Day 3 - AL Block Seminar

Ahmad Dawar Hakimi

06.07.2025

Seminar Overview

Day	Morning (9:00-12:00)	Afternoon (13:00-16:00)
Day 1	Outline, Motivation Foundations, Query Strategies	Coding - Query Strategies
Day 2	Deep Active Learning Practical Considerations, AL in the Era of LLMs (1/2)	Coding - Deep Active Learning
Day 3	Active Learning in the Era of LLMs (2/2), Synthetic Data Genration, Useful Resources	Coding - LLMs in AL Cycle
Day 4	Prepare Pitch	Pitches + Q&A

Day 3 - Active Learning in the Era of LLMs

- Learning Objectives
  - How to annotate data?
  - Synthetic Data Generation
  - (Optional) Uncertainty Estimation for Language Models

How to annotate data?

Why Clear Guidelines Matter?

- **Consistency:** Ensures all annotators interpret “positive” and “negative” the same way
- **Data Quality:** Reduces noise from ambiguous or contradictory labels
- **Model Performance:** Reliable labels lead to better learning and evaluation
- **Efficiency:** Fewer disputes, faster throughput once rules are internalized

Task Definition & Label Schema

- Objective: Label each review/sentence as one of:
  - Positive → expresses clear approval or satisfaction
  - Negative → expresses clear disapproval or dissatisfaction
  - Neutral → factual or mixed sentiment (no strong valence)
- Examples:
  - Positive: “I loved the cinematography!”
  - Negative: “The plot was unbearably slow.”
  - Neutral: “The movie runs for two hours.”

Annotation Rules & Edge Cases

- Mixed Sentiment: If one sentiment dominates, choose that; otherwise mark Neutral
- Sarcasm & Irony: Look for cues (quotation marks, emojis); when in doubt, annotate by implied meaning
- Modifiers & Intensifiers:
  - “Very good” → Positive
  - “Not bad” → Positive (avoid literal negation trap)
- Comparative Statements: “Better than the last one” → Positive (relative praise)

Quality Control & Iteration

- Pilot Annotation: label 100 samples, compute Cohen’s κ; target ≥ 0.8
- Adjudication: discuss disagreements in weekly syncs; update guidelines with FAQs
- Gold-Label Checks: embed 5 known examples per batch; flag annotator drift
- Versioning: track guideline changes alongside data snapshots

Tools & Best Practices

- Use annotation platforms (Label Studio, Doccano, Prodigy) with built-in validation
- Document Decisions: maintain an FAQ log for new edge cases
- Annotator Training: walkthrough guidelines and examples before starting
- Regular Feedback: share model errors back to annotators to refine boundaries
- Batch Size & Breaks: limit to ~200 labels/session to avoid fatigue-driven errors

Synthetic Data Generation

Language models are built on data

Pre-training

Raw text x ⟶ P(x)

Supervised Fine Tuning

Input x, output y ⟶ P(y | x)

Reasoning Training

Input x, output y, latent reasoning z ⟶ P(y | z, x) P(z | x)

Where do we get the data?

- Scraping the internet
- Labeling manually
- Collecting from system users
- Creative curation

Why is this not enough?

- Scraping the internet
- Labeling manually
- Collecting from system users
- Creative curation

- Too noisy, too massive
- Too expensive, annotators not available
- Chicken and Egg Problem, privacy implications
- Limited Applicability

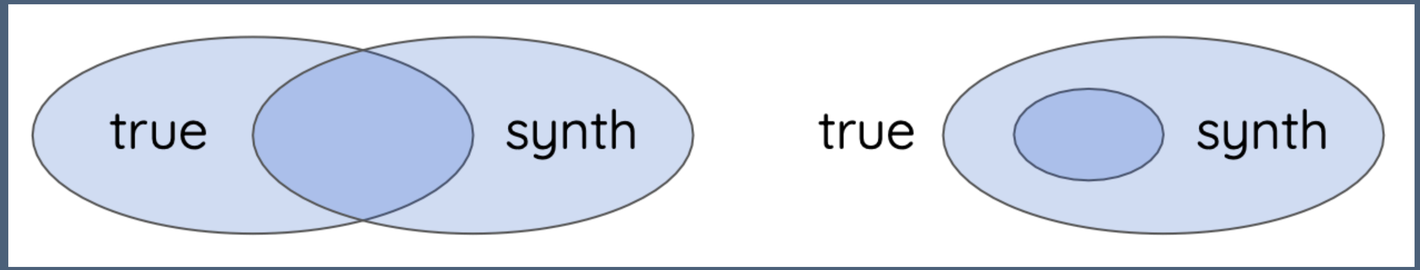
Synthetic data to the rescue!

- Create data order-made that is
- Relatively clean
- Appropriately sized (not too big/small)
- Tailored to individual tasks
- Flexible

But generating good synthetic data is hard…

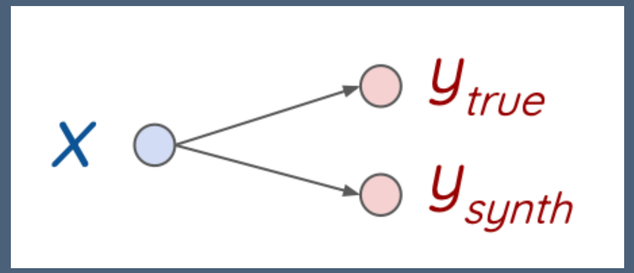
- The input distribution may be **off**, or not **diverse enough**

$$P_{true}(x) \neq P_{synth}(x)$$



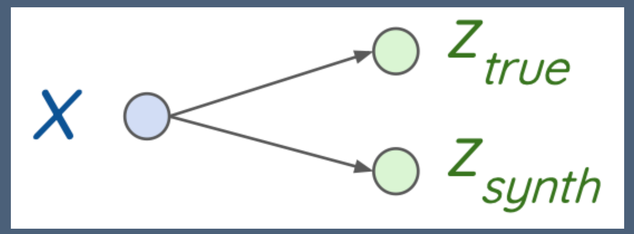
- The labels may be **wrong**

$$P_{true}(y \mid x) \neq P_{synth}(y \mid x)$$



- The reasoning may be **flawed**

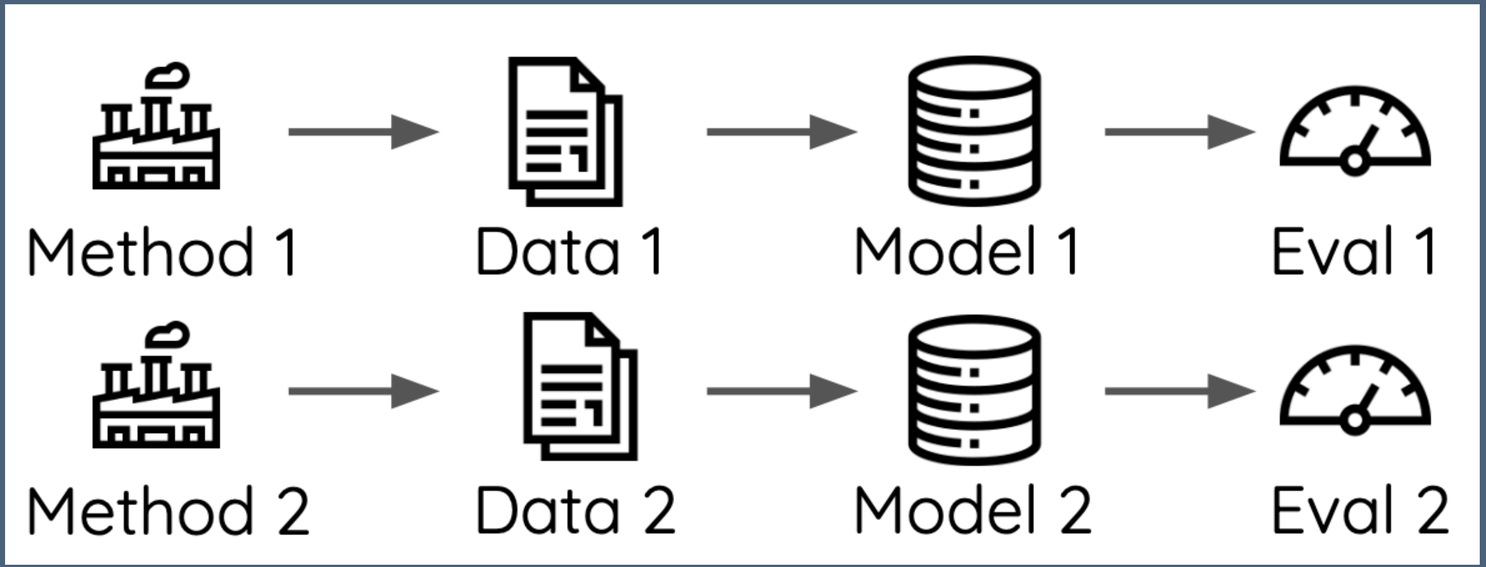
$$P_{true}(z \mid x) \neq P_{synth}(z \mid x)$$



What is “high-quality” synthetic data?

Evaluation of synthetic data

- Extrinsic: Does it help in a downstream task?



- Intrinsic: What are the characteristics of the data or generation process?

Intrinsic Eval: Data Correctness

- Questions regarding whether the data is correct, judged by manual or automatic methods
- E.g. Self-Instruct manually annotates:

Quality Review Question	Yes %
Does the instruction describe a valid task?	92%
Is the input appropriate for the instruction?	79%
Is the output a correct and acceptable response to the instruction and input?	58%
All fields are valid	54%

Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et al., 2022)

Intrinsic Eval: Data Diversity/Coverage

- How well does the generated data cover the plausible data
- E.g. DataTune evaluates bigram diversity

Dataset	Unique Bigrams Per Example	Total Tokens Per Example
Code Line Description		
Gold	13.2	32.3
Synthetic	2.5	35.0
Transformed	14.9	86.9
Elementary Math		
Gold	10.8	48.6
Synthetic	3.3	34.4
Transformed	11.6	43.8
Implicatures		
Gold	9.9	24.1
Synthetic	2.7	27.7
Transformed	17.8	39.8
Temporal Sequences		
Gold	1.0	99.7
Synthetic	20.8	54.6
Transformed	0.2	73.7
Medical Questions in Russian		
Gold	62.0	79.4
Synthetic	20.8	54.6
Transformed	11.6	44.8

Table 3: We observe that dataset transformation yields datasets with greater lexical diversity than synthetic dataset generation on 3 of 5 datasets.

