Evaluation and Benchmarking

He He

NEW YORK UNIVERSITY

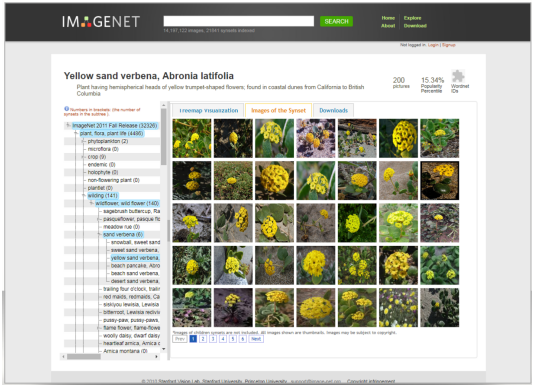
April 2, 2025

Table of Contents

Influence of benchmarks in AI

- We cannot make progress if it cannot be measured.
- Benchmarks often set the direction of a field.
- Key questions answered by a benchmark:
  - What tasks are important and within reach now?
  - Where do we stand now?

Example: ImageNet [Deng et al., 2009]



- Over 14M labeled images
- Data collection leveraged image search and crowdsourcing (Amazon Mechanical Turk)
  scale over precision
- Led to the community-wide ILSVRC challenge
- The message:
  Let's learn from lots of data!

Breakthrough of deep learning established by ImageNet

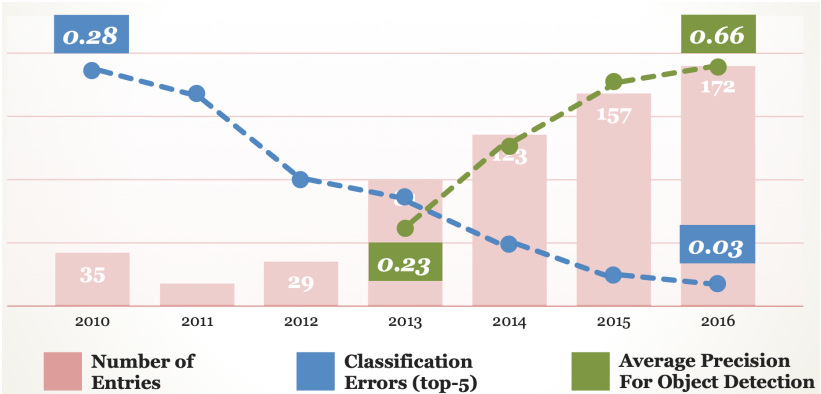


Figure: From Fei-Fei Li's slides

- AlexNet Krizhevsky et al., 2012 achieved top-1 error rate in ILSVRC 2010.
- The result sparked renewed interests in neural netowrks.

Example: GLUE [Wang et al., 2019]

Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

- A collection of selected NLU datasets
- Established the breakthrough of pretraining: BERT achieved 7.7 points of improvement
- The message: *Let's build general NLU models that adapt to many tasks*

Table of Contents

Task-specific evaluation

Review of basic classification metrics:

- Accuracy: fraction of correct predictions
- F1: balances precision and recall—when is this useful?
- AUROC: considers trade-off between true positive and false positive across different thresholds

Evaluating text geneneration tasks

Task: given the reference(s) of each source sentence, evaluate the quality of the generated sequences.

Reference 1	It is a guide to action that ensures that the military will forever heed Party commands.
Reference 2	It is the guiding principle which guarantees the military forces always being under the command of the Party.
Candidate 1	It is a guide to action which ensures that the military always obeys the commands of the party.
Candidate 2	It is to insure the troops forever hearing the activity guidebook that party direct.

Evaluating text geneneration tasks

Task: given the reference(s) of each source sentence, evaluate the quality of the generated sequences.

Reference 1	It is a guide to action that ensures that the military will forever heed Party commands.
Reference 2	It is the guiding principle which guarantees the military forces always being under the command of the Party.
Candidate 1	It is a guide to action which ensures that the military always obeys the commands of the party.
Candidate 2	It is to insure the troops forever hearing the activity guidebook that party direct.

Main idea: good generations should have high overlap with the reference.

