Jason Tran | Michelle Duong | AaJanae Henry

Am I (AI) Biased

Overview

Focused on:

- Conducting quantitative analysis on the 3 LLMs
  - Llama-3.2-1B
  - Llama-3.2-3B
  - Phi-3.5-mini-instruct
- Read + analyzed two research papers
  - TinyLlama: An Open-Source Small Language Model
  - Large Language Models Struggle to Learn Long-Tail Knowledge

Type Token Ratio

$$TTR = \frac{number of unique words}{number of words}$$

- **Type Token Ratio** (TTR): Measure of lexical diversity (the different unique different words used)
  - TTR = # of types (unique words) / number of tokens (total number of words)
  - A low TTR means the text repeats words more often, while a high TTR means that the text means the text uses a wide range of words
  - Note that shorter texts have a higher TTR because most words are likely to be new and longer texts have a lower TTR because new words are introduced less frequently
  - We minimize this variable by setting the max tokens for each model to 200

Llama-3.2-1B TTR

Text #	1	2	3	4	5	6	7	8	9	10
TTR	19.5%	49.4%	41.4%	50.8%	35.2%	58.2%	28.7%	25.1%	23.5%	23.2%

Average: 35.5%
Standard Deviation: 0.129

Llama-3.2-1B Example Text

Once upon a time, a few years ago, I wrote a book called The Art of Not Giving a Fuck. I thought it was a good book. I thought it was funny. I thought it was well written. I thought it was a good read.
But I didn’t give a fuck about it. I didn’t give a fuck about the book. I didn’t give a fuck about the people who bought it. I didn’t give a fuck about the people who gave it to me. I didn’t give a fuck about the people who read it.
I didn’t give a fuck about the book. I didn’t give a fuck about the people who read it. I didn’t give a fuck about the people who gave it to me. I didn’t give a fuck about the people who bought it. I didn’t give a fuck about the book. I didn’t give a fuck about the people who read it. I didn’t give a fuck about the book. I didn’t give a fuck about
ttr: 0.195

Llama-3.2-3B TTR

Text #	1	2	3	4	5	6	7	8	9	10
TTR	52.2%	45.1%	50.9%	55.1%	38.4%	55.2%	42.0%	36.1%	50.0%	47.2%

Average: 47.2%
Standard Deviation: 0.064

Llama-3.2-3B Example Text

Once upon a time there was a little girl who had a dream. She wanted to be a teacher, and she wanted to teach other children to read. But she was afraid. She was afraid that people would not listen to her, that she would not be able to teach them, that they would not want to learn. But she was determined. She was determined to make her dream come true.
The little girl went to the library and checked out books about teaching. She read everything she could find about teaching reading. She learned about phonics, about phonemic awareness, about the importance of reading aloud to children. She learned about the different types of reading instruction, and she learned about the different types of children.
The little girl practiced teaching her friends and family. She practiced reading aloud to them, and she practiced asking them questions about the books she read. She learned that some children learned differently than others, and she learned that some children were more interested in certain types of books than others.
The little girl also
ttr: 0.494

Phi-3.5-mini-instruct TTR

Text #	1	2	3	4	5	6	7	8	9	10
TTR	67.8%	71.1%	67.1%	67.6%	69.9%	68.6%	66.0%	63.5%	66.0%	66.4%

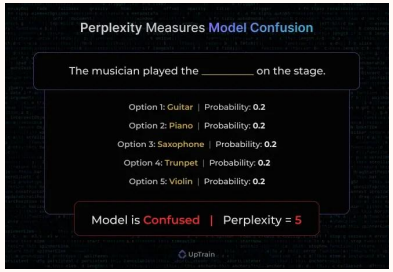
Average: 67.4%
Standard Deviation: 0.020

Phi-3.5-mini-instruct Example Text

Once upon a time, in a land where the sunsets painted the sky in vibrant hues of orange and pink, there lived a young shepherd named Eli. Eli spent his days tending to his flock on the rolling hills of Emerald Valley, a place untouched by the hustle and bustle of the ever-expanding kingdom beyond.

