Current developments in NLP using LLMs

Job Schepens & Nils Reiter
job.schepens@uni-koeln.de
nils.reiter@uni-koeln.de

Contents

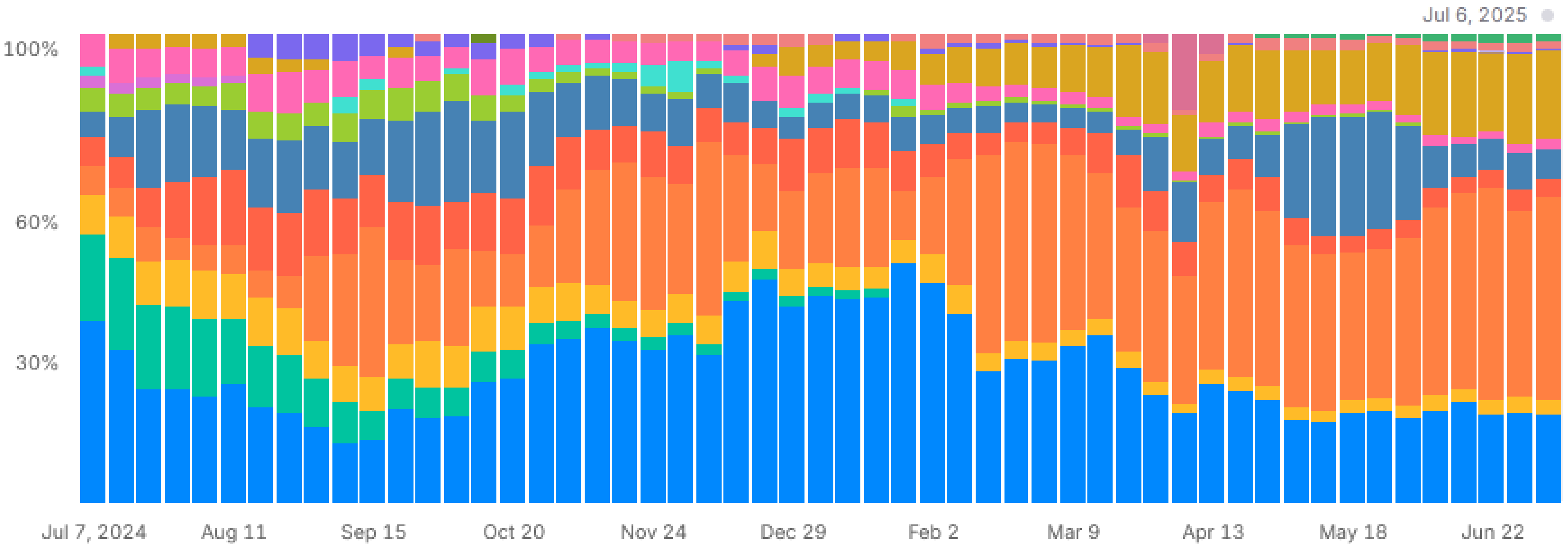
1. Current LLMs
2. Using LLMs for experimental stimulus generation
   – Toy example using vibe coding
3. Using LLMs for corpus building
   – Schepens, Wołoszyn, Marx, & Gagl (resubmitted).
4. Nils: three more developments

15.00-16.00: Discussion on use cases and current research projects

- Five minute presentations by: Nils Reiter, Job Schepens, Mark Ellison, Ziyue Liu, Philip Georgis

Market Share

Compare OpenRouter token share by model author

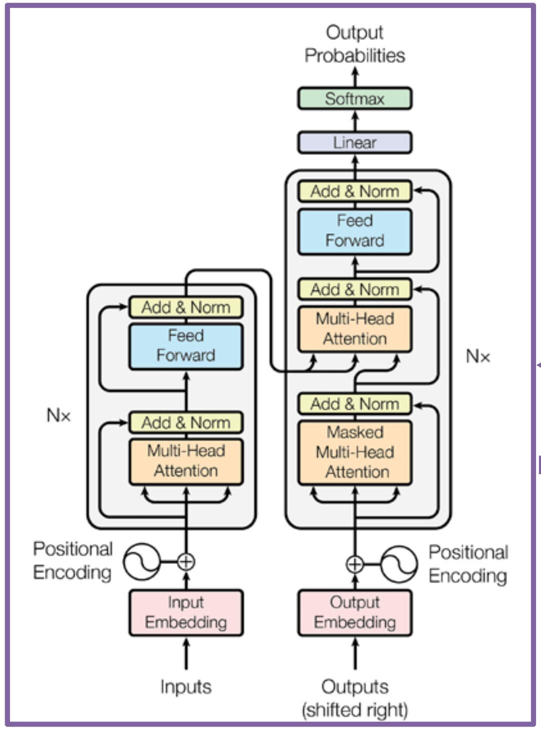


1.	google	43.4%	6.	mistralai	3.3%
2.	deepseek	18.9%	7.	others	2.1%
3.	anthropic	18.7%	8.	qwen	1.6%
4.	openai	6.4%	9.	x-ai	1.5%
5.	meta-llama	3.7%	10.	nousresearch	0.4%

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, etc. (focus of today's talk)
  - As (robust?) **models** (of what?): e.g. next word prediction, cloze probability
- But also many responsibilities (not the focus of today)
  - Ethics, code of conduct, standards, best practices (transparency, human verification, etc.)
- LLMs in the SFB1252
  - Research Data & Methods, Brown Bag Lunches, Reproducibilitea, DH Colloquium, UzK Data Steward Network, Informal discussion groups, etc.
  - Ongoing research: come to the discussion round

Large Language Models (LLMs)



https://medium.com/@sharanharsoor

Interface

Prompt

Model-generated Text

ChatGPT

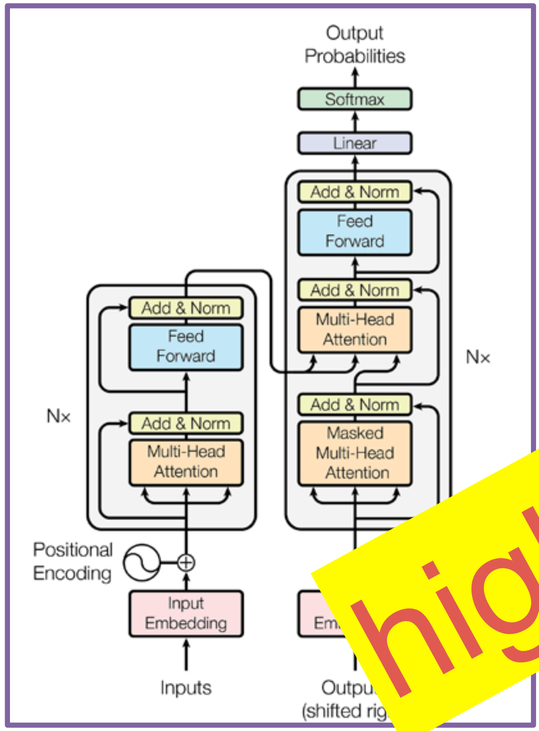
Large Language Models (LLMs)