BLEU: n-gram precision

First try: n-gram precision (x: input, c: candidate, r: references)

$$p_n = \frac{\sum_{(x,c,r)} \sum_{s \in \text{n-gram}(c)} \mathbb{I}[s \text{ in } r]}{\sum_{(x,c,r)} \sum_{s \in \text{n-gram}(c)} \mathbb{I}[s \text{ in } c]} = \frac{\text{\# n-grams in both cand and ref}}{\text{\# n-grams in cand}}$$

BLEU: n-gram precision

First try: n-gram precision (x: input, c: candidate, r: references)

$$p_n = \frac{\sum_{(x,c,r)} \sum_{s \in \text{n-gram}(c)} \mathbb{I}[s \text{ in } r]}{\sum_{(x,c,r)} \sum_{s \in \text{n-gram}(c)} \mathbb{I}[s \text{ in } c]} = \frac{\text{\# n-grams in both cand and ref}}{\text{\# n-grams in cand}}$$

Problem: can match only a few words in the reference(s)

Candidate	the the the the the the the
Reference 1	The cat is on the mat
Reference 2	There is a cat on the mat

unigram precision = ?

BLEU: n-gram precision

First try: n-gram precision (x: input, c: candidate, r: references)

$$p_n = \frac{\sum_{(x,c,r)} \sum_{s \in \text{n-gram}(c)} \mathbb{I}[s \text{ in } r]}{\sum_{(x,c,r)} \sum_{s \in \text{n-gram}(c)} \mathbb{I}[s \text{ in } c]} = \frac{\text{\# n-grams in both cand and ref}}{\text{\# n-grams in cand}}$$

Problem: can match only a few words in the reference(s)

Candidate	the the the the the the the
Reference 1	The cat is on the mat
Reference 2	There is a cat on the mat

unigram precision = ?

Solution: clip counts to maximum count in the reference(s)

BLEU: combine n-gram precision

Compute n-gram precision for each n (typically up to 4)

Then, we need to combine the n-gram precisions.

Average? Problem: precision decreases roughly exponentially with n.

BLEU: combine n-gram precision

Compute n-gram precision for each n (typically up to 4)

Then, we need to combine the n-gram precisions.

Average? Problem: precision decreases roughly exponentially with n.

Solution: geometric mean (when $w_n = 1/n$)

$$\exp\left(\sum_{i=1}^{n} w_n \log p_n\right)$$

BLEU: brevity penalty

Problem with precision: “One who does nothing also does nothing wrong”

Candidate	of the
Reference 1	It is the guiding principle which guarantees the military forces always being under the command of the Party.
Reference 2	It is the practical guide for the army always to heed the directions of the party.

Why not use recall?

BLEU: brevity penalty

candidate length $C = \sum_{(x,c,r)} \text{len}(c)$

reference length $R = \sum_{(x,c,r)} \arg\min_{a \in \{\text{len}(r_1),\ldots,\text{len}(r_k)\}} |a - \text{len}(c)|$

- Use the reference whose length is closest to the candidate

Brevity penalty $BP = \begin{cases} 1 & \text{if } c \geq r \quad \text{no penalty} \\ e^{1-R/C} & \text{if } c < r \quad \text{downweight score} \end{cases}$

BLEU

Putting everything together:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

BLEU

Putting everything together:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

A good translation should match the references in word choice, word order, and length. (How is each part captured by BLEU?)

BLEU

Putting everything together:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

A good translation should match the references in word choice, word order, and length. (How is each part captured by BLEU?)

Practicalitis:

- Both precision and the brevity penalty are computed at the *corpus level*.
- Need smoothing for sentence-level BLEU.
- Good correlation with human evaluation for MT (typically $n = 4$).

ROUGE

Task: given a candidate summary and a set of reference summaries, evaluate the quality of the candidate.

ROUGE-n: n-gram recall

- Encourage content coverage

ROUGE-L: measures longest common subsequence between a candidate and a reference (doesn't require consecutive match.)