Better Synthetic Data by Retrieving and Transforming Existing Datasets (Gandhi et al., 2024)

Intrinsic Eval: Other Metrics

- Many other dimensions, e.g. privacy, fairness, distributional
- E.g. SynthTextEval toolkit

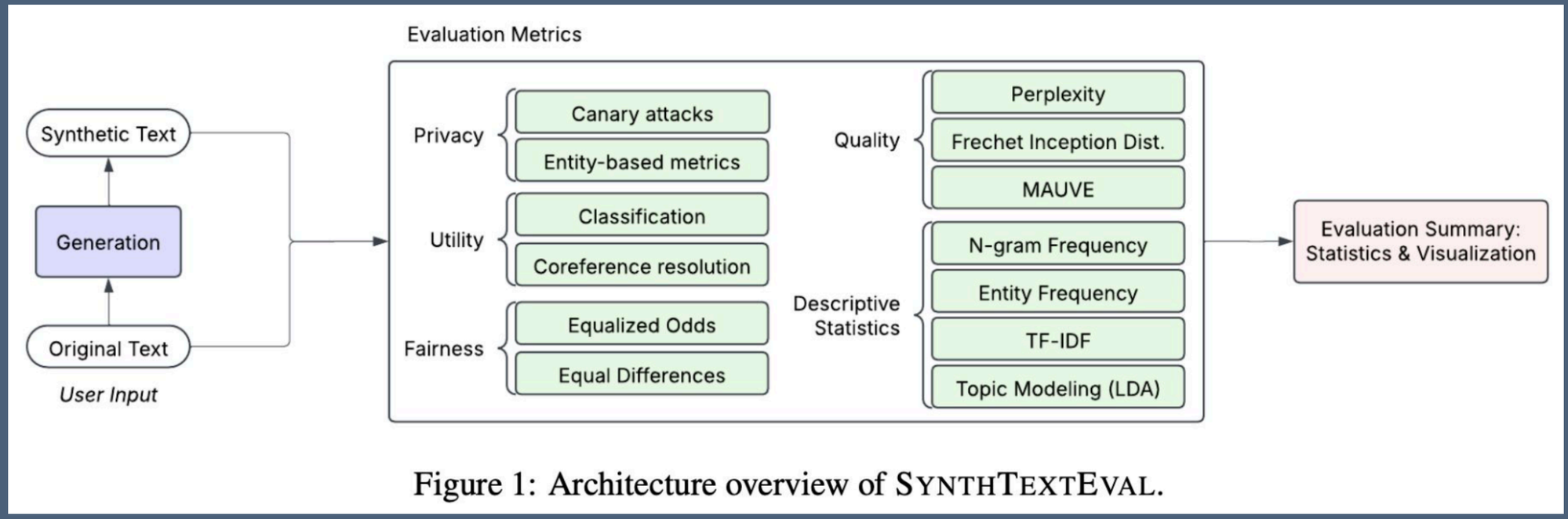


Figure 1: Architecture overview of SYNTHTEXTEVAL.

SynthTextEval: Synthetic Text Data Generation and Evaluation for High-Stakes Domains (Ramesh et al. 2025)

Evaluating Language Models as Data Generators

- We can also evaluate language models based on their ability to generate synthetic data
- E.g. AgoraBench, which measures synthetic data by different LMs based on its ability to match manually created data (at what cost)

Data Generator	API Cost		Prob. Solv.	Data Gen.
	Input	Output	Avg	Agora Bench
GPT-4o	$2.50	$10.00	80.9	29.5%
GPT-4o-mini	$0.15	$0.60	75.4	19.2%
Claude-3.5-Sonnet	$3.00	$15.00	80.5	23.6%
Llama-3.1-405B	$1.79	$1.79	75.0	11.3%
Llama-3.1-70B	$0.35	$0.40	69.6	14.1%
Llama-3.1-8B	$0.055	$0.055	50.2	15.9%

Evaluating Language Models as Synthetic Data Generators (Kim et al. 2024)

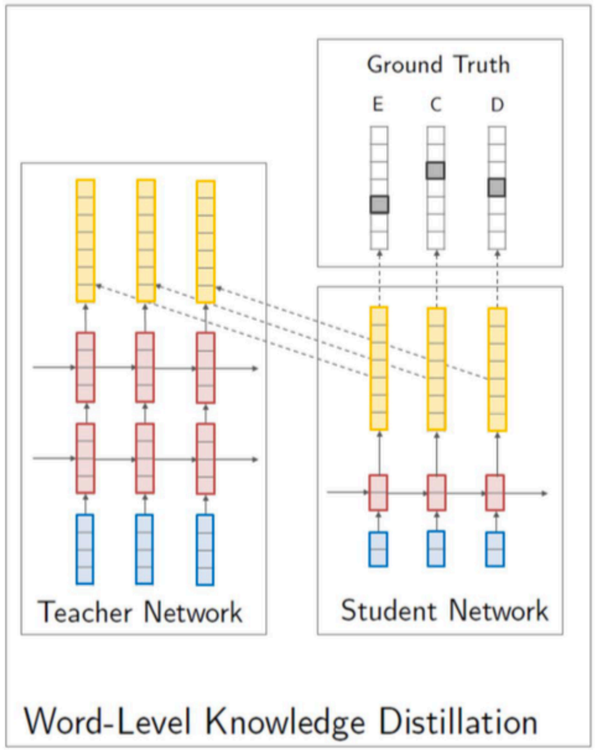
How do we create synthetic data?

Approaches to synthetic data creation

- Sampling-based generation
- Back-translation
- Transformation of existing data
- Human-AI collaboration
- Symbolic generation

Background: knowledge distillation

- Train student model to mimic the teacher's predicted probability distribution (e.g., over words)

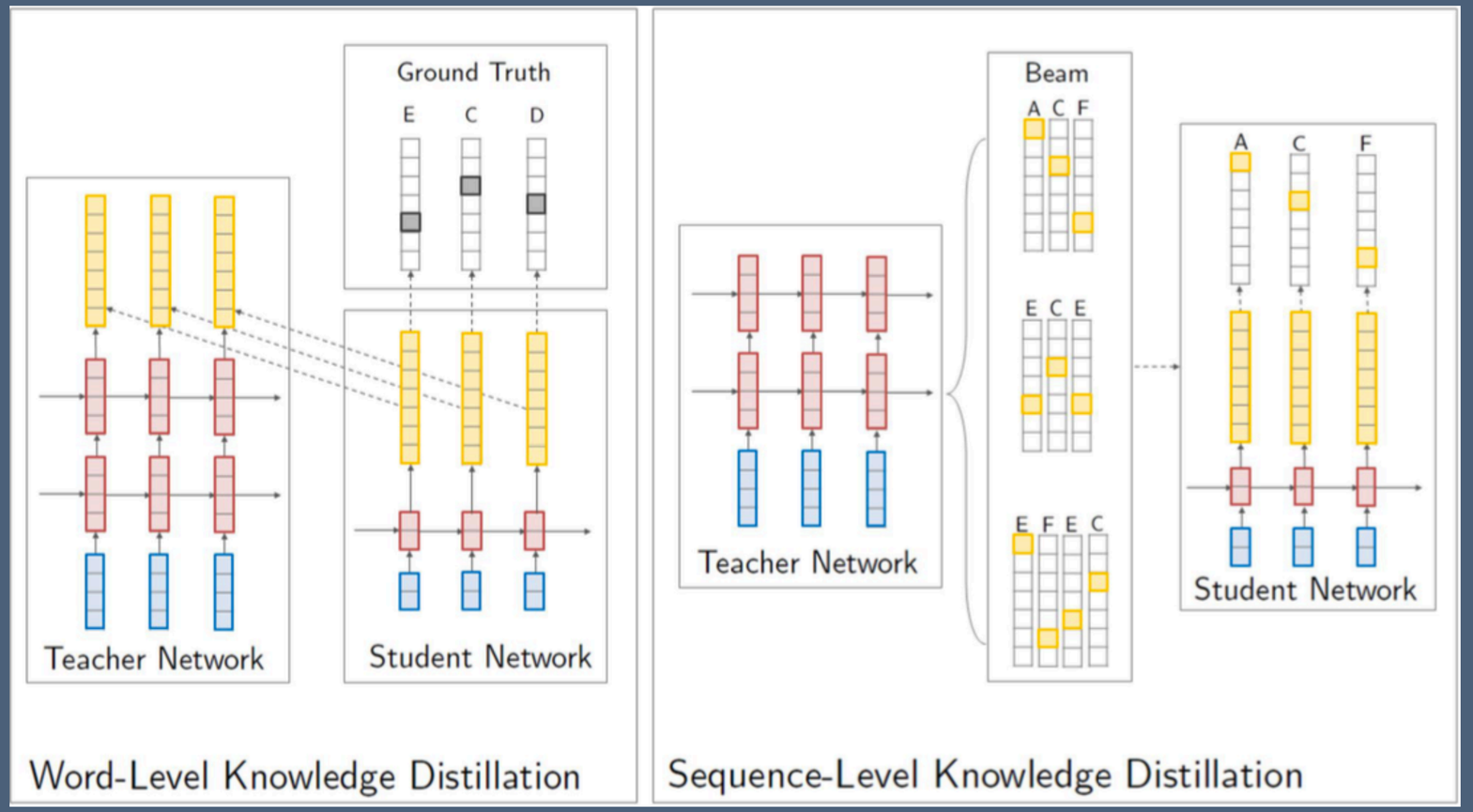


"Once the cumbersome model has been trained, we can then use a different kind of training, which we call "distillation" to transfer the knowledge to a small model"

Distilling the Knowledge in a Neural Network (Hinton et al., 2015)

Sequence -level knowledge distillation

- Train student on complete generations (i.e., sequences of words) from the teacher



Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

Generating task data from LMs

- Use GPT-3's in-context learning ability to generate new examples of arbitrary tasks