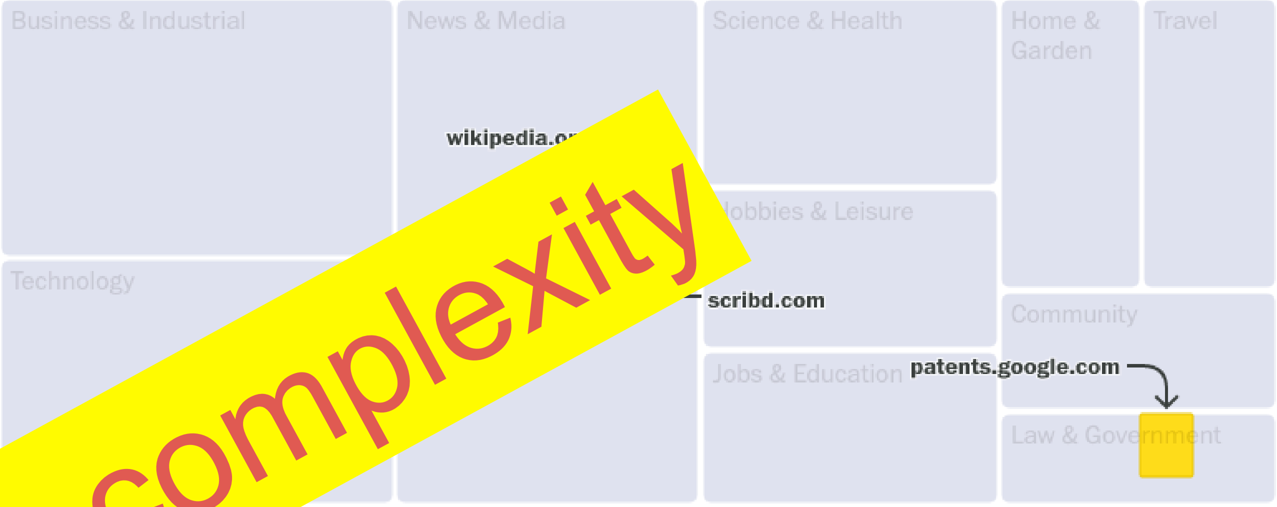
GPT-3, 3.5, 4

175 billion parameters

~10 trillion tokens in training data?

~ 5,000 years reading time for a normal reader per one trillion tokens





high complexity

w.washingtonpost.com/technology/interactive/2023/ai-chatbot-learning/

RANK	DOMAIN	CATEGORY	PERCENT OF ALL TOKENS
1	patents.google.com	Law & Government	0.46%
2	wikipedia.org	News & Media	0.19%

Project	Availability						Documentation					Access methods	
(maker, bases, URL)	Open code	LLM data	LLM weights	RLHF data	RLHF weights	License	Code	Architecture	Preprint	Paper	Data sheet	Package	API
chatGPT	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	~
OpenAI	LLM base: GPT3.5, GPT4			RLHF base: Instruct-GPT								https://chat.openai.com	

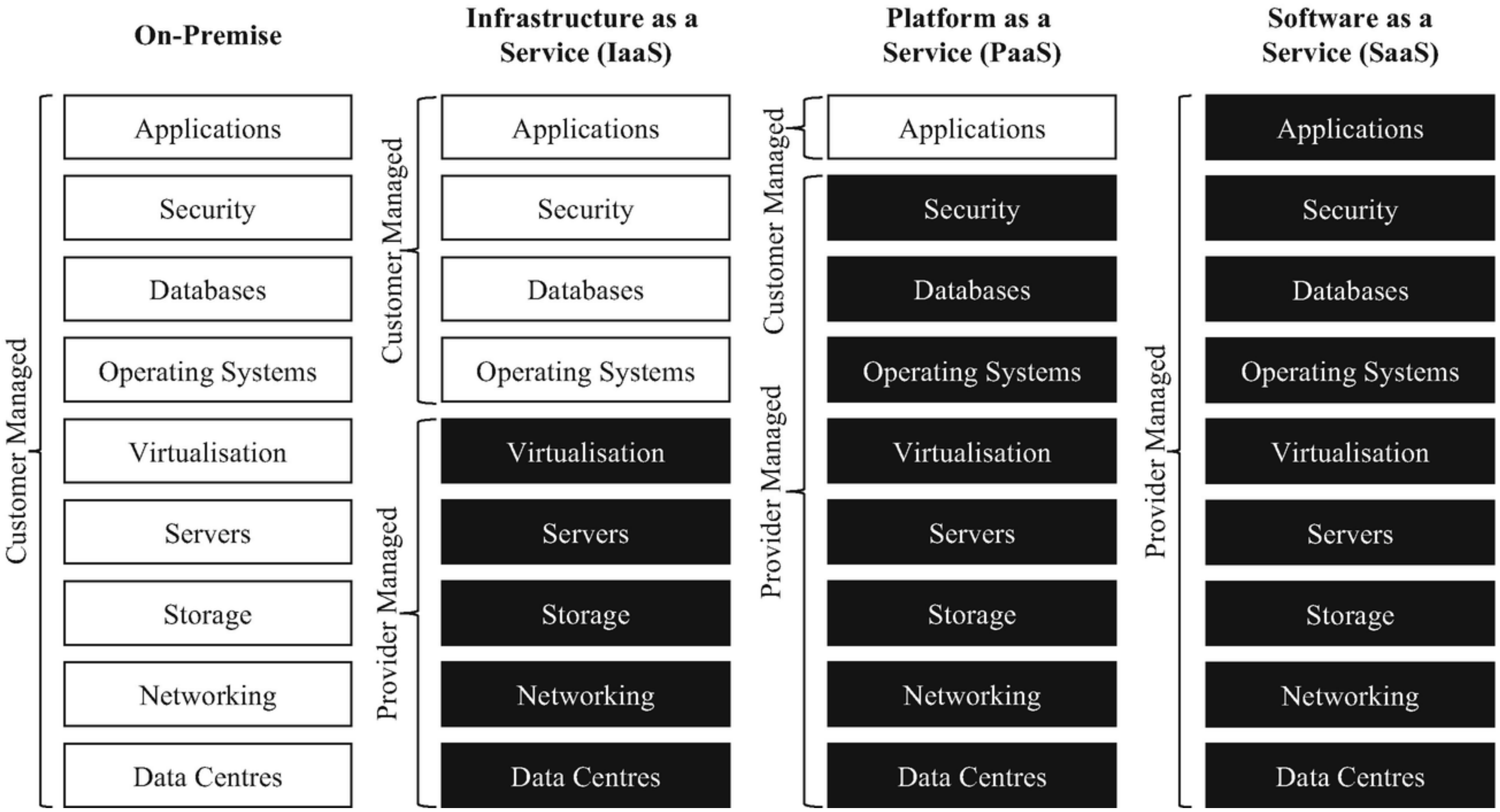
Liesenfeld, Lopez, & Mark Dingemanse (2023) Opening up ChatGPT