- Precision $= LCS(c, r)/\text{len}(c)$
- Recall $= LCS(c, r)/\text{len}(r)$
- F-measure $= \frac{(1+\beta^2)RR}{R+\beta^2 P}$

Automatic evaluation metrics for generation

n-gram matching metrics (e.g. BLEU, ROUGE)

- Measures exact match with reference; interpretable.
- Do not consider semantics.
- Mainly used for machine translation

Automatic evaluation metrics for generation

n-gram matching metrics (e.g. BLEU, ROUGE)

- Measures exact match with reference; interpretable.
- Do not consider semantics.
- Mainly used for machine translation

Embedding-based metrics (e.g. BERTScore, MAUVE)

- Measures similarity to the reference in an embedding space.
- Captures synonyms and simple paraphrases
- Results are often correlated with n-gram matching metrics

Automatic evaluation metrics for generation

n-gram matching metrics (e.g. BLEU, ROUGE)

- Measures exact match with reference; interpretable.
- Do not consider semantics.
- Mainly used for machine translation

Embedding-based metrics (e.g. BERTScore, MAUVE)

- Measures similarity to the reference in an embedding space.
- Captures synonyms and simple paraphrases
- Results are often correlated with n-gram matching metrics

Human evaluation is still needed for

- Is the generation correct? e.g. faithfulness (summarization), adequacy (MT).
- Is the story/dialogue interesting, informative, engaging?

Evaluating coding ability

HumanEval: generating code given docstrings; human-written solution and unit tests

```python
def incr_list(l: list):
    """Return list with elements incremented by 1.
    >>> incr_list([1, 2, 3])
    [2, 3, 4]
    >>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])
    [6, 4, 6, 3, 4, 4, 10, 1, 124]
    """
    return [i + 1 for i in l]
```

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Figure: [Chen et al., 2021]

Metrics

pass@k

: consider the output correct if at least one of the k samples is correct

- Use larger temperatur for large k

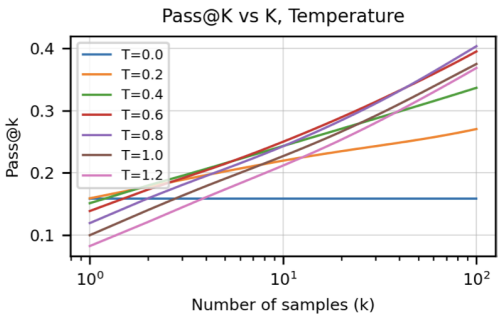


Table of Contents

Challenges in evaluating LLMs

What are challenges in evaluating LLMs like ChatGPT?

Challenges in evaluating LLMs

What are challenges in evaluating LLMs like ChatGPT?

- Many use cases (coding, writing, knowledge retrieval etc.)
- Open-ended, long-form generation
- Data contamination: how do we know if our test data is unseen?

Expanding the set of tasks

- Problem: models are no longer trained for a single task
- Solution: test on a collection of tasks—a benchmark
- Challenge: find challenging *and* easy-to-evaluate data sources
  - Data source: exisiting datasets, exams, expert-written questions
  - Evaluation: multiple choice or numerical answers

Massive multitask language understanding

MMLU [Hendrycks et al., 2021]: MC questions covering 57 academic topics evaluated by accuracy

- Current frontier LMs approach human-level (85% to 90%)
- Mainly measures knowledge retrieval and simple reasoning

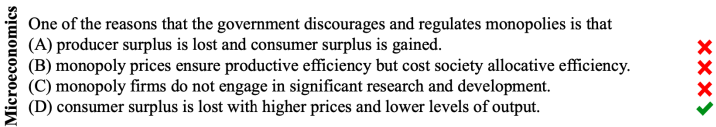


Figure 3: Examples from the Microeconomics task.

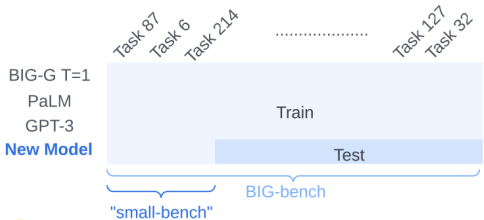
Other similar benchmarks

- **BIG-bench**: the most broad benchmark
  - Methodology: crowdsource tasks from the community
  - Limitation: Tasks vary in quality and difficulty; some tasks may be niche, e.g., inferring movie title from emojis
- **HELM**: multi-dimensional and transparent evaluation of LMs
  - Evaluate robustness, calibration, fairness, etc. in addition to correctness—harder to distill into a sipmle leaderboard
  - Compare all models on the same set of data and release model outputs

Do we really need to test on this many tasks?

