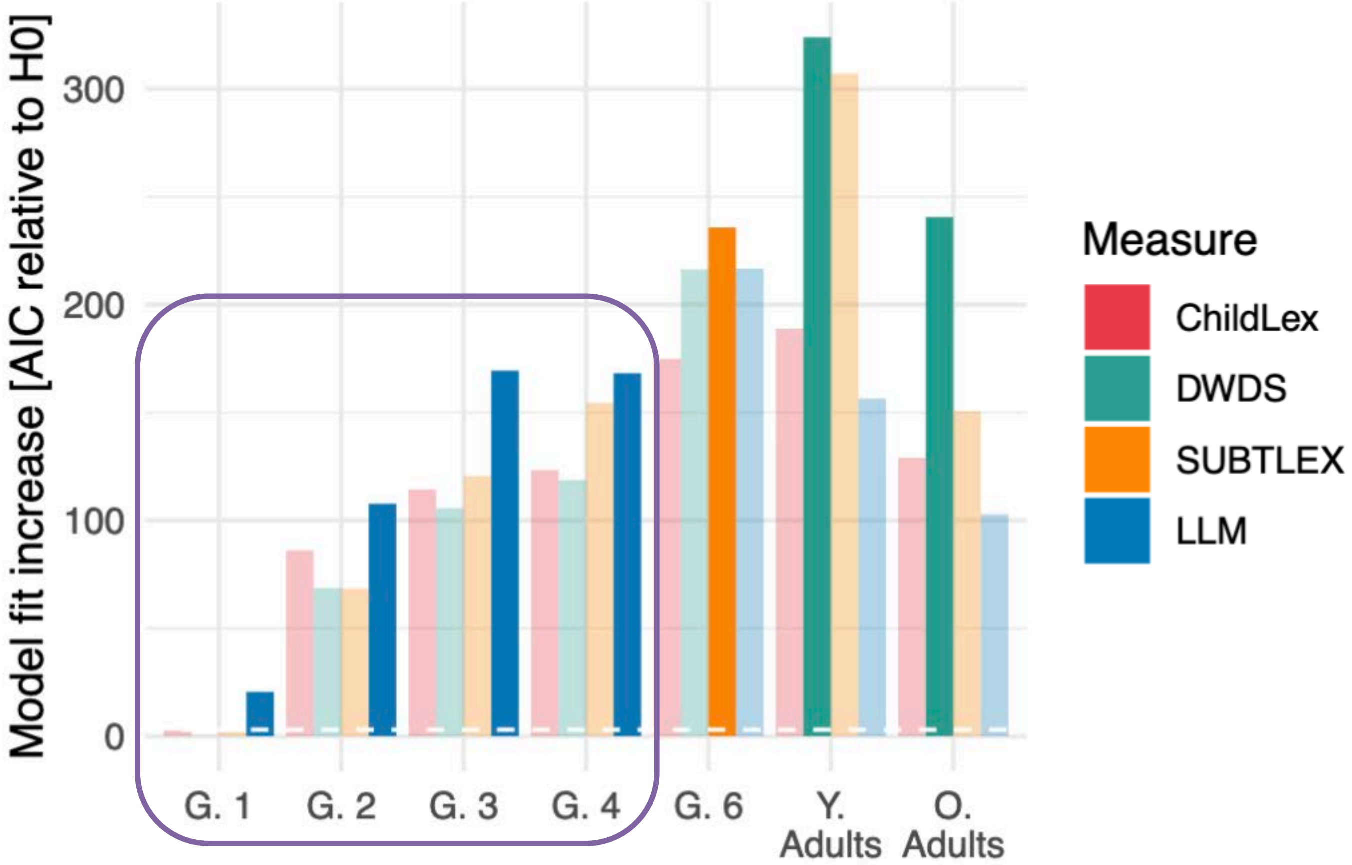
Effect size comparison (all corpora)