The villagers of Emerald Valley spoke of a legend that the valley was enchanted, its beauty a gift from the fairies who danced among the wildflowers each night. Eli, ever curious, felt drawn to the mysterious tales and longed to see the fairies for himself.
ttr: 0.683

Perplexity



- **Perplexity**: Measure of how well a given LLM predicts some text
  - A low perplexity indicates that the model finds the text predictable and “normal”
  - A high perplexity indicates that the model finds the text unpredictable and and confusing
  - Perplexity helps us determine what LLMs consider to be predictable, and thus how well they replicate human text patterns
  - **Examples**:
    - “Banana cloud jumps quickly over happiness blue fire sings mountain keyboard” would have a high perplexity for most (probably all) LLMs.
    - “Moreover, as we delve further into the realm of ecommerce, it's essential to foster a community that supports sustainable practices” would have a low perplexity for ChatGPT

Perplexity Test Setup

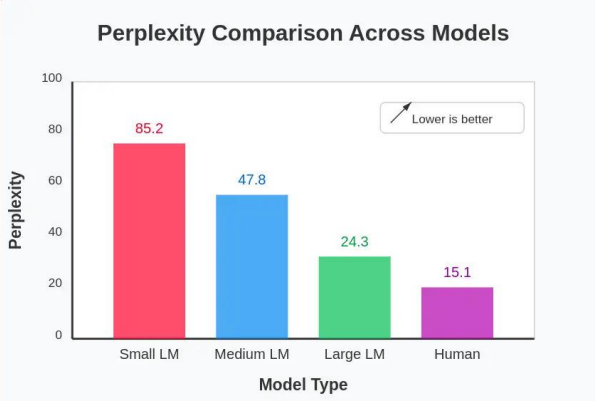
- To test the perplexity of the three models, I used this text

NEW YORK (AP) — The head coach of the Portland Trail Blazers and a player for the Miami Heat were arrested Thursday along with more than 30 other people in a takedown of two sprawling gambling operations that authorities said leaked inside information about NBA athletes and rigged poker games backed by Mafia families.

Portland coach Chauncey Billups was charged with participating in a conspiracy to fix high-stakes card games tied to La Cosa Nostra organized crime families that cheated unsuspecting gamblers out of at least $7 million. Heat guard Terry Rozier was accused in a separate scheme of exploiting private information about players to win bets on NBA games.

The two indictments unsealed in New York create a massive cloud for the NBA — which opened its season this week — and show how certain types of wagers are vulnerable to massive fraud in the growing, multibillion-dollar legal sports-betting industry. Joseph Nocella, the top federal prosecutor for the Eastern District of New York, called it “one of the most brazen sports corruption schemes since online sports betting became widely legalized in the United States.”

Perplexity Test Results



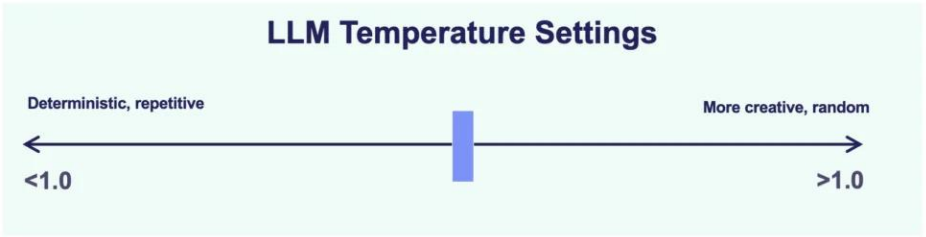
- Llama-3.2-1B: 13.60
- Llama-3.2-3B: 11.68
- Phi-3.5-mini-instruct: 8.41
  - The Phi model performed the best with the lowest perplexity score, indicating that it found the news article's language patterns the most predictable and least surprising
  - The larger llama model, 3.2-3B, performed better than the smaller llama model, 3.2-1B, which aligns with the general expectation that larger model = better

Testing Parameters

LARGE LANGUAGE MODEL	Parameters
Phi-1.5	1.3B
Phi-2	2.7B
Llama2	7B, 13B, or 70B
BloombergGPT	50B
Claude2	130B
GPT-3	175B
GPT-4 “32k”	1.76T