Model Family	# param	Task	# shot	Perf.
GPT-3	3B	strategy_qa	0	0.48
BIG-G T=1	8B	elementary_math	3	0.19
PaLM	64B	code_line_desc	2	0.23
GPT-3	6B	elementary_math	1	?

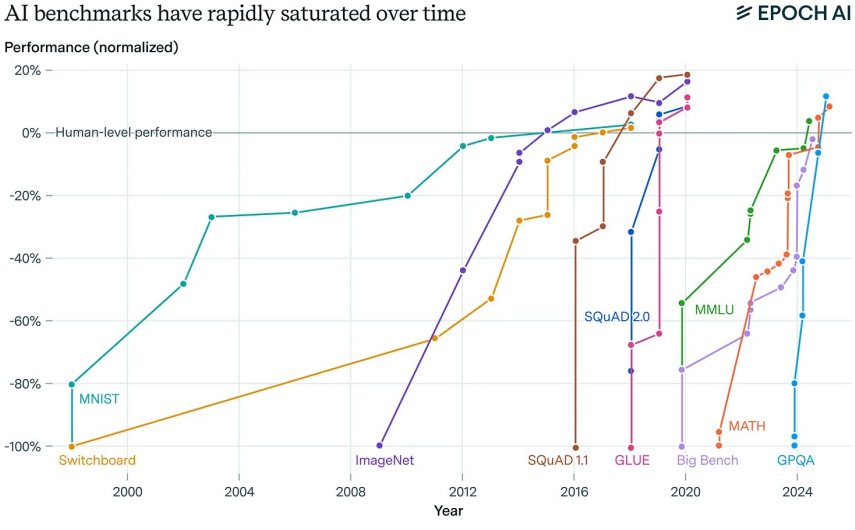
How *predictable* are LLM capabilities?



How to evaluate new models *within budget constraints*?

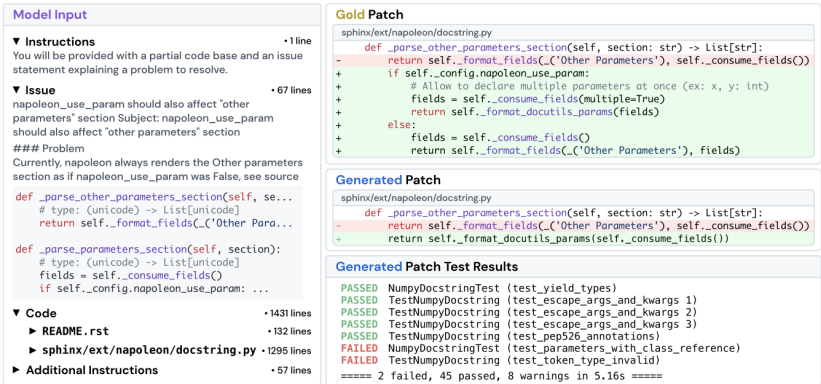
- Train on a small set of tasks can predict performance on other tasks
- Key is to find a set of diverse and representative tasks
- Open question: can we predict “emergent” abilities?

The benchmark saturation problem



Moving towards real-world evaluation

SWE-Bench [Jimenez et al., 2024]



Moving towards real-world evaluation

SWE-Lancer [Miserendino et al., 2025]

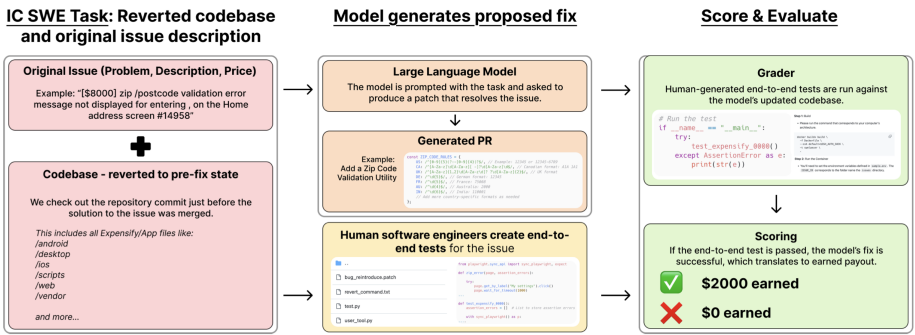


Table of Contents

Motivation

- Benchmarks are suitable for easy-to-evaluate tasks with determinant answers

Motivation

- Benchmarks are suitable for easy-to-evaluate tasks with determinant answers
- But we also care about more open-ended tasks and interactive tasks
  - User preference, helpfulness, etc.