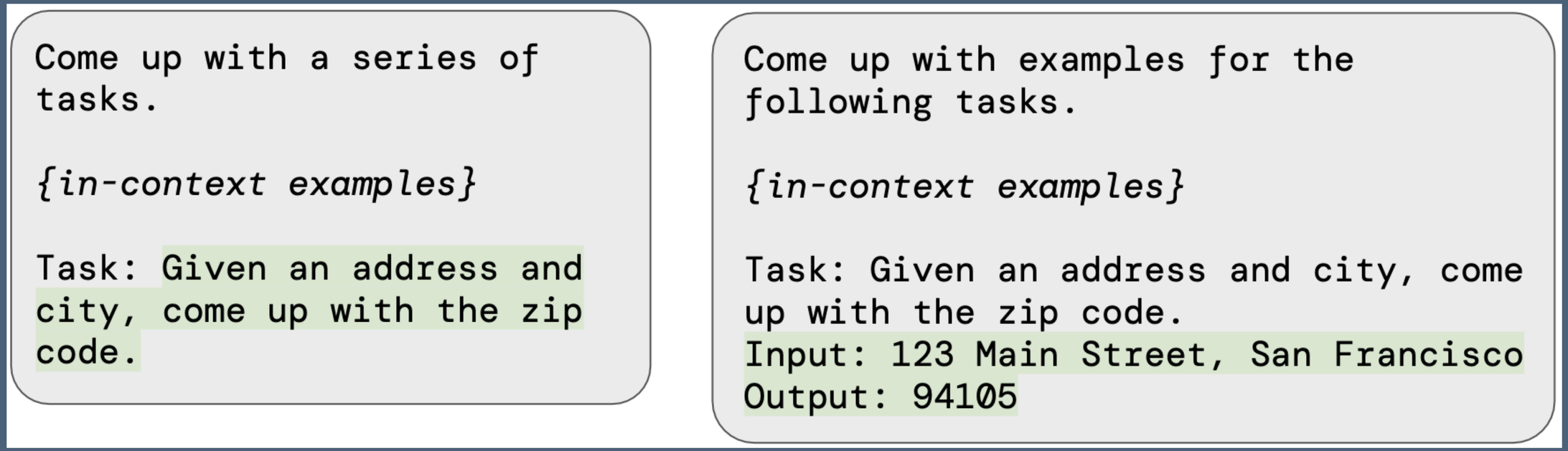
```
Task: Write two sentences that mean
the same thing

Sentence 1: A man is playing the flute
Sentence 2: He's playing the flute
```

Create sentence-similarity examples by prompting the model to write similar (or dissimilar) sentences!

Generating Datasets with Pretrained Language Models (Schick & Schütze, 2021)

Generating instruction data from scratch

- Instead of generating more examples under a given task, generate completely new tasks

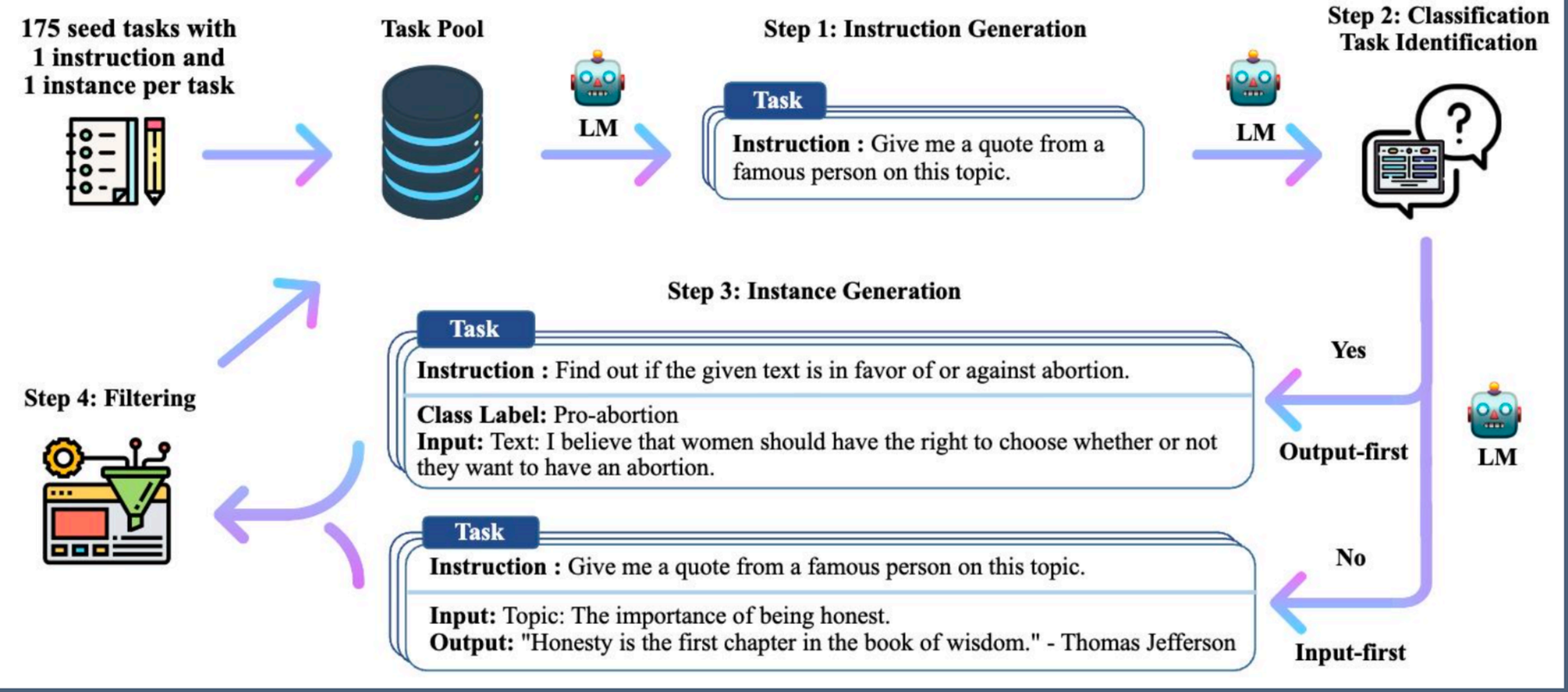


Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et al., 2022)

Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor (Honovich et al., 2022)

Generating instruction data from scratch

From just 175 seed examples → ~100K new examples

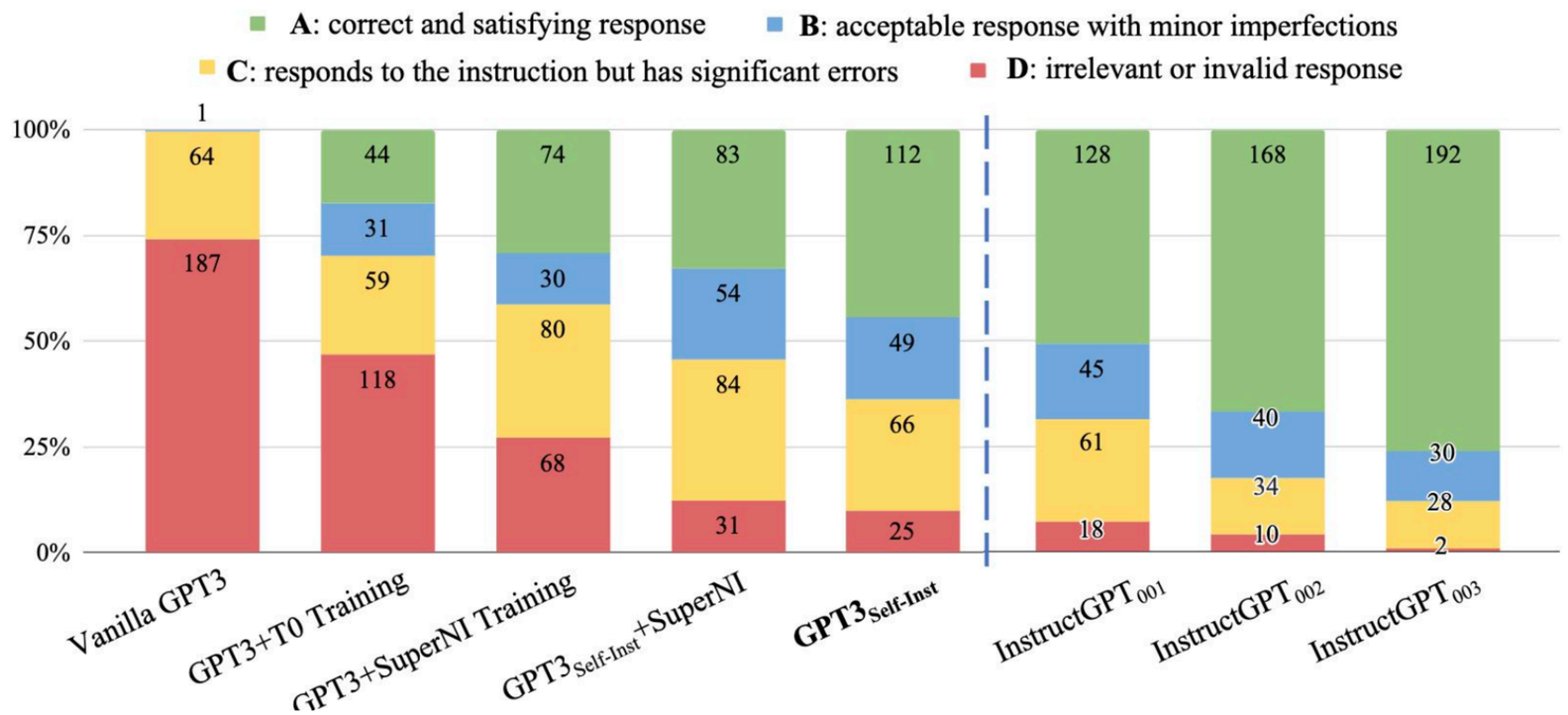


Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et al., 2022)

Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor (Honovich et al., 2022)

Generating instruction data from scratch

Finetuning GPT-3 on self-generated data improves over existing instruction datasets