Ollama

Google Colab

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

Slide created with help of 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")

```
# 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']
```

```
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_to
        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

Code generated using Claude Sonnet 4
Sentences generated using GPT-2

Interim Summary: LLMs for stimulus generation

Strengths

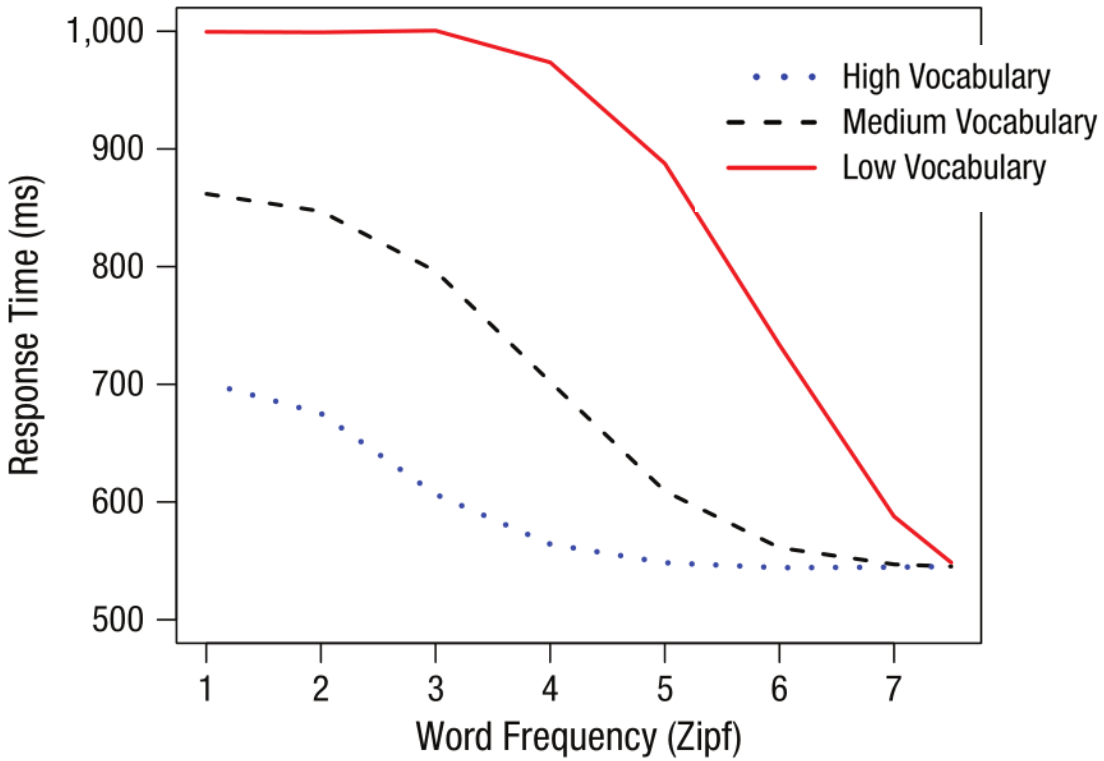
- **Scalability**: Generate many stimuli
- **Consistency and transparency**: Formalized criteria
- **Flexibility**: Easy to regenerate with different criteria
- **Reproducibility**: Documented and version controlled implementation

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

GPT, DeepSeek, LLama

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)

“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.