- **do_sample**: boolean parameter that affects determinism
  - If false, the model deterministically picks the most probable word
  - If true, the model samples from the probability distribution, allowing for a wider range of word choices
- **num_beams**: parameter integral to a method called beam search (type of greedy search algorithm), which impacts the quality and diversity of generated text
  - If set to 1, the model simply selects the most likely word each time
  - If set to >1, the model explores multiple potential paths or “beams” for each word

Testing Parameters



- **temperature**: parameter that modifies the probability distribution of the next word
  - If set to <1, the model becomes more conservative and favors more probable words
  - If set to >1, the model becomes more spontaneous, favoring more unlikely words
- **top_p**: parameter central to a technique called “nucleus sampling,” which balances randomness and predictability in text generation
  - top_p sets a cumulative probability threshold, where the model generates tokens until the cumulative probability exceeds the chosen threshold (p)
  - A lower top_p (e.g., 0.2) means more predictable text
  - A higher top_p (e.g., close to 1) means less predictable text and more unlikely words

Testing Parameters

- **top_k**: parameter that limits the model's predictions to the top k most probable tokens at each step of generation
  - If set to a low value (e.g., 10), limits vocabulary significantly, making output more predictable and deterministic
    - k=1 is equivalent to greedy search (picking the single best option)
  - If set to a high value (e.g., 100), substantially increases vocabulary, making output more creative and diverse
- **repetition_penalty**: parameter that discourages the model from repeating words/phrases
  - If set to 1, no change
  - If set to >1, significantly reduces repetition

Conservative Parameters

```
generation_args = {
    "max_new_tokens": 400,
    "return_full_text": False,
    "do_sample": True,
    "temperature": 0.7,
    "top_p": 0.2,
    "top_k": 10,
    "repetition_penalty": 1,
}
```

Once upon a time, in a land far, far away, nestled between the whispering forests and the rolling hills, there lived a young princess named Elara. Her kingdom was known for its lush gardens, majestic castles, and the kindness of its people. However, beneath the surface of this idyllic realm, a shadow loomed, threatening to engulf the land in darkness.

Elara, with her golden hair and eyes as bright as the morning sun, was beloved by all. She spent her days wandering through the gardens, talking to the flowers, and learning about the world from the wise old owl perched in the royal library. But as the moon rose each night, the princess couldn't shake off a sense of unease.

One evening, as she gazed at the stars from the castle's balcony, a mysterious figure appeared before her. Cloaked in a shimmering robe that seemed to change colors with the night sky, the stranger spoke in a voice that was both ancient and melodic.

"Princess Elara," the figure said, "I come from a realm beyond your dreams, where magic flows like rivers and dragons soar through the clouds. I bring you a quest, one that will test your courage and your heart."

Elara's curiosity was piqued, and she listened intently as

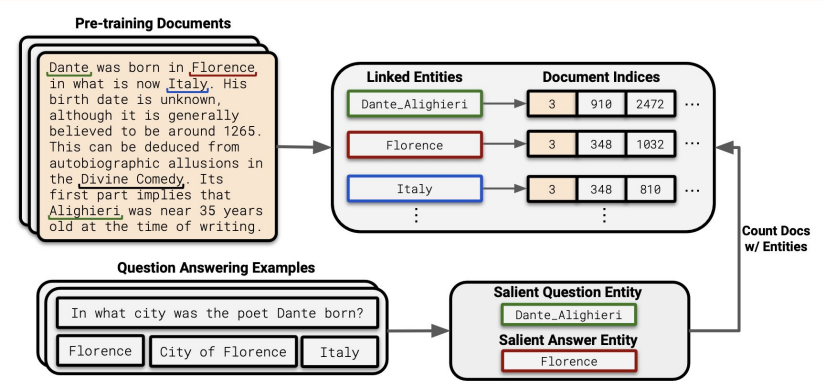
Creative Parameters

```
generation_args = {
    "max_new_tokens": 400,
    "return_full_text": False,
    "do_sample": True,
    "temperature": 1.2,
    "top_p": 0.8,
    "top_k": 1000,
    "repetition_penalty": 1.1,
}
```