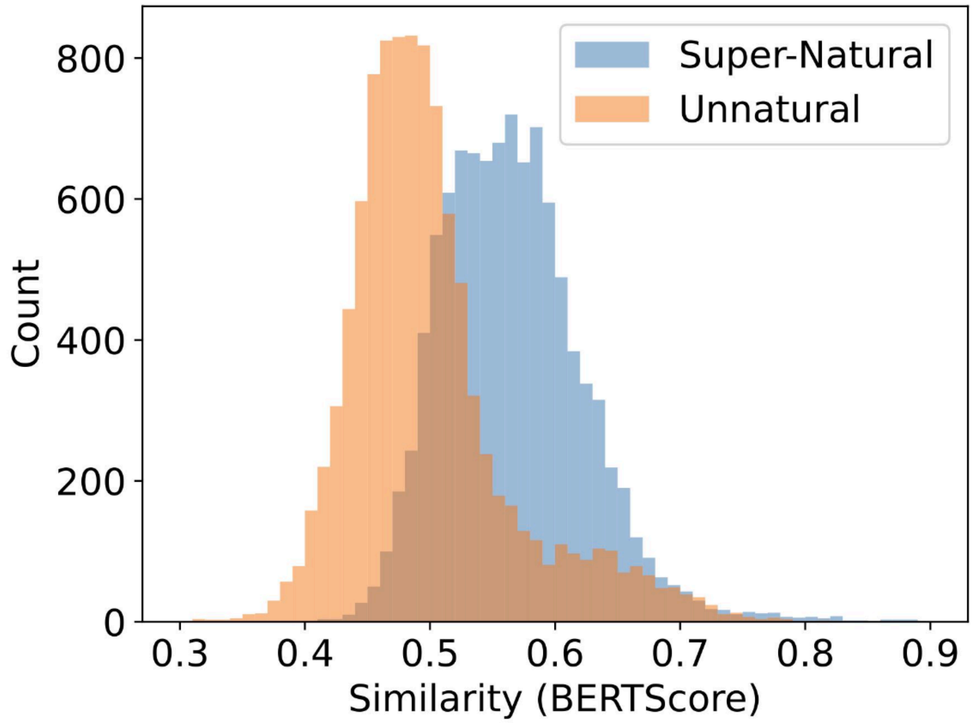


Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et al., 2022)

Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor (Honovich et al., 2022)

Generating instruction data from scratch

Generated data is more diverse than human-written data



Generating instruction data from scratch

In both Self-Instruct data and Unnatural Instructions, only half of the examples are actually correct (!!)

Quality Review Question	Yes %
Does the instruction describe a valid task?	92%
Is the input appropriate for the instruction?	79%
Is the output a correct and acceptable response to the instruction and input?	58%
All fields are valid	54%

Table 2: Data quality review for the instruction, input, and output of the generated data.

113 of the 200 analyzed examples (56.5%) are correct. Of the 87 incorrect examples, 9 (4.5%) had incomprehensible instructions, 35 (17.5%) had an input that did not match the task description, and 43 (21.5%) had incorrect outputs.

Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et al., 2022)

Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor (Honovich et al., 2022)

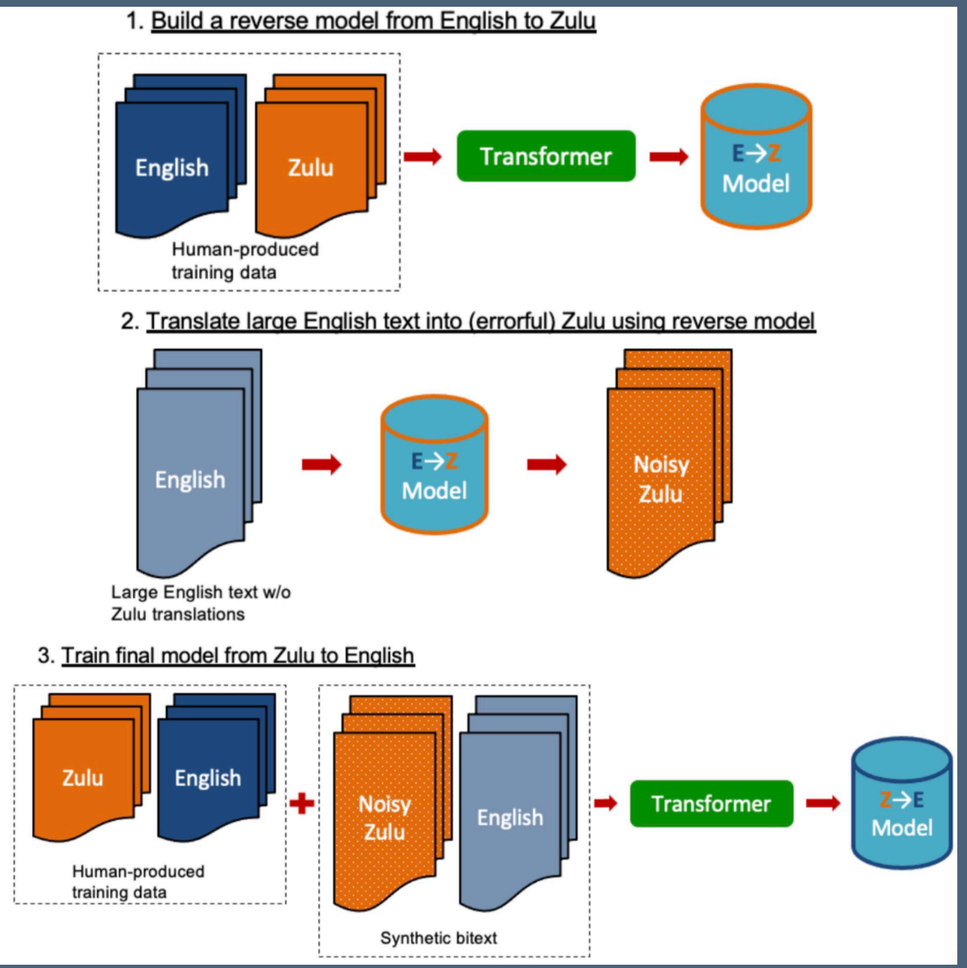
Takeaways from early efforts

- Synthetic data can reflect creativity & diversity difficult to elicit from crowdworkers
- Diversity can be more valuable than correctness!
- Synthetic data can sometimes enable self-improvement
- Data creation becomes a complex pipeline

Backtranslation in MT

Given an output, generate a corresponding input

Since models are trained to produce outputs, we want those to be natural (inputs can be unnatural)



Improving Neural Machine Translation Models with Monolingual Data (Sennrich et al., 2016)

Instruction back-translation

Given web text y, generate instruction x for which y would be a good response

```
Output: It doesn't matter where you are in the world, how old
you are, or how much you know about meditation, it's for
everyone. The benefits of meditation are endless. Medication
can be as simple as sitting quietly for five minutes…

What kind of instruction could this be the answer to?
Instruction: Write an essay about the benefits of meditation.
```

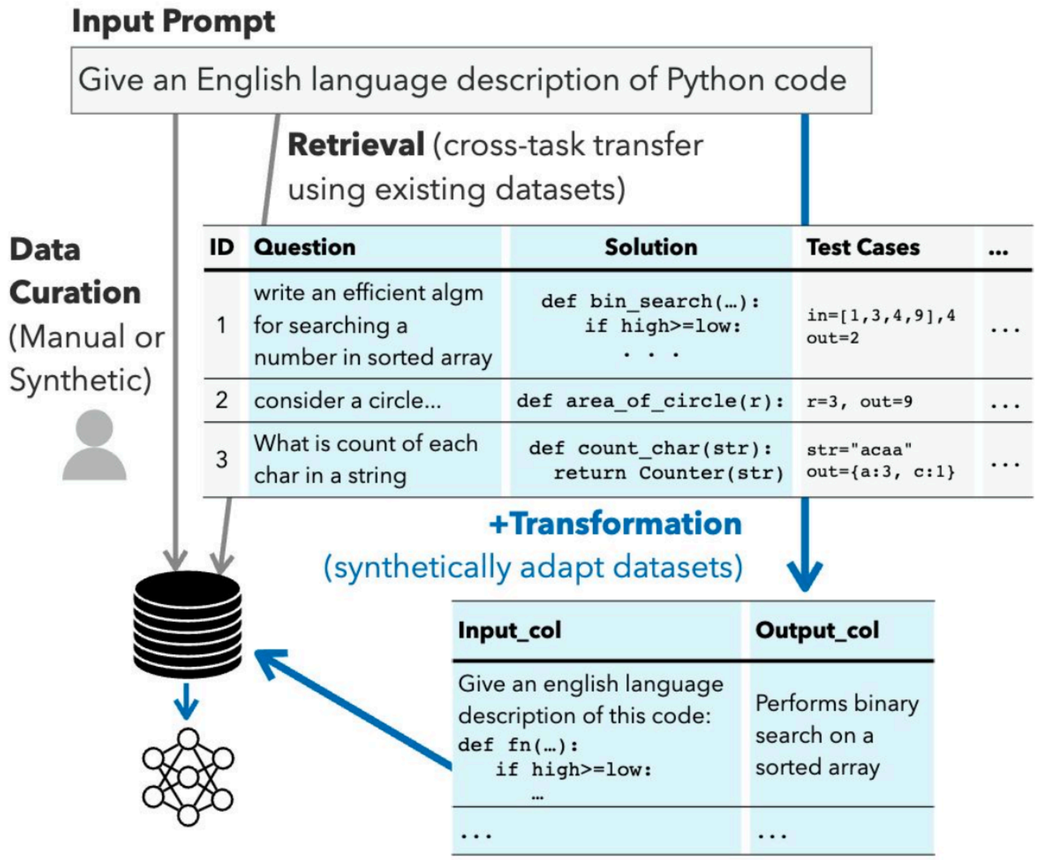
LongForm: Effective Instruction Tuning with Reverse Instructions (Köksal et al., 2023)

Self-Alignment with Instruction Backtranslation (Li et al., 2023)

Transformation of existing data

Given a task,

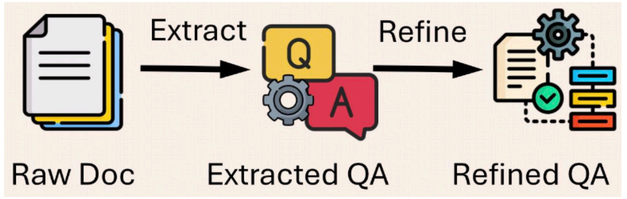
1. Retrieve relevant datasets or documents
2. Transform them into data under the desired task



Better Synthetic Data by Retrieving and Transforming Existing Datasets (Gandhi et al., 2024) Synthetic Dataset Generation Through Corpus Retrieval and Augmentation (Ziegler et al., 2024)

Extract instruction data from the web

Identify pages that may contain questions & answers, then extract and refine them!