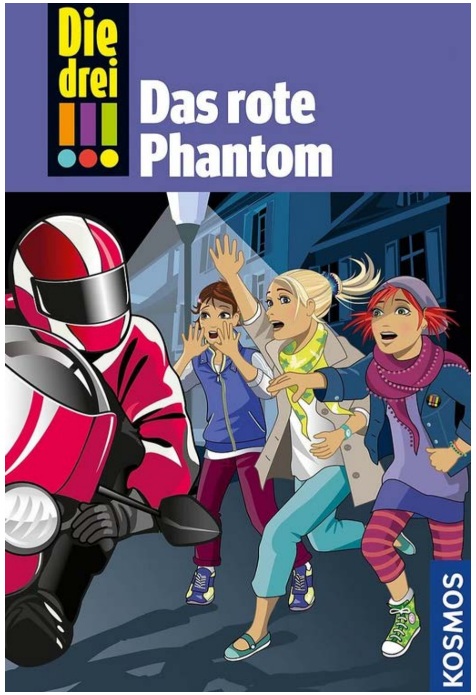
Eines Tages, als Max im Wald spielte, hörte er ein seltsames Geräusch. Es war ein leises Knurren, das immer lauter wurde. Max drehte sich um und sah ein rotes Phantom auf ihn zukommen. Es hatte riesige Augen und scharfe Krallen.

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.

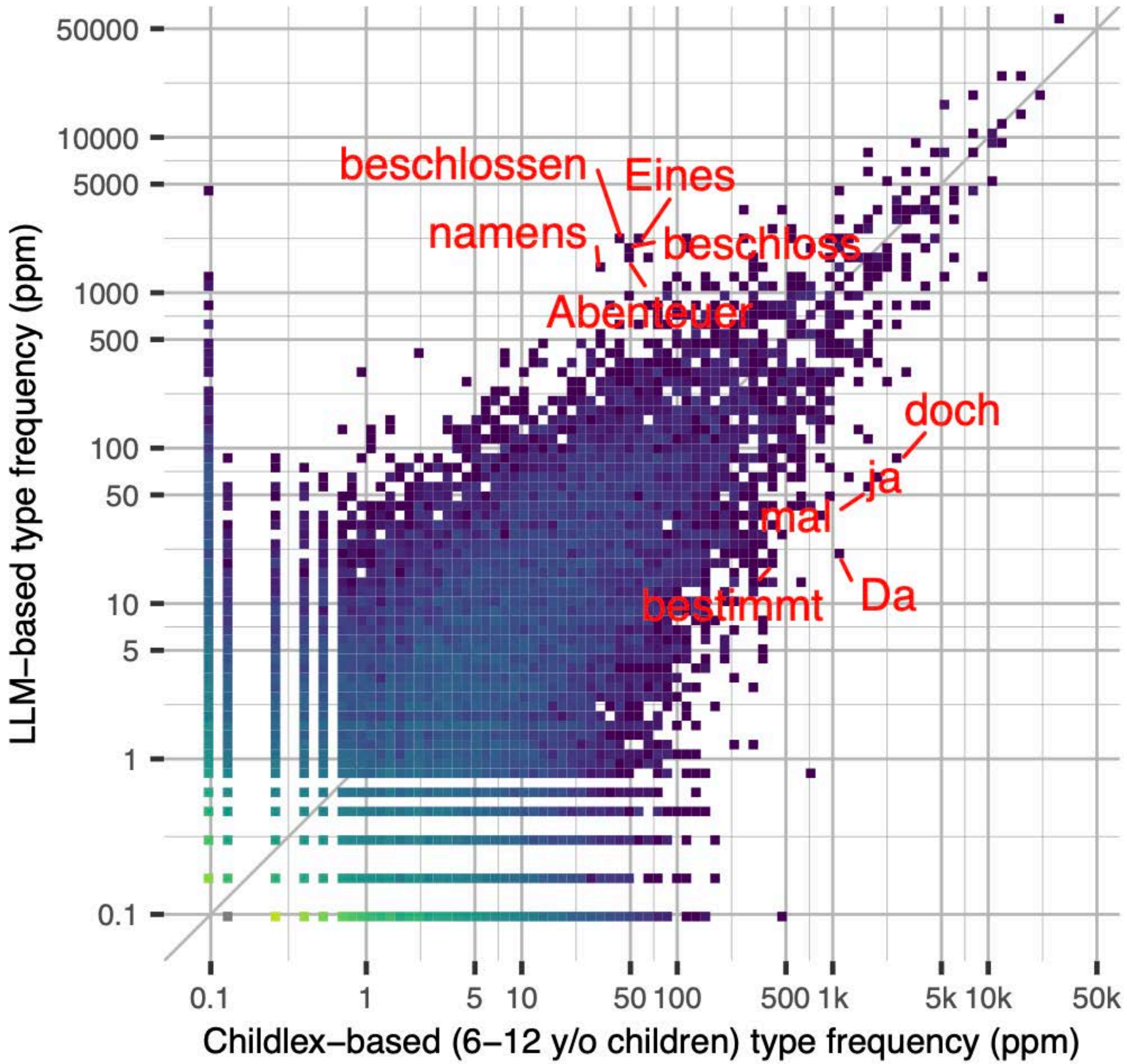
Klappentext

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

Word frequency comparison (childLex vs. GPT 3.5)

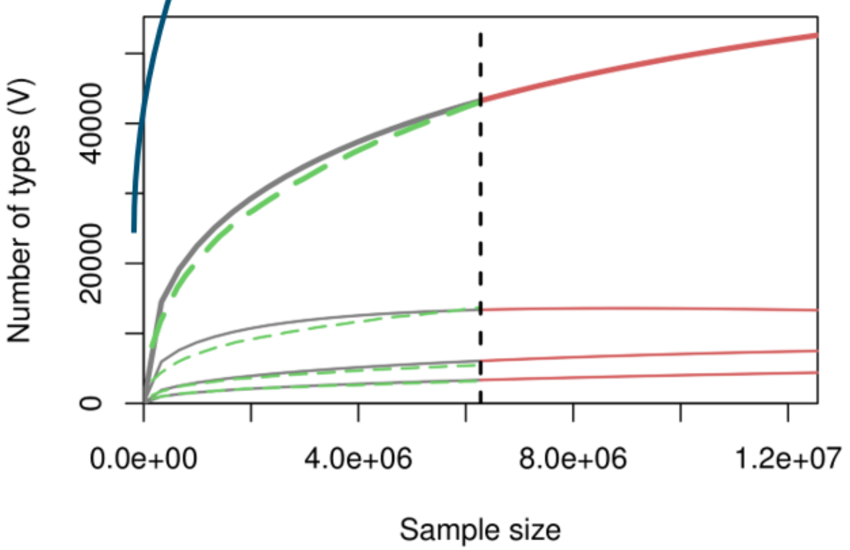


r = .88

Lexical richness comparison (childLex vs. GPT-3.5)

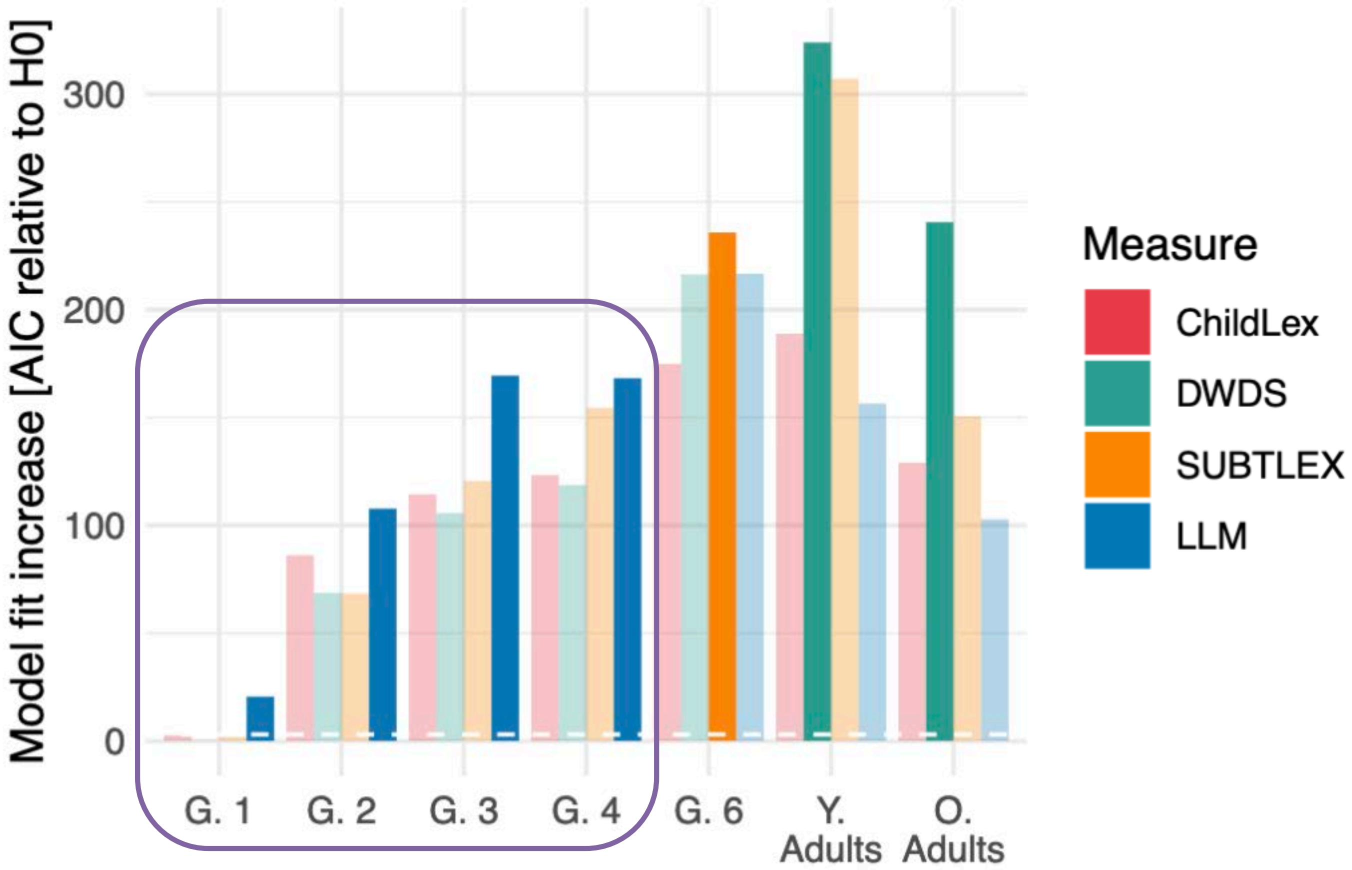
Measure	childLex	LLM-corpus
n Books	500	500
Tokens	9,850,786	6,252,808
Types	182,454	46,409
Lemmas	117,952	34,519

Low lexical richness of LLM corpus



Model fit comparison (childLex vs. GPT-3.5)

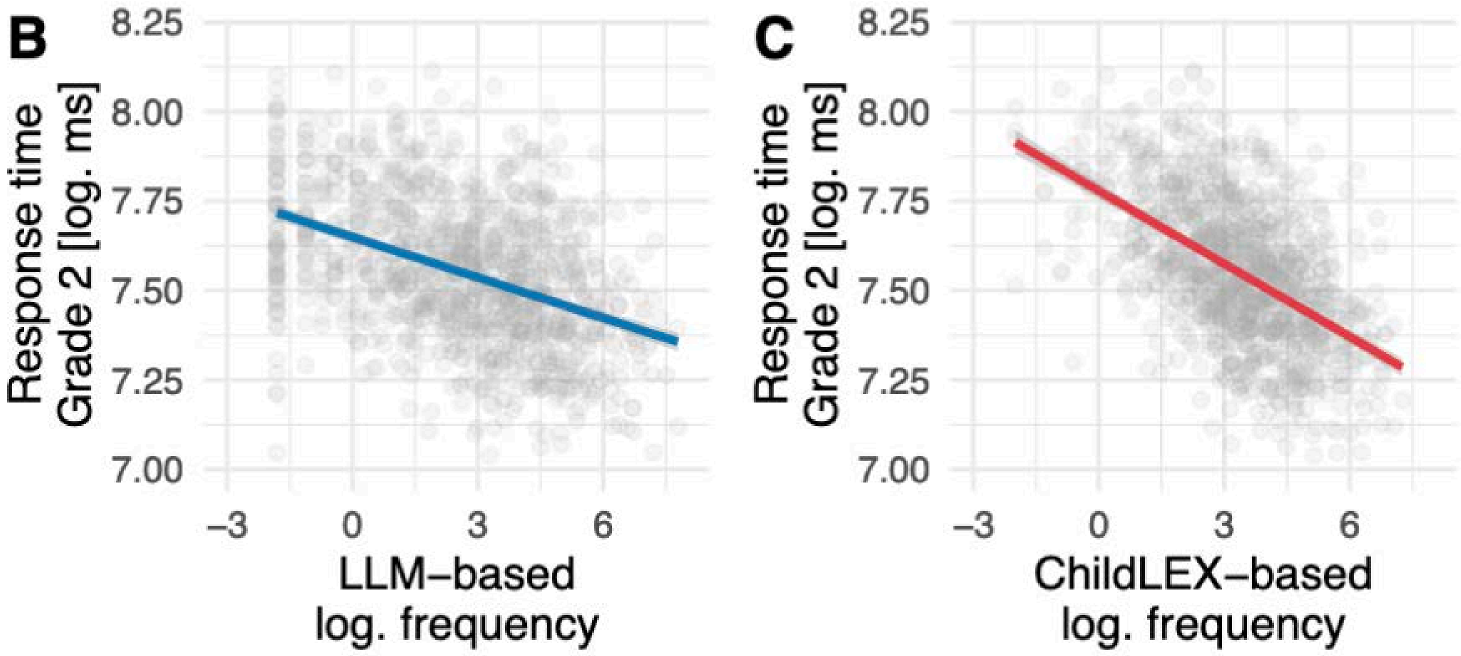
RT ~ old20 + aoa + letter.cnt + word.frequency