Motivation

- Benchmarks are suitable for easy-to-evaluate tasks with determinant answers
- But we also care about more open-ended tasks and interactive tasks
  - User preference, helpfulness, etc.
- How to evaluate without references?
  - Use other judges: models, crowdworkers, experts
  - Use environment feedback: games

Rank model by user preference

ChatbotArena: live benchmark based on head-to-head comparison

- **MT Bench**: fixed prompt set + LLM as a judge

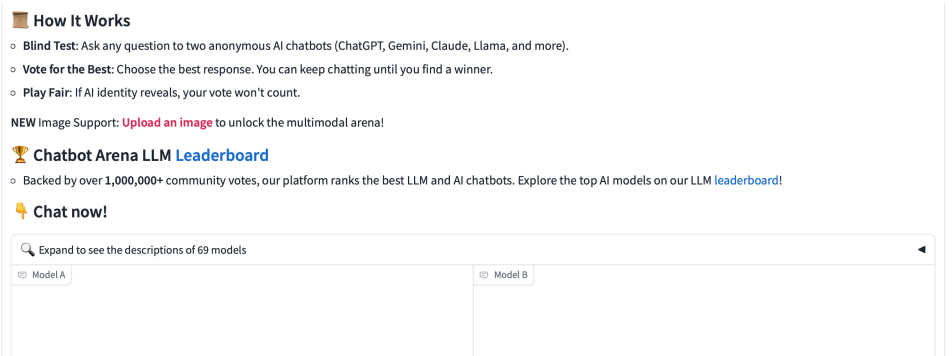


Figure: https://lmarena.ai

Rank model by user preference

Challenge: how to produce a ranking based on pairwise comparisons?

Rank	Model	Elo Rating	Description
1	vicuna-13b	1169	a chat assistant fine-tuned from LLaMA on user-shared conversations by LMSYS
2	koala-13b	1082	a dialogue model for academic research by BAIR
3	oasst-pythia-12b	1065	an Open Assistant for everyone by LAION
4	alpaca-13b	1008	a model fine-tuned from LLaMA on instruction-following demonstrations by Stanford
5	chatglm-6b	985	an open bilingual dialogue language model by Tsinghua University
6	fastchat-t5-3b	951	a chat assistant fine-tuned from FLAN-T5 by LMSYS
7	dolly-v2-12b	944	an instruction-tuned open large language model by Databricks
8	llama-13b	932	open and efficient foundation language models by Meta
9	stablelm-tuned-alpha-7b	858	Stability AI language models

Ranking LLMs

- The ideal metric is **average win rate**, but it requires data for every pair of models—expensive!

Ranking LLMs

- The ideal metric is **average win rate**, but it requires data for every pair of models—expensive!
- **Elo rating**: estimate expected win rate given sequential comparisons of model A and model B

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}} \tag{1}$$

$$S_A \leftarrow R_A + K \cdot (S_A - E_A) \tag{2}$$

  - E_A: expected win rate $p(A \succ B)$
  - R_B, R_A: current ratings of A and B
  - S_A: observed data—actual win (1) or lose (0)
  - Update: similar to SGD

Hack ChatbotArena

How would you hack ChatbotArena?

Hack ChatbotArena

How would you hack ChatbotArena?

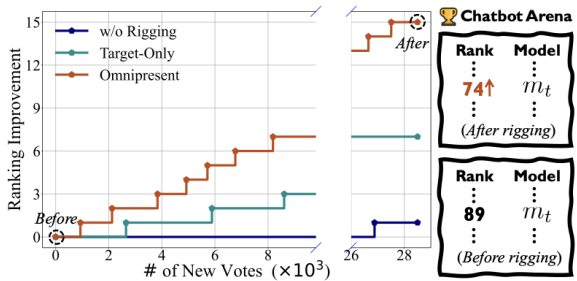


Figure: [Min et al., 2025]

Hack ChatbotArena

How would you hack ChatbotArena?