Raw Docs *Unformatted Text, Site Information, Ads*

Topics Science\nAnatomy&Physiology\nAstronomy\nAstrophysics \nBiology\nChemistry \n...Socratic Meta...Featured Answers
How do you simplify #((u^4v^3)/(u^2v^-1)^4)^0# and write it using only positive exponents?
Answer by NickTheTurtle (Apr 1, 2017)
Explanation:\nAnything raised to the \(0^{th}\) power is simply 1. \n\nRelated Questions\n What is the quotient of powers property? \n\n\n\n How do you simplify expressions using the quotient rule?...\nImpact of this question\n1274 views around the world #Apps\niOS\nAndroid\nLinks\n[Privacy](#)\n[Terms](#)\n[Help](#)

Extract

Extracted QA *Formatted QA but lacking detailed solutions*

Question: How do you simplify $(u^4\,v^3/(u^2\,v^{-1}\,)^4)^0$ and write it using only positive exponents?
Answer: Explanation: Anything to the $0th$ power is just simply 1.

Rewrite

Rewritten QA *Formatted QA augmented with detailed solutions*

Question:
How do you simplify $(u^4\,v^3/(u^2\,v^{-1}\,)^4)^0$ and write it using only positive exponents?

Answer:
To simplify the expression $(u^4\,v^3/(u^2\,v^{-1}\,)^4)^0$ and rewrite it using only positive exponents, we start by evaluating the expression from the innermost operation outward.

First, consider the exponent of zero on the entire fraction:
Any expression raised to the power of zero is equal to 1 (provided the base is not zero). *Therefore*, $(u^4\,v^3/(u^2\,v^{-1}\,)^4)^0 = 1$

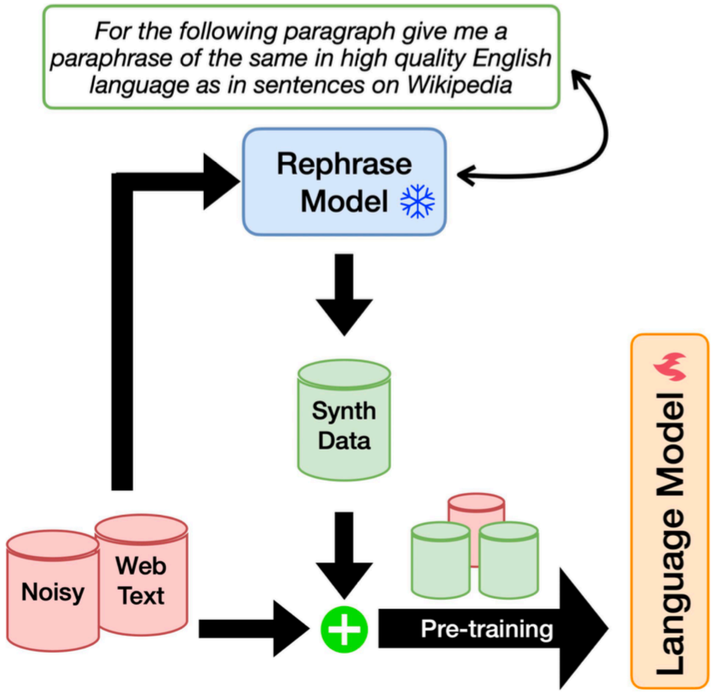
This simplification makes the other calculations unnecessary because raising any non-zero expression to the power of zero will always result in 1.

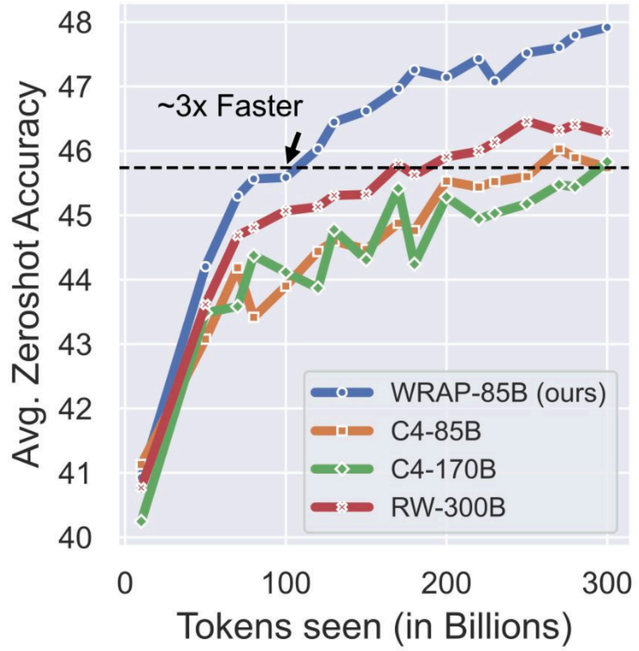
So, the simplified expression is simply: 1

MAmmoTH2: Scaling Instructions from the Web (Yue et al., 2024)

Rephrasing documents for pretraining

Use LMs to paraphrase noisy web text to create new data!

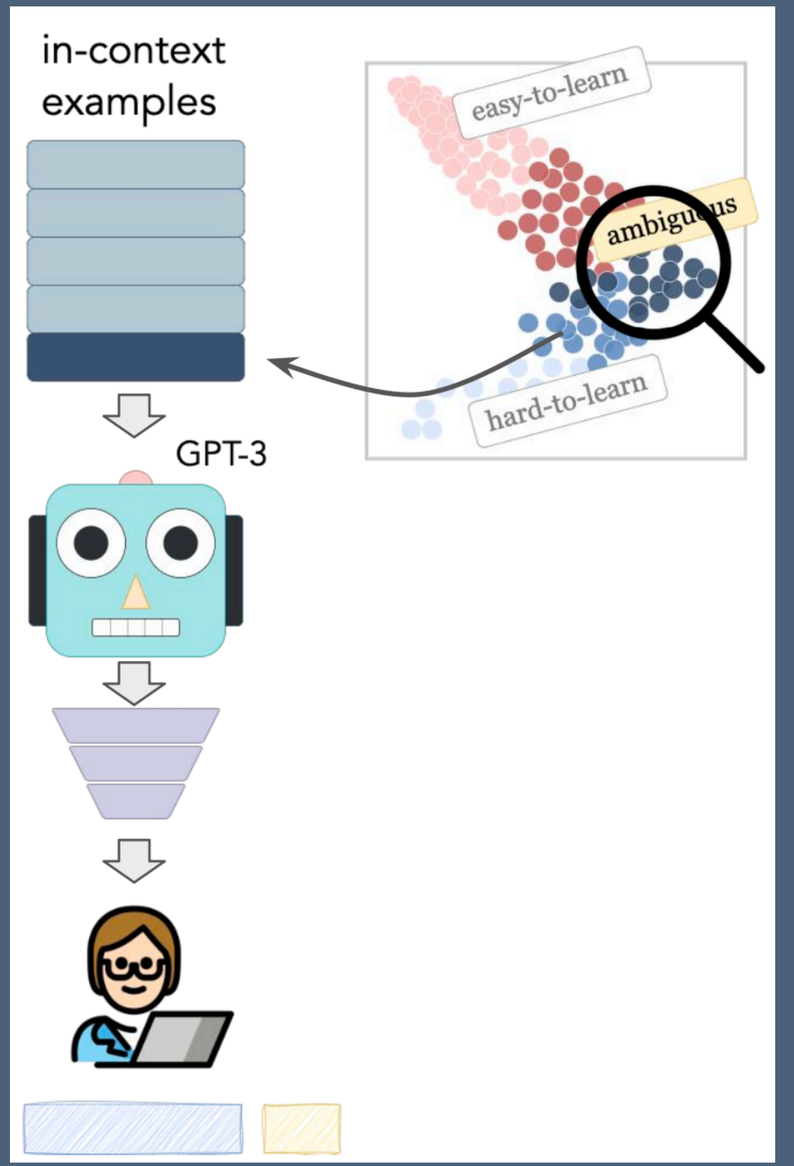




Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling (Maini et al., 2024)

Human-AI collaboration

- LMs are creative & diverse, but not reliably correct
- Humans can verify & improve correctness, but are not good at enumerating what they know
- Combine the best of both worlds for data creation!
- Crowdworkers revise & label generated data
- Turns writing task into editing task!