Model fit is higher

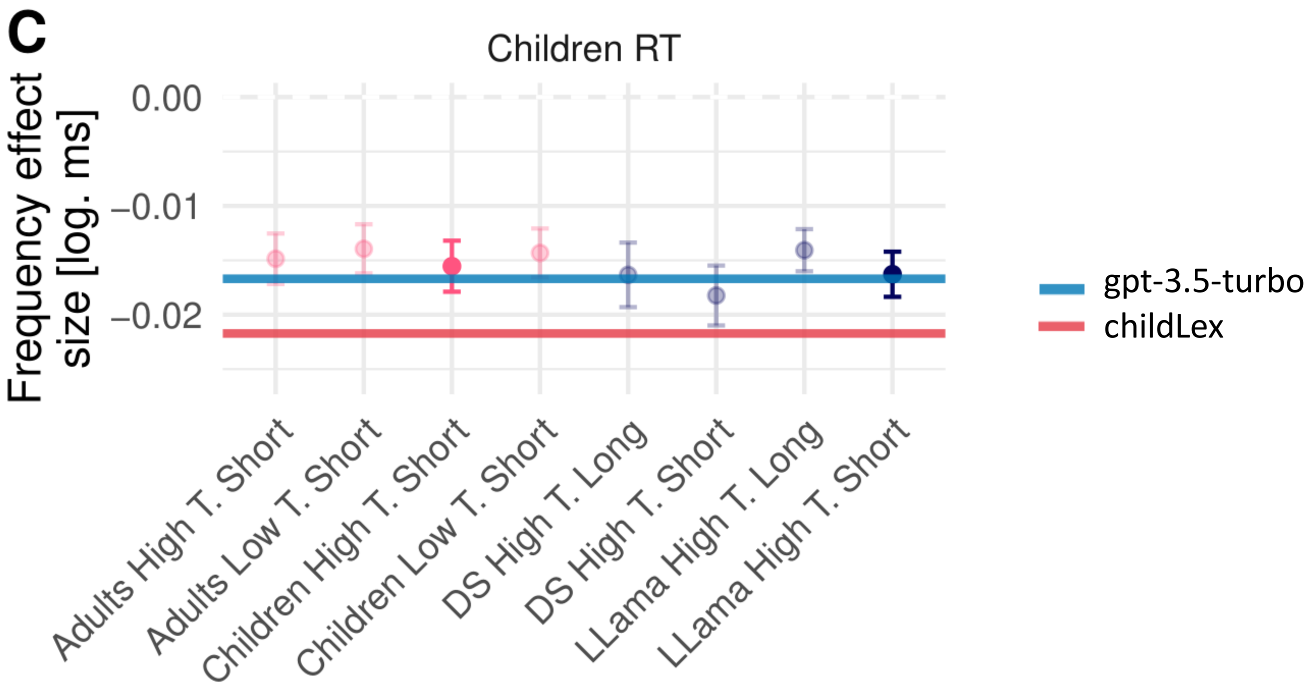
Effect size comparison (childLex vs. GPT-3.5)

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



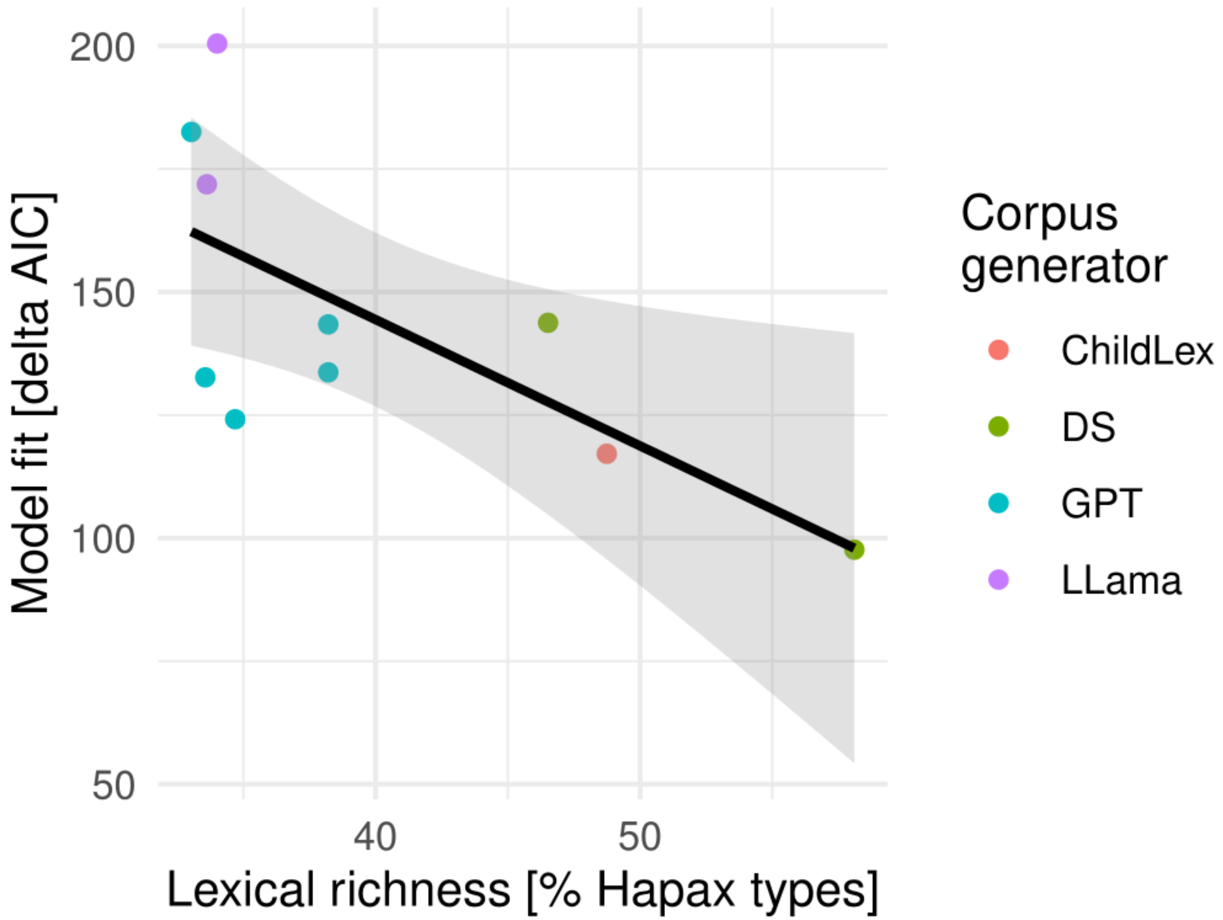
Effect size is lower

Effect size comparison (all corpora)



Effect sizes are comparable across models

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

Section 1

What's New in LLM-Land?

LLM Developments

- LLM development continues and is fast
- By now: Inseparable from general AI developments
- Heavily influenced by industry – not a purely academic thing
- Relevant for our research – but also general lives
  - E.g., student theses, examination, social media, ...

LLM Developments

- LLM development continues and is fast
- By now: Inseparable from general AI developments
- Heavily influenced by industry – not a purely academic thing
- Relevant for our research – but also general lives
  - E.g., student theses, examination, social media, ...

Three Developments of Interest (subjective choice)

1. Beyond Language: Tools, MCP, RAG & co
2. Fragility
3. Reasoning

❶ LLMs and Tools

- Classical, most well known use case: Text as in- and output (“chatbot”)
- Massive development in past year: LLMs call tools
  - Trained to (also) produce machine-readable output
    - E.g. `count-words("some text")`
  - Execution environment detects and handles function call
  - Results passed back to the model

❶ LLMs and Tools

- Classical, most well known use case: Text as in- and output ("chatbot")
- Massive development in past year: LLMs call tools
  - Trained to (also) produce machine-readable output
    - E.g. `count-words("some text")`
  - Execution environment detects and handles function call
  - Results passed back to the model
- Why is this important?
  - Because models are bad at certain tasks