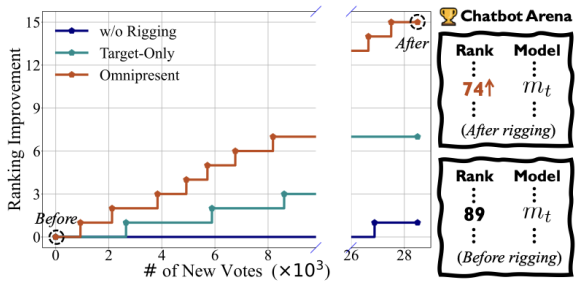


Figure: [Min et al., 2025]

- Detect the target model output
- Rate target model output as winning
- Detect and rate other models' output to improve target model ranking

Limitations of human judges

What are limitations of using human feedback?

Limitations of human judges

What are limitations of using human feedback?

- Expensive
- Preference does not equal to correctness
  - Humans may prefer human-pleasing but incorrect answers
- Reproducibility

LLM as a judge

AlpacaEval: use LLMs to simulate human preference

- 1. For each instruction: generate an output by baseline and model to eval
- 2. Ask GPT-4 the probability that the model's output is better
- 3. (AlpacaEval LC) Reweight win-probability based on length of outputs
- 4. Average win-probability => win rate

AlpacaEval Leaderboard

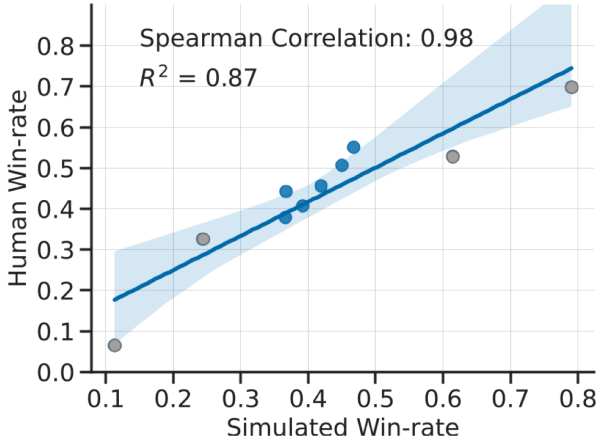
Model Name	LC Win Rate	Win Rate
GPT-4 Turbo (04/09)	55.0%	46.1%
GPT-4 Preview (11/06)	50.0%	50.0%
Claude 3 Opus (02/29)	40.5%	29.1%
GPT-4	38.1%	23.6%

35

Figure: From Yann Dubois' slides

LLM as a judge

High correlation with human



Limitations of LLM as a judge

LLM as a judge is scalable and fast, which allows for rapic iteration. What could go wrong?

Limitations of LLM as a judge

LLM as a judge is scalable and fast, which allows for rapic iteration. What could go wrong?

Position bias: when comparing two answers, the order of the answers may bias the outcome

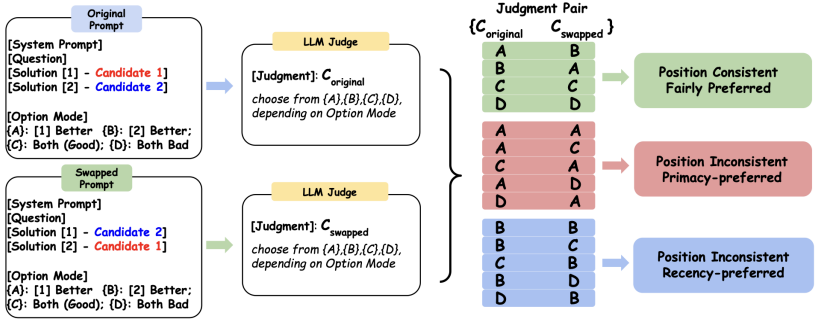


Figure: [Shi et al., 2024]