Effect sizes are comparable across models

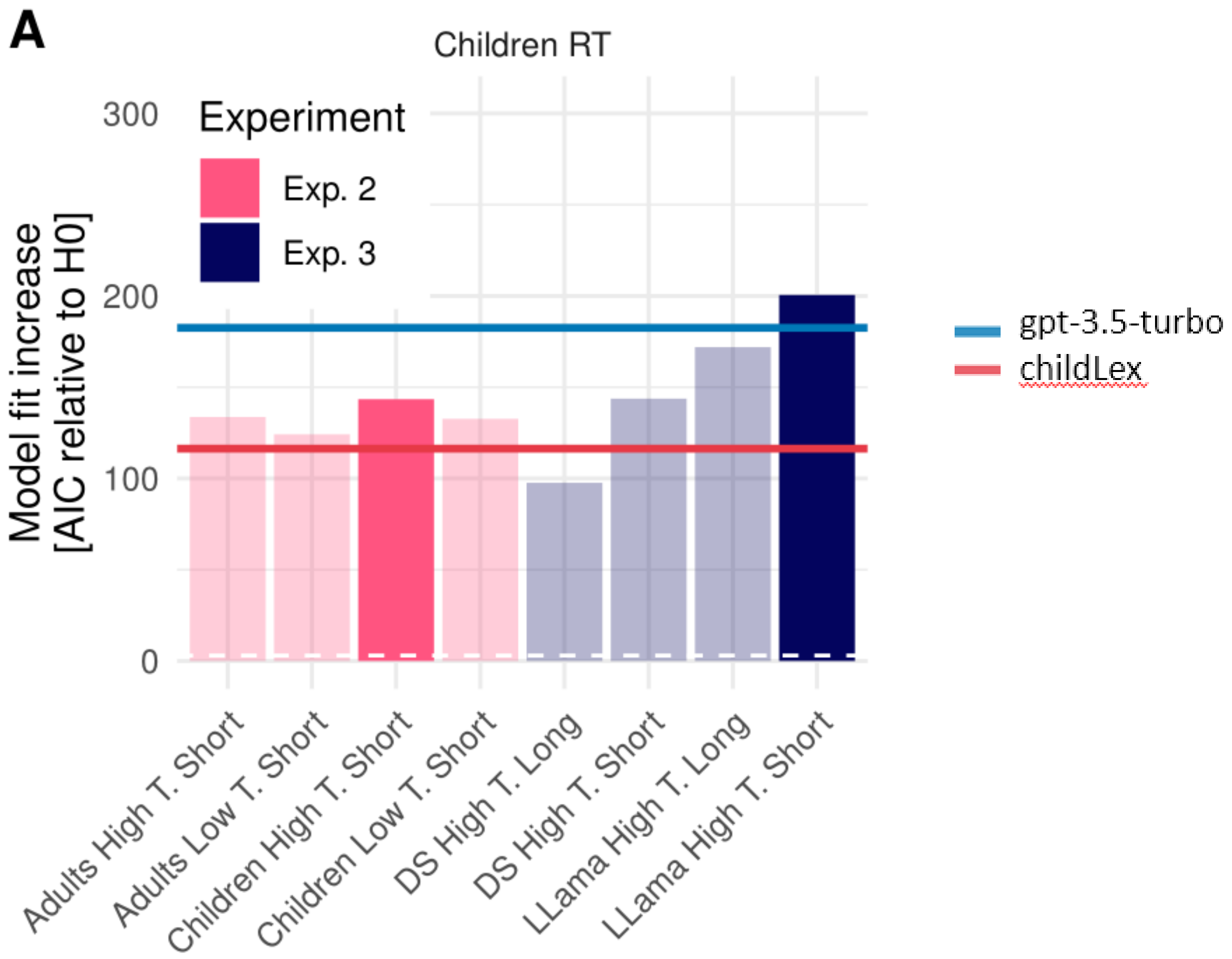
Model fit comparison (childLex vs. GPT-3.5)