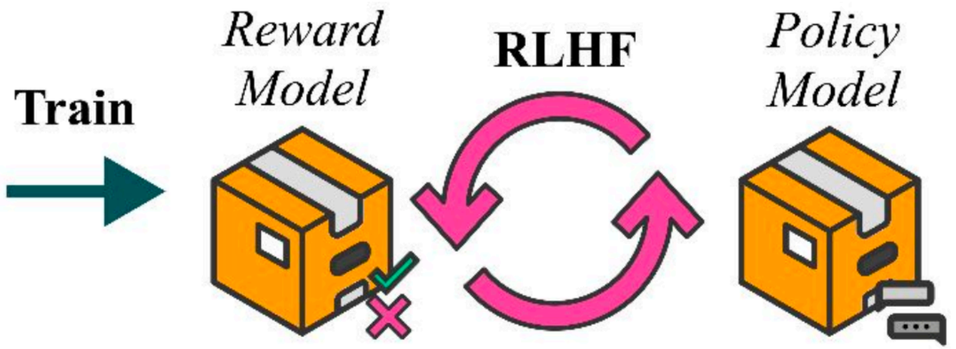


WANLI: Worker and AI Collaboration for Natural Language Inference Dataset Creation (Liu et al., 2022)
SynthBio: A Case Study in Faster Curation of Text Datasets (Yuan et al., 2021)

Route instances for human vs. AI feedback

Use router to solicit preference annotation from either human or LM

Prompt	A	B	Annotator
$x^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$	
$x^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$	
⋮	⋮		⋮
$x^{(n)}$	$y_1^{(n)}$	$y_2^{(n)}$	

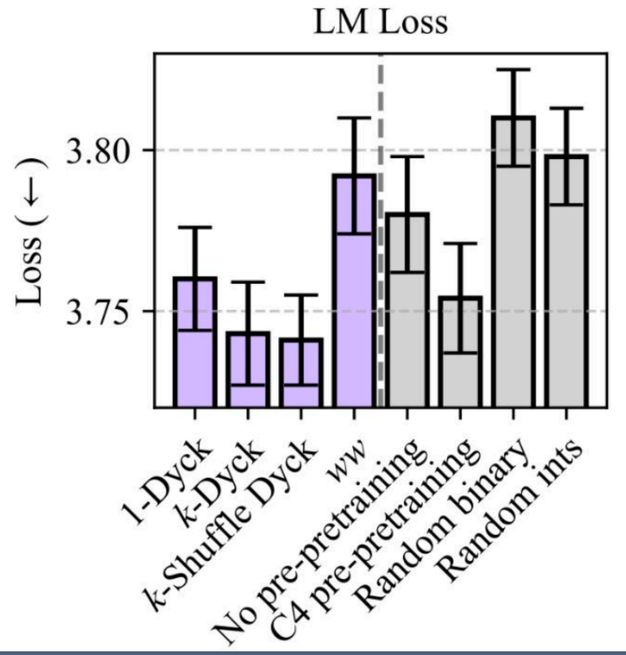


Hybrid Preferences: Learning to Route Instances for Human vs. AI Feedback (Miranda et al., 2024)

Symbolic generation

Doing initial pretraining on *formal languages* can lead to faster LM training and better generalization

Language	Example
1-Dyck	((( )))
k-Dyck	([{ }])
k-Shuffle Dyck	([{]) }
ww	1 2 3 1 2 3



Between Circuits and Chomsky: Pre-pretraining on Formal Languages Imparts Linguistic Biases (Hu et al., 2024)

Summary

- Sampling-based generation: Generate examples from scratch from LMs
- Back-translation: Given an output, generate an input
- Transformation of existing data: Transform existing data into examples of the desired task
- Human-AI collaboration: Mix LM generation & human annotation
- Symbolic generation: Rule-based generation

Data Filtering: surface-level heuristics

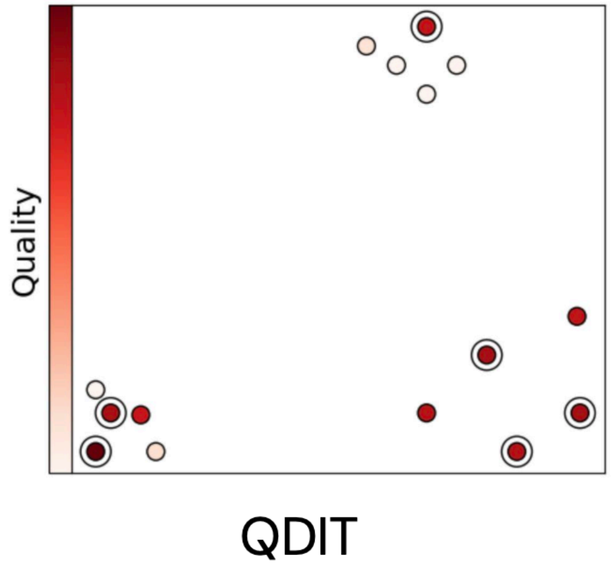
Filter similar examples as defined by

- Rouge-L (Self-Instruct; Impossible Distillation)

Data Filtering: surface-level heuristics

Filter similar examples as defined by

- Rouge-L (Self-Instruct; Impossible Distillation)
- Embedding similarity (QDIT, DiverseEvol, DEITA)



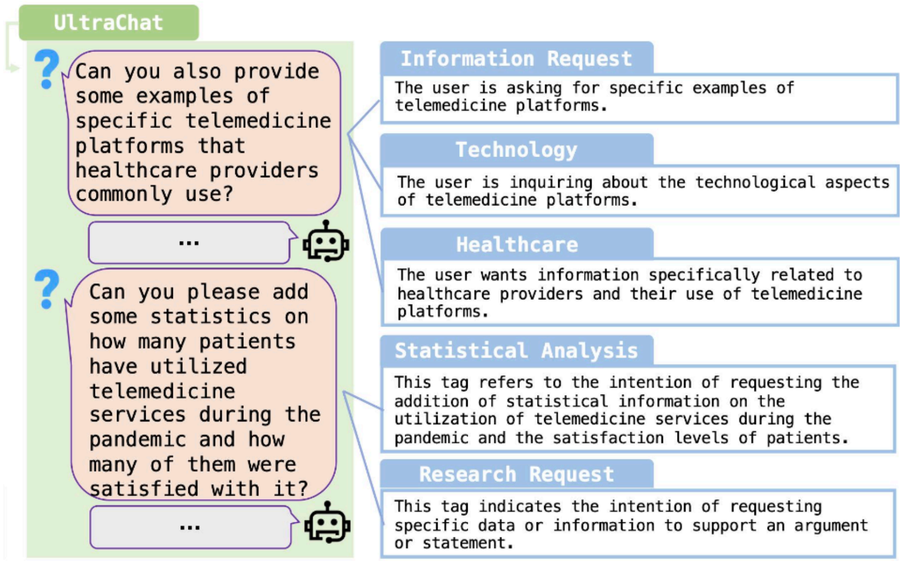
Data Filtering: surface-level heuristics

Filter similar examples as defined by

Rouge-L (Self-Instruct; Impossible Distillation)

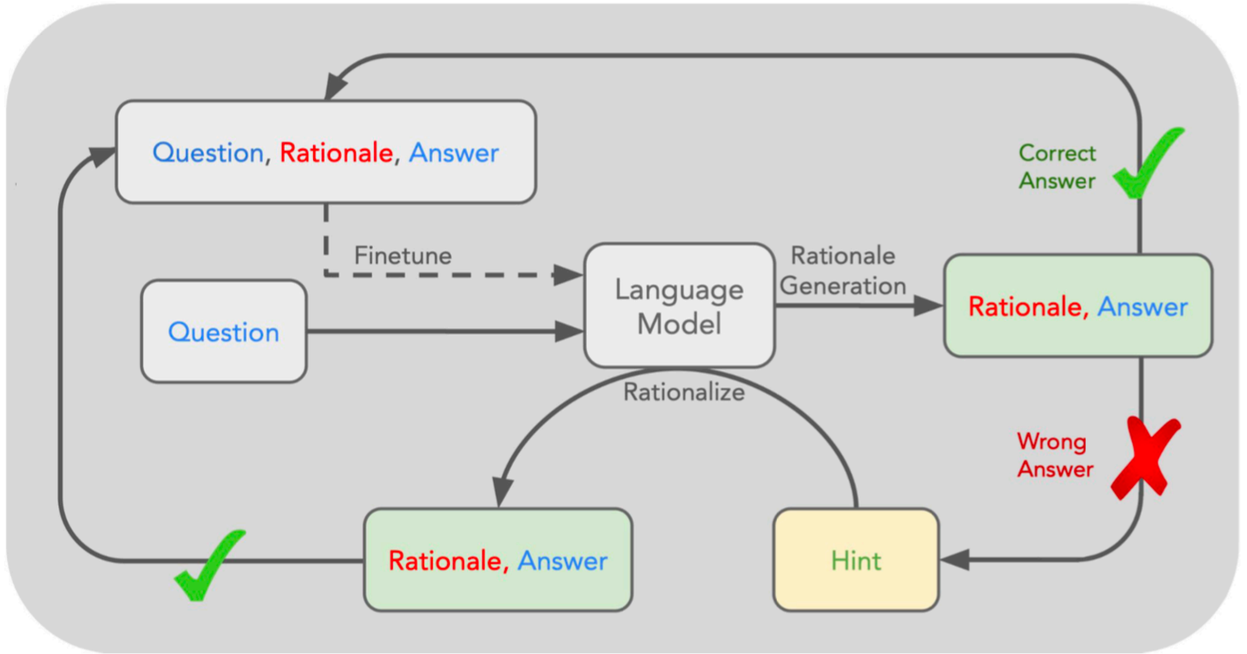
Embedding similarity (QDIT, DiverseEvol, DEITA)

Semantic tags (#InsTag)



Correctness filtering: final answer verification

When generating synthetic reasoning data, only keep generations whose final answers are correct