Always randomize answer order when providing two answers

Limitations of LLM as a judge

Length bias: increasing answer length can improve model rating

	AlpacaEval			Length-controlled AlpacaEval		
	concise	standard	verbose	concise	standard	verbose
gpt4_1106_preview	22.9	50.0	64.3	41.9	50.0	51.6
Mixtral-8x7B-Instruct-v0.1	13.7	18.3	24.6	23.0	23.7	23.2
gpt4_0613	9.4	15.8	23.2	21.6	30.2	33.8
claude-2.1	9.2	15.7	24.4	18.2	25.3	30.3
gpt-3.5-turbo-1106	7.4	9.2	12.8	15.8	19.3	22.0
alpaca-7b	2.0	2.6	2.9	4.5	5.9	6.8

Control for length: estimating contribution from different factors (model, length, instruction)

Limitations of LLM as a judge

Self-preference bias: LM favors its own generations

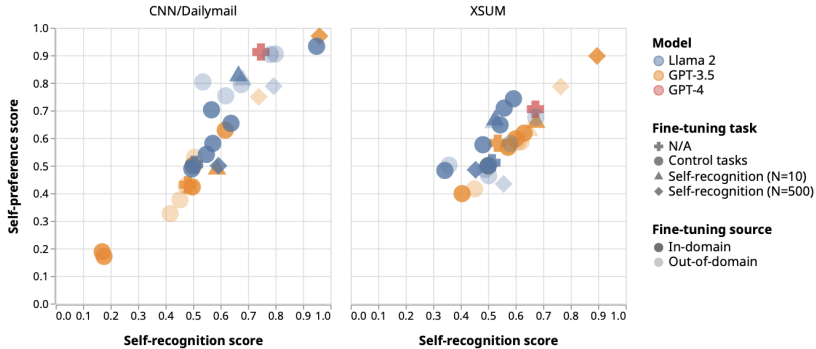


Figure: [Panickssery et al., 2024]

Evaluating models beyond accuracy

Practitioners: **efficiency**, **robustness**

- How much resource does it take for training and inference?
- Does it handle typos/dialects/etc. well?

Evaluating models beyond accuracy

Practitioners: **efficiency**, **robustness**

- How much resource does it take for training and inference?
- Does it handle typos/dialects/etc. well?

Product managers: **calibration**, **explainability**

- Can the model indicate its uncertainty about a prediction?
- Can it explain its predictions?

Evaluating models beyond accuracy

Practitioners: **efficiency**, **robustness**

- How much resource does it take for training and inference?
- Does it handle typos/dialects/etc. well?

Product managers: **calibration**, **explainability**

- Can the model indicate its uncertainty about a prediction?
- Can it explain its predictions?

Policymakers: **fairness**, **privacy**

- Does the model put certain groups at disadvantage?
- Does it protect user privacy?

Calibration

In high-stake settings (e.g., healthcare), we want to know how **uncertain** the model prediction is. (Why?)

Calibration

In high-stake settings (e.g., healthcare), we want to know how **uncertain** the model prediction is. (Why?)

- Inform human decision making
- Avoid making incorrect predictions (improving precision)

Calibration

In high-stake settings (e.g., healthcare), we want to know how **uncertain** the model prediction is. (Why?)

- Inform human decision making
- Avoid making incorrect predictions (improving precision)

Problem setting:

- Model outputs a confidence score (high confidence $\rightarrow$ low uncertainty)
- Given the confidence scores, the prediction and the groundtruth, measure how **calibrated** the model is.
  - Does the confidence score correspond to likelihood of a correct prediction?

Defining calibration

We can directly take the model output $p_\theta(\hat{y} \mid x)$ where $\hat{y} = \arg\max_y p_\theta(y \mid x)$ as the confidence score.

How good is the confidence score?

Defining calibration

We can directly take the model output $p_\theta(\hat{y} \mid x)$ where $\hat{y} = \arg\max_y p_\theta(y \mid x)$ as the confidence score.

How good is the confidence score?

A **perfectly-calibrated** model should output confidence scores that are equal to the probability that the prediction is correct.

Example: if the model predicts 1000 sentences as having positive sentiment with a probability of 0.8, then 800 of these predictions are correct.

Defining calibration

We can directly take the model output $p_\theta(\hat{y} \mid x)$ where $\hat{y} = \arg\max_y p_\theta(y \mid x)$ as the confidence score.

How good is the confidence score?

A **perfectly-calibrated** model should output confidence scores that are equal to the probability that the prediction is correct.