In times long past and places veiled in mist, there once was a quiet village nestled between rolling hills blanketed with verdant meadows under a sky of endless blue. The villagers led simple lives full of song and dance during harvest festivals but always yearned for adventure beyond their known horizons. One evening as twilight embraced the land, fires crackling softly against encroaching darkness and children playing hide-and-seek near homes whispering secrets through chimneys, something miraculous unfolded—a light streaked across the heavens like liquid silver cutting into night itself. Whispers rose among men and women alike; they spoke ancient tongues not heard since days when magic still roamed free unbridled by skepticism's evergreen scorn. A young maiden named Lila listened intently from her open windowpane below, senses tingling with every shimmer, daring thoughts spinning within herself like leaves caught upward on wind's whimsical choreography. Tonight might just be forever etched amongst legends; tomorrow could awaken history reborn. And so it began...

Large Language Models Struggle to Learn Long-Tail Knowledge

- Research question: Can LLMs learn rare facts effectively?
- Method: Measured model accuracy by removing specific data
- Focused on “long-tail” or uncommon knowledge



Key Findings

- Accuracy drops sharply for rare facts
- Bigger models perform better, but pattern stays
- Accuracy correlates with “fact count” in training data
- Models mainly memorize, not infer

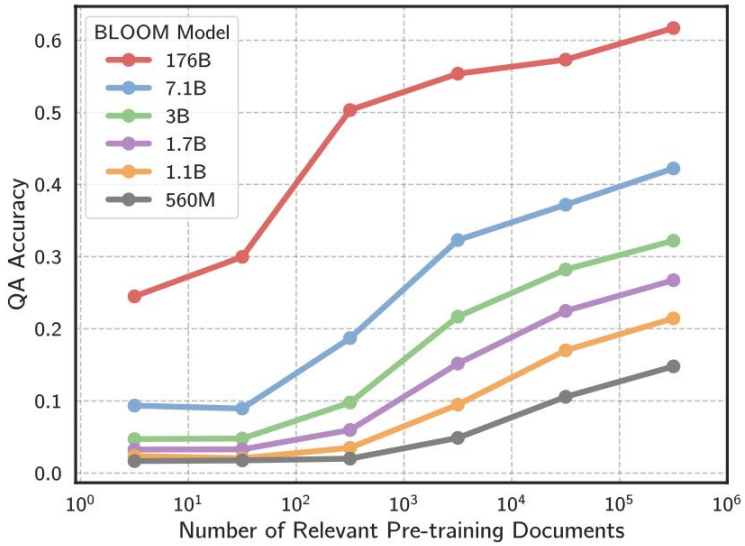


Figure 1. Language models struggle to capture the long-tail of information on the web. Above, we plot accuracy for the BLOOM model family on TriviaQA as a function of how many documents in the model’s pre-training data are relevant to each question.

Humans vs. LLMs

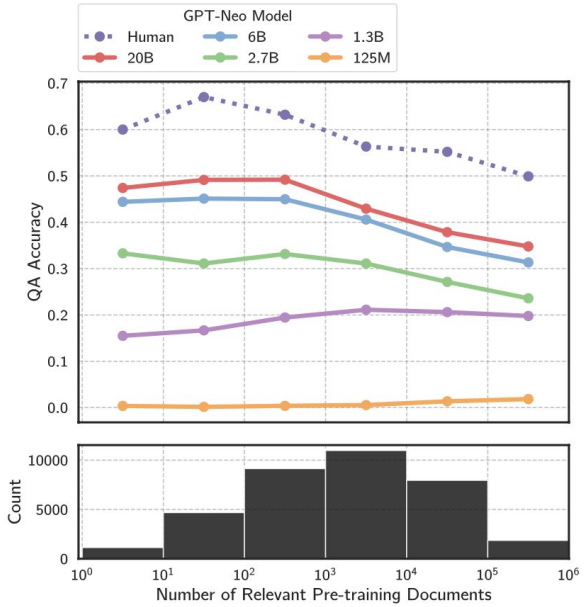


Figure 7. Models with access to the required background context do not struggle on questions with low relevant document count. Concretely, we provide questions and gold paragraphs to GPT-Neo models on Natural Questions, and their accuracy trends roughly match the trends of humans.