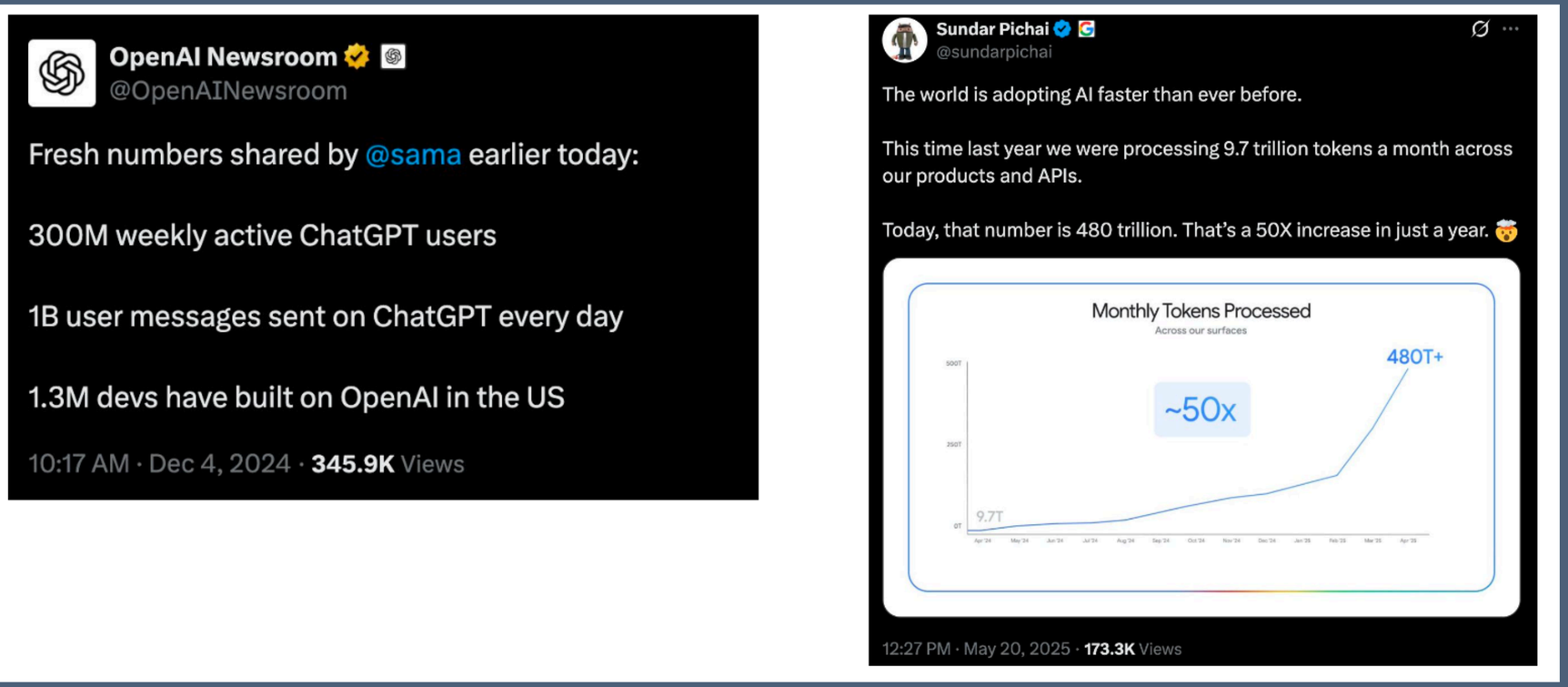

```
Q: What can be used
to carry a small dog?
Answer Choices:
(a) swimming pool
(b) basket
(c) dog show
(d) backyard
(e) own home
A: The answer must be
something that can be
used to carry a small
dog. Baskets are
designed to hold things.
Therefore, the answer
is basket (b).
```

STaR: Bootstrapping Reasoning With Reasoning (Zelikman et al., 2022) DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via RL (Deepseek-AI, 2025)

Limitations and Open Challenges

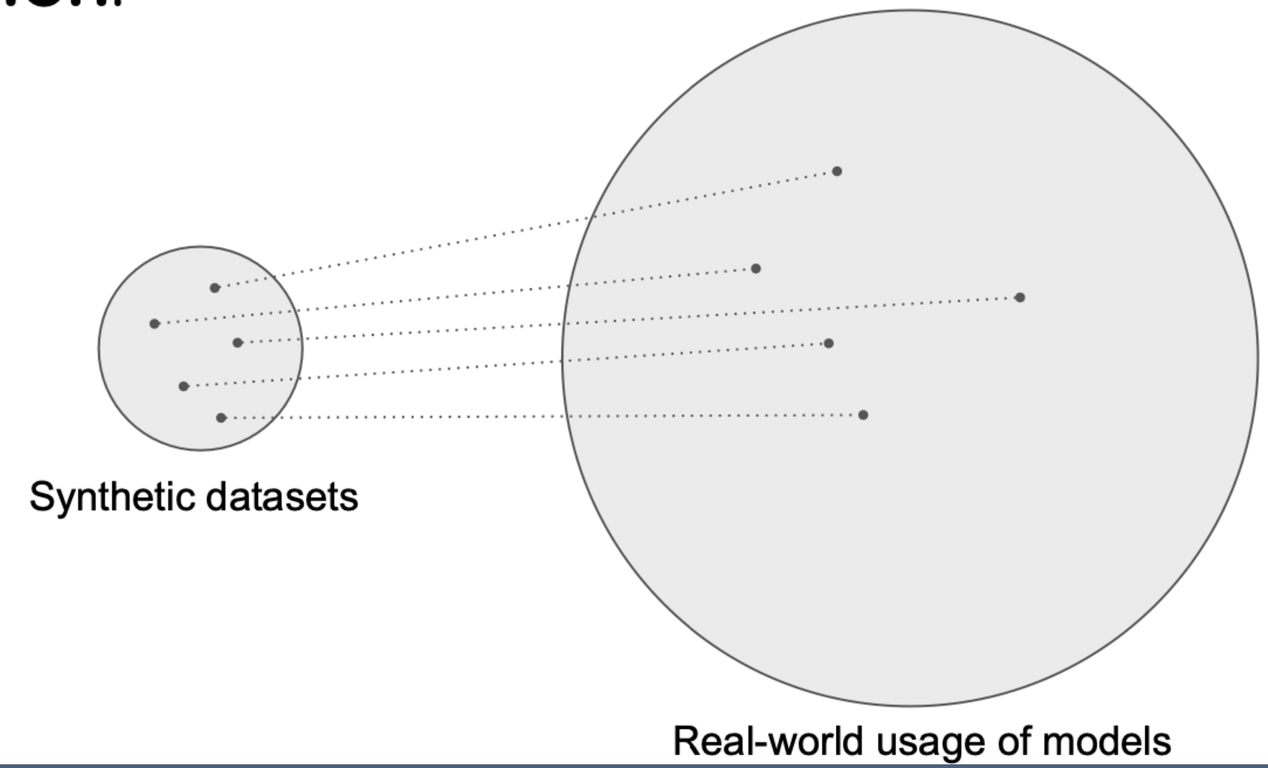
Synthetic vs Real-world data

- Real-world data refers to data produced by real users as they interact with a real product or service.
- The size of real-world user interactions is increasing fast!



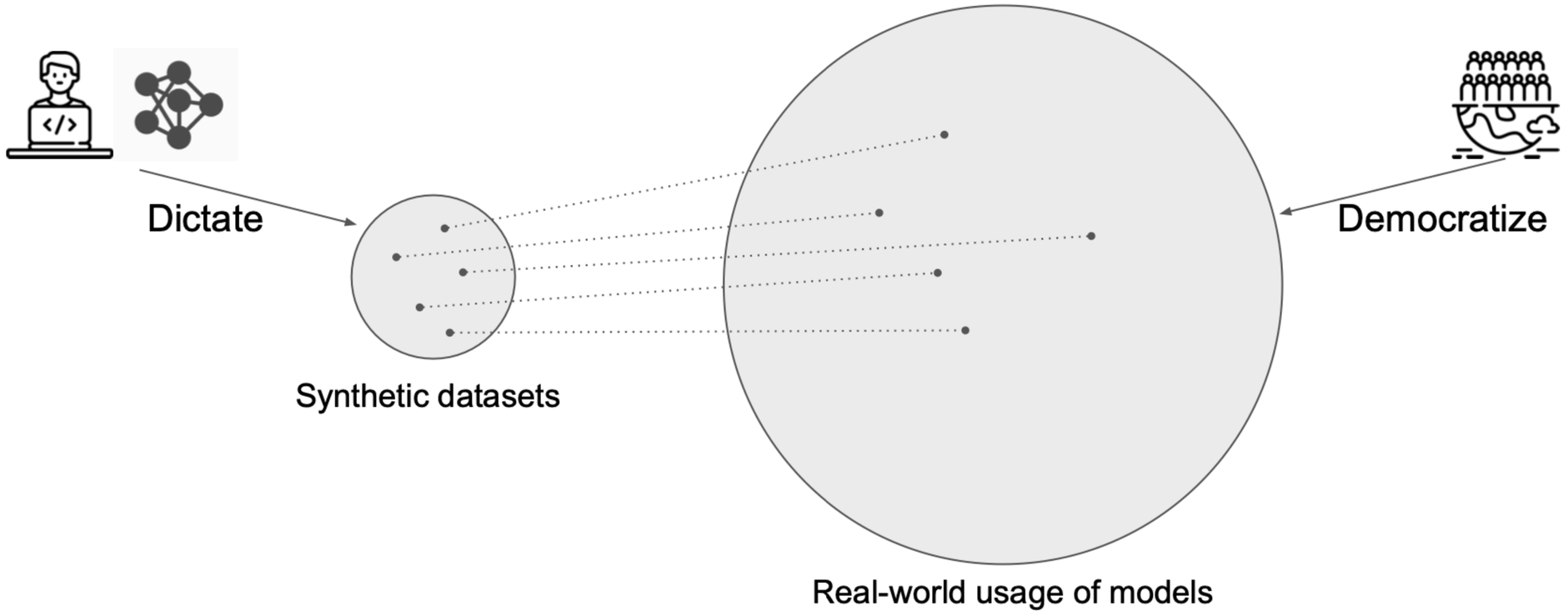
Synthetic vs Real-world data

There is still a significant **gap in size, diversity, & distribution**!



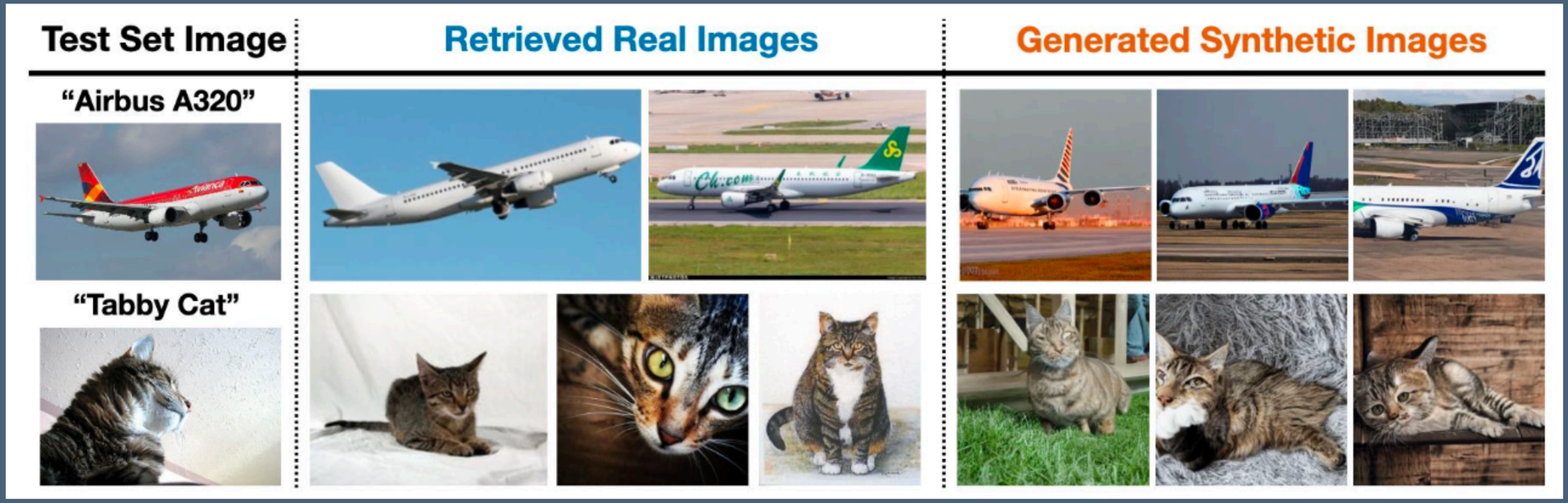
Synthetic vs Real-world data

- Fundamentally, they are produced differently.