```
>>> How many "r" are in strawberry?
There are 2 "R"s in the word "strawberry".
```

1 LLMs and Tools

- Classical, most well known use case: Text as in- and output ("chatbot")
- Massive development in past year: LLMs call tools
  - Trained to (also) produce machine-readable output
    - E.g. `count-words("some text")`
  - Execution environment detects and handles function call
  - Results passed back to the model
- Why is this important?
  - Because models are bad at certain tasks
- Model Context Protocol
  - Standardized interface between ('all') LLMs and tools

```
>>> How many "r" are in strawberry?
There are 2 "R"s in the word "strawberry".
```

❶ LLMs and Tools

Example Applications and Current State

- Dataset analysis for stimuli generation
  - Tools can provide quantitative analysis over a data set
  - LLM takes it into account
  - E.g.: "Generate 20 stimuli in the same pattern as … Use nouns with similar frequency."

❶ LLMs and Tools

Example Applications and Current State

- Dataset analysis for stimuli generation
  - Tools can provide quantitative analysis over a data set
  - LLM takes it into account
  - E.g.: "Generate 20 stimuli in the same pattern as … Use nouns with similar frequency."
- LLM ask questions back – as part of their 'reasoning' process
  - Model identifies key decision (e.g., what exactly is meant by a word)
  - Key decision made by human, model continues with result
  - New possibilities for "human in the loop"-approaches

❶ LLMs and Tools

Example Applications and Current State

- Dataset analysis for stimuli generation
  - Tools can provide quantitative analysis over a data set
  - LLM takes it into account
  - E.g.: "Generate 20 stimuli in the same pattern as … Use nouns with similar frequency."
- LLM ask questions back – as part of their 'reasoning' process
  - Model identifies key decision (e.g., what exactly is meant by a word)
  - Key decision made by human, model continues with result
  - New possibilities for "human in the loop"-approaches
- Current state
  - MCP specification proposed by Anthropic in Nov. 2024
  - Supported/adopted by major LLM developers, including OpenAI
  - Not end-user ready (open questions around security)

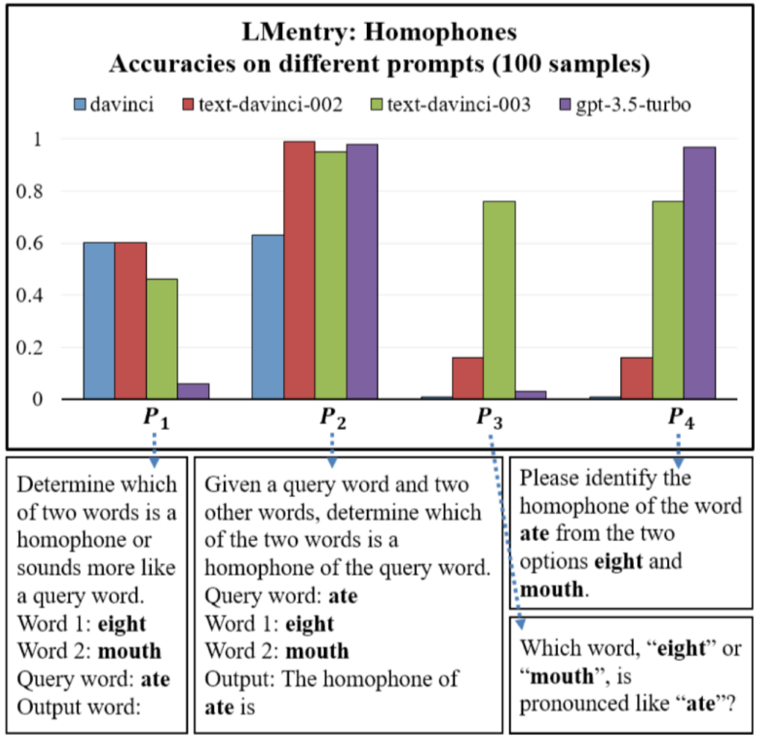
❷ Fragility

- Many aspects around LLMs are surprisingly fragile

❷ Fragility

- Many aspects around LLMs are surprisingly fragile

"Prompt brittleness" Mizrahi et al. (2024)



- Changing insignificant details in a prompt can lead to drastically different outcomes