- Both asked the same questions
- Humans reason; LLMs recall
- Humans outperform on unseen or rare facts
- Shows need for reasoning and retrieval systems

Implications & Our Takeaways

- Scaling alone isn't enough
- Retrieval and reasoning are key
- Long-tail knowledge affects real-world reliability
- Our takeaway is models still need "understanding," not just data

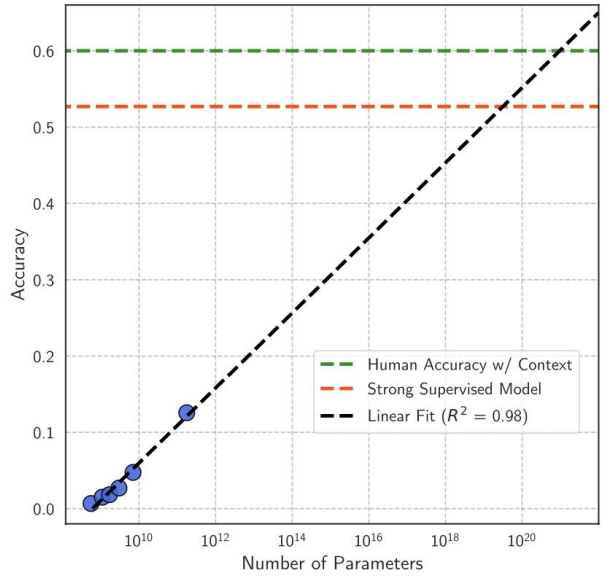


Figure 6. Scaling trends for fact learning. We plot BLOOM accuracy on rare instances from Natural Questions (< 100 relevant docs) as a function of the log of the model size. Extrapolating from the empirical line of best fit—which approximates the trend well at $R^2 = 0.98$—implies that immensely large models would be necessary to get high accuracy.

TinyLlama: An Open-Source Small Language Model

Overview:

- an open-source, compact language model with **≈ 1.1 billion parameters**
- Built on the architecture and tokenizer of Llama 2
- Deemed to have a relatively small size , so not considered a large language model
- Trained on 1 trillion tokens for about 3 epochs

Strengths:

- High efficiency and speed: TinyLlama requires significantly less computational power and memory to run compared to larger language models (LLMs). This allows it to run faster and with less energy.
- Strong performance for its size: pre-trained on a larger dataset compared to other models in its size class. This enables it to perform extremely well on a variety of tasks for a small model, often outperforming other open-source models of comparable size.
- On-device and edge deployment: Its lightweight design allows TinyLlama to run directly on devices with limited resources, such as smartphones, laptops, and Internet of Things (IoT) devices(speakers, thermostats, etc).

Key Vocab

Epochs:
Epochs are one complete pass of the entire training dataset through a learning algorithm.

During a single epoch, the model processes every training example once updating its internal parameter to minimize the loss function and improve performance.

Transformer Architecture:
A deep learning framework that uses self attention mechanisms to process sequences, enabling parallel processing and capturing relationships between words regardless of their position

Details of the model Architecture

Positional Embedding:
A method used in transformer models to provide information about the position of tokens in a sequence, the model used **RoPE** (Rotary Positional Embedding) which unifies absolute and relative approaches

RMSNorm: uses the root mean square of the inputs for scaling and simplifies calculation

Grouped-Query Attention: increases the efficiency of the attention mechanisms in transformer models

Speed Optimization

Fully Shared Data Parallel (FSDP):
Reduces the memory required per GPU by sharding the models parameters, gradients and optimized states

Flash Attention:
a power optimization transformer attention mechanism which **provides 15% efficiency in terms of wall-clock speed** with no approximation.

xFormers:
Allowed the developers to reduces the memory footprints and enabled the 1.1B model to fit within 40 GB of GPU RAM

Performance Analysis and Comparison with Other Models:
Because of these speed optimizatiopn elements the tiny llama is abole to perform at 24,000 tokens per secon per A100-40G GPU, demonstrating a superior training speed

Conclusion and Results

Was evaluated on a wide range of commonsense reasoning and problem solving tasks and compared to other existing open source language models with similar model parameters.

TinyLlama outperformers baselines on many of the tasks and obtains the highest average scores for commonsense reasoning tasks, and also demonstrates better problem solving skills compared to existing models.