How does synthetic data compare to real data in terms of quality?

- In a controlled setup, synthetic data still underperforms real data (if available) [Geng et al.,
- Analysis shows synthetic data often contains generator artifacts and distort class-level visual content.



How can we measure the quality of data?

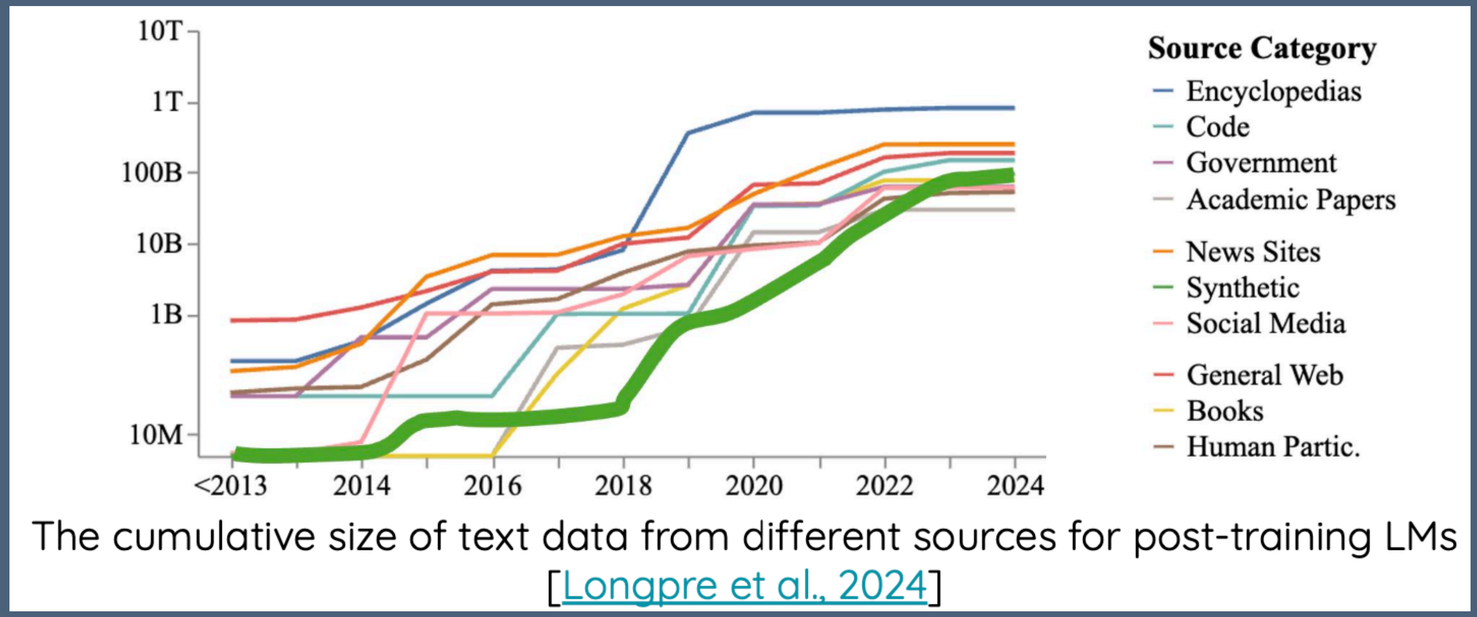
- Quality control is a critical step in classic data pipeline [Daniel et al., 2018]
- Many synthetic datasets from the community do not implement quality checks.
- There are two common proxies for synthetic data quality:
  - Better downstream performance of the trained model implies better quality of the data.
  - Distilling from stronger generation models (e.g., GPT4) produces data of better quality.

Ideally, we also want to measure instance-level quality

- Reward models
- LLM as a judge
- Rule-based verification
- Decomposition [Min et al., 2023; Li et al., 2024]
- However, building a generalist verifier is hard! [Sutton, 2001]

Synthetic data is scaling up

- AI models will be trained on increasing amount of model-generated data, inevitably.
  - Model builders intentionally add them.
  - Model-generated content populates on the Internet.



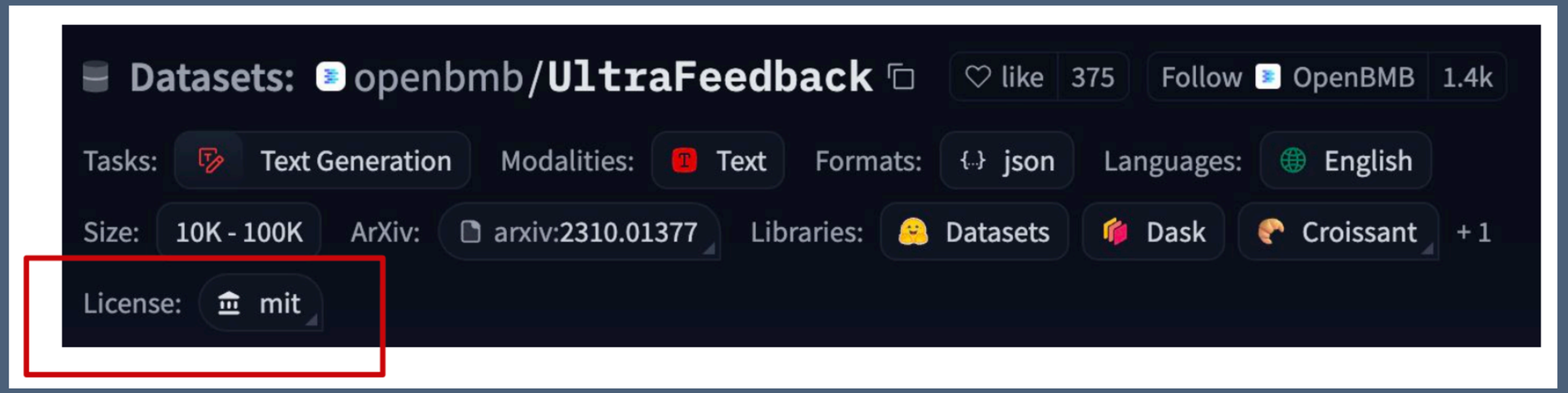
The cumulative size of text data from different sources for post-training LMs [Longpre et al., 2024]

Bringing in additional information

- Human selection, editing, & supervision.
- Human prior in the generation pipeline design.
  - E.g., prompts, principles for filtering, etc.
- Grounded documents, retrieved information, tool results.
- Rewards from interacting with environments.
- ...
- Can we synthesize new information that is useful? (i.e., new knowledge discovery)

The licensing of synthetic datasets is tricky

- Ultrafeedback as an example: A widely-used synthetic preference dataset with MIT license [Cui et al., 2024]



The licensing of synthetic datasets is tricky

The dataset might be transformed from **other less permissive datasets.**

Source datasets of Ultrafeedback	License
evol_instruct	MIT
false_qa	**Unclear**
flan	**Apache 2.0 only for generation code**
sharegpt	**Unclear**
truthful_qa	Apache 2.0
ultrachat	MIT

Distillation-friendly model

DeepSeek ToS

“ You may apply the Inputs and Outputs of the Services to a wide range of use cases, including personal use, academic research, derivative product development, **training other models (such as model distillation)**, etc. ”

Quoted as of 07/22/2025

Distillation-unfriendly models

OpenAI ToS	Anthropic ToS	Gemini API ToS
“ You may not use our Services for any illegal, harmful, or abusive activity. For example, you may not: … **Use output to develop models that compete with OpenAI.** ”	“ You may not access or use our services … **To develop any products or services that compete with our Services, including to develop or train any AI or ML algorithms or models or resell the Services.** ”	“ You may not use the Services **to develop models that compete with the Services** (e.g., Gemini API or Google AI Studio).”

Quoted as of 07/22/2025

Summary

- Synthetic data vs real-world data
  - Diversity & distribution: understand the richfulness of
  - Quality: develop quality checks/validation methods to promote high-quality data.
- The scaling of synthetic data
  - Bring in additional information in model self-improving, and
- Licensing & copyright
  - Call for lawful guidance on the use of synthetic data, and the community for technical innovation and ethical practice.

Practical Day 3

Coding Session

- **GOAL:** Explore LLM-based Active Learning Strategies
- Part B: LLM-based Generation
  - Annotate 50 movie reviews with positive or negative
  - Design a prompt for data annotation and compare the annotations

Coding Session

- **GOAL:** Explore LLM-based Active Learning Strategies
- Part C: LLM-based Selection
- Objective: Use LLMs as intelligent query strategies
    - Model: Qwen-2.5-0.5B/1.5B Instruct Key Experiments:
        - Prompt-based Instance Selection
    - Comparative Analysis:
        - LLM selections vs. traditional uncertainty sampling
        - LLM selections vs. diversity-based sampling
        - Prompt variation impact on selection quality
    - Selection Criteria Exploration:
        - Informativeness-based prompts
        - Diversity-based prompts
        - Difficulty-based prompts
        - Hybrid approaches

Coding Session

- **GOAL:** Explore LLM-based Active Learning Strategies
- Part D: LLM-based Generation
  - Objective: Generate synthetic training data
    - Conditional text generation for data augmentation
    - Style transfer and paraphrasing
    - Synthetic minority class generation
    - Quality assessment of generated samples