- LLM evaluation/comparison always needs to look at multiple prompts
- This makes LLM evaluation more costly

2 Fragility

Super weights

Yu et al. (2025)

- Llama2 7B has 7B parameters ("7 Milliarden") – removing one parameter halves performance

Llama-7B	**Arc-c**	**Arc-e**	**Hella.**	**Lamb.**	**PIQA**	**SciQ**	**Wino.**	**AVG**
Original	41.81	75.29	56.93	73.51	78.67	94.60	70.01	70.11
Prune SW	19.80	39.60	30.68	0.52	59.90	39.40	56.12	35.14
Prune Non-SW	41.47	74.83	56.35	69.88	78.51	94.40	69.14	69.22
Prune SW, +SA	26.60	54.63	56.93	12.79	67.95	61.70	70.01	50.09

- Practical relevance for model compression
- Also shows that we only partially understand what's going on there

❸ Reasoning

- More models optimized for “reasoning”
- Reasoning: Document and express intermediate steps
- Presumably higher level of transparency

3 Reasoning

- More models optimized for "reasoning"
- Reasoning: Document and express intermediate steps
- Presumably higher level of transparency
- First attempts
  - "Chain-of-thought prompting": Tell the model that it should think step by step
- Since 2024
  - Specific models for reasoning
  - RLM: Reasoning language models

❸ Reasoning

Reasoning Language Models (RLMs)

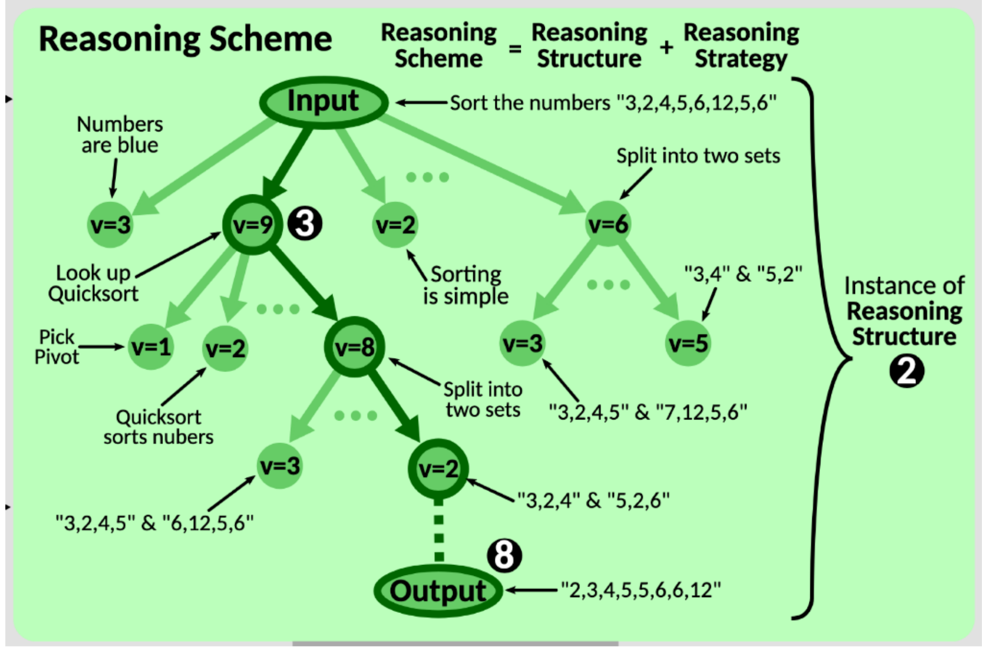
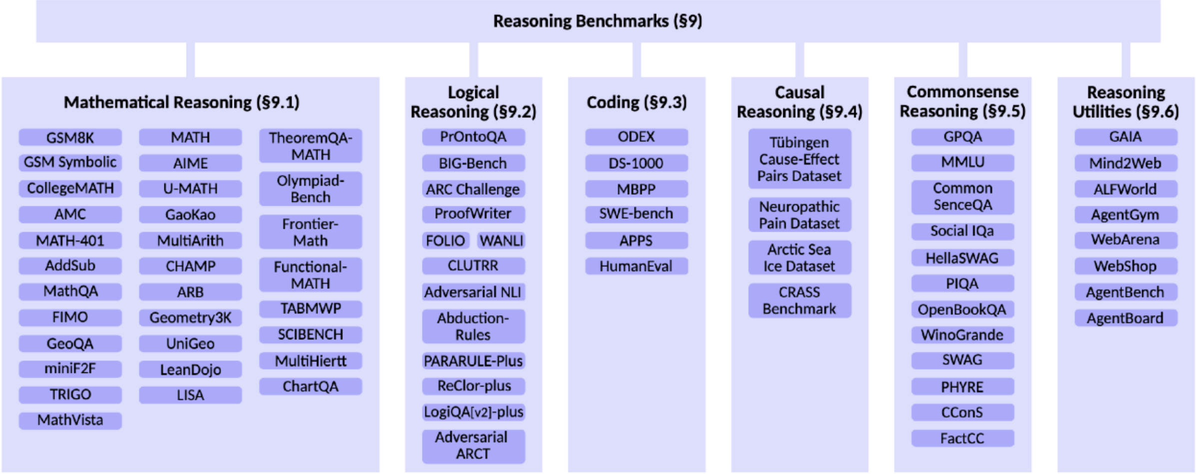


Figure: Besta et al. (2025)

- Computationally very expensive
- Multiple models working in conjunction
  - "Policy model": Generate new reasoning steps
  - "Value model": Evaluate quality of reasoning paths
- Training data: Mostly automatically generated
- Benchmarks from various domains; math in majority



References I

Besta, Maciej/Julia Barth/Eric Schreiber/Ales Kubicek/Afonso Catarino/Robert Gerstenberger/Piotr Nyczyk/Patrick Iff/Yueling Li/Sam Houliston/Tomasz Sternal/Marcin Copik/Grzegorz Kwaśniewski/Jürgen Müller/Łukasz Flis/Hannes Eberhard/Zixuan Chen/Hubert Niewiadomski/Torsten Hoefler (2025). *Reasoning Language Models: A Blueprint*. DOI: 10.48550/arXiv.2501.11223. arXiv: 2501.11223[cs]. URL: http://arxiv.org/abs/2501.11223 (visited on 07/09/2025).

Mizrahi, Moran/Guy Kaplan/Dan Malkin/Rotem Dror/Dafna Shahaf/Gabriel Stanovsky (2024). *State of What Art? A Call for Multi-Prompt LLM Evaluation*. _eprint: 2401.00595. URL: https://arxiv.org/abs/2401.00595.

Yu, Mengxia/De Wang/Qi Shan/Colorado J. Reed/Alvin Wan (2025). *The Super Weight in Large Language Models*. DOI: 10.48550/arXiv.2411.07191. arXiv: 2411.07191[cs]. URL: http://arxiv.org/abs/2411.07191 (visited on 07/08/2025).