RT ~ old20 + aoa + letter.cnt + word.frequency



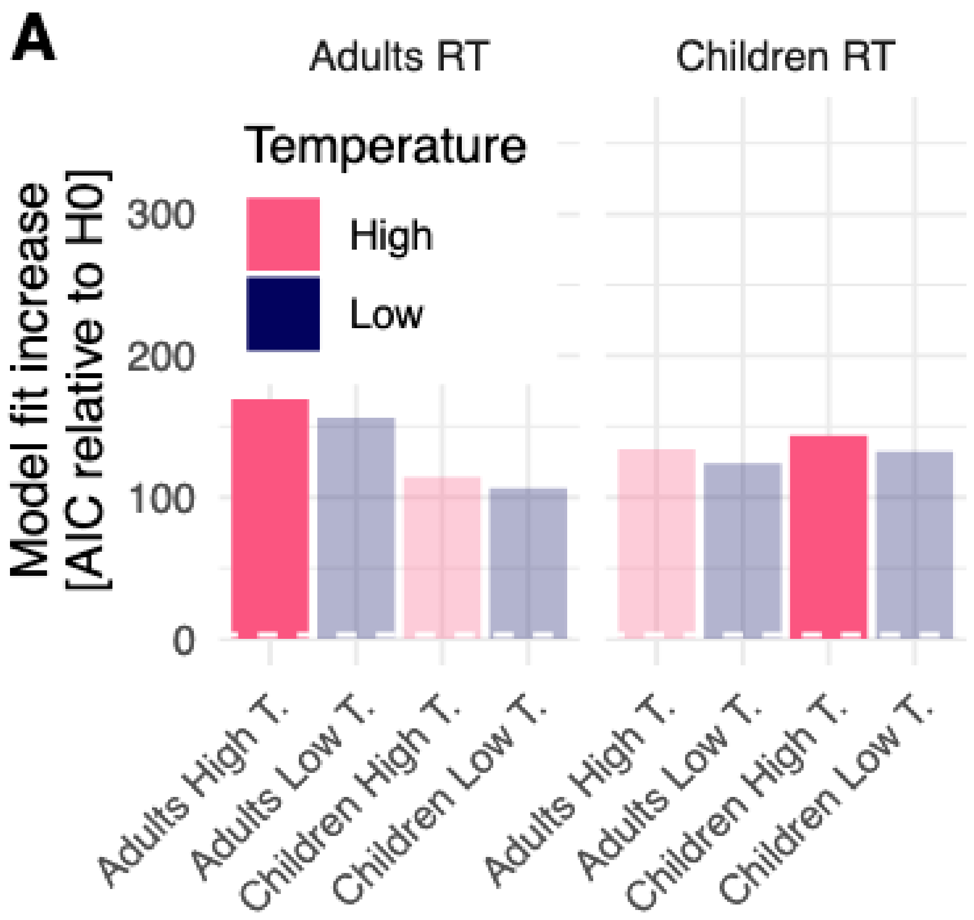
Model fit is higher

All corpora: Estimation of the word frequency effect



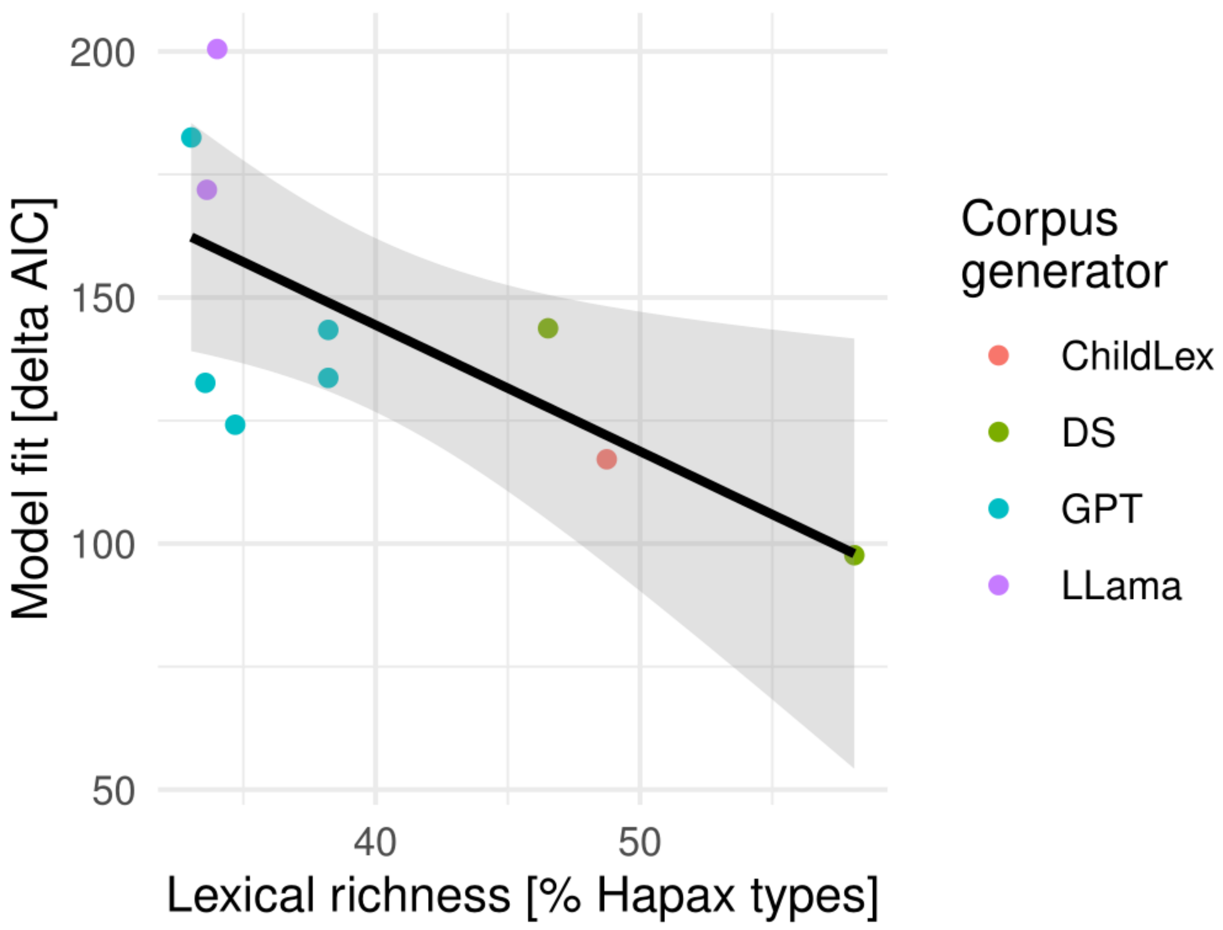
Highest model fit least lexically rich model

All corpora: Estimation of the word frequency effect



Highest model fit for age specific LLM-based frequency

Inverse scaling effect (all corpora)



Less rich → better fit

Summary: Using LLMs for building linguistic corpora

- High **correlation** with childLex word frequency, despite lower richness
- Better **model fit**, but smaller effect size
- Temperature & target audience: as expected
- **Inverse scaling**: Less richness results in better model fit

- *Better* representation of word frequency than authors of kids' books?
- Surprising differences in language use

Find Preprint here

Can large language models generate useful linguistic corpora?
A case study of the word frequency effect in young German readers

Job Schepens - Institute for Digital Humanities

Hanna Wołoszyn - Self learning systems lab

Nicole Marx - Mercator Institute for Literacy and Language Education

Benjamin Gagl - Self learning systems lab

SSLAC

Skills and Structures in Language and Cognition

Self Learning Systems Lab

Practice Session

- Task: generate a corpus, estimate word frequencies, fit to reading times, compare to other measures
- Github repo already consists of code and data to do this for English word reading times
  – https://github.com/jobschepens/mlschool-text
- The goal is to try out the pipeline, explore, run new experiments, and possibly extend the analysis
- You can run the code locally (recommended) or online
- I recommend using an Open Router API key to access cheap and fast models such as qwen-30b