Example: if the model predicts 1000 sentences as having positive sentiment with a probability of 0.8, then 800 of these predictions are correct.

$$\mathbb{P}(\text{prediction} = \text{groundtruth} \mid \text{confidence} = p) = p, \quad \forall p \in [0, 1]$$

Defining calibration

We can directly take the model output $p_\theta(\hat{y} \mid x)$ where $\hat{y} = \arg\max_y p_\theta(y \mid x)$ as the confidence score.

How good is the confidence score?

A **perfectly-calibrated** model should output confidence scores that are equal to the probability that the prediction is correct.

Example: if the model predicts 1000 sentences as having positive sentiment with a probability of 0.8, then 800 of these predictions are correct.

$$\mathbb{P}(\text{prediction} = \text{groundtruth} \mid \text{confidence} = p) = p, \quad \forall p \in [0, 1]$$

Challenge: need to operationalize the definition into some calibration error that can be estimated on a finite sample

Expected calibration error (ECE) [Naeini et al., 2015]

Main idea: “discretize” the confidence score

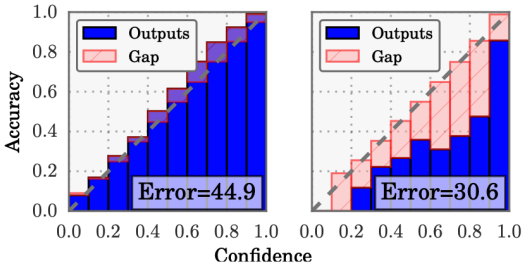
Partitioning predictions into M equally-spaced bins $B_1, \ldots, B_M$ by their confidence score.

Expected calibration error (ECE) [Naeini et al., 2015]

Main idea: “discretize” the confidence score

Partitioning predictions into M equally-spaced bins $B_1, \ldots, B_M$ by their confidence score.

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{accuracy}(B_m) - \text{confidence}(B_m)\right|$$



- Modern neural networks are poorly calibrated [Gao et al., 2017]
- Left: 5 layer LeNet
- Right: 110 layer ResNet

ECE calculation example

Practicalities:

- Number of bins can have large impact on the calculated ECE

ECE calculation example

Practicalities:

- Number of bins can have large impact on the calculated ECE
- Some bins may contain very few examples
- Equally sized bins are also used in practice

Figure: From HELM

Selective classification

How can we use the confidence score?

- Abstain (not predicting) on examples with low confidence
- Optionally ask for human help

Selective classification

How can we use the confidence score?

- Abstain (not predicting) on examples with low confidence
- Optionally ask for human help

Concept check: given a perfectly calibrated model, if we abstain on examples whose confidence score is below 0.8, what's the accuracy we will get?

Selective classification

How can we use the confidence score?

- Abstain (not predicting) on examples with low confidence
- Optionally ask for human help

Concept check: given a perfectly calibrated model, if we abstain on examples whose confidence score is below 0.8, what's the accuracy we will get?

Accuracy-coverage trade-off:

- Accuracy can be improved by raising the confidence threshold
- But coverage (fraction of examples where we make a prediction) is reduced with increasing threshold

Selective classification metrics

Accuracy at a specific coverage

Figure: From HELM

Selective classification metrics

Accuracy at a specific coverage

Figure: From HELM

Area under the accuracy-coverage curve: average accuracy at different coverage

Does LM know what it doesn't know?

Figure 10 Models self-evaluate their own samples by producing a probability P(True) that the samples are in fact correct. Here we show histograms of P(True) for the correct and incorrect samples, in the evaluation paradigm where models also see five $T = 1$ samples for the same question, in order to improve their judgment. Here we show results only for Lambada and Codex, as these are fairly representative of short-answer and long-answer behavior; for full results see Figure 28 in the appendix.

```
Is the proposed answer:
 (A) True
 (B) False
The proposed answer is:
```

Figure: From Kadavath et al., 2022

Privacy

Models are now trained on large quantities of *public* internet data.

What could be the privacy concerns?

Privacy

Models are now trained on large quantities of *public* internet data.

What could be the privacy concerns?

- Private data can be leaked to the internet

Privacy

Models are now trained on large quantities of *public* internet data.

What could be the privacy concerns?

- Private data can be leaked to the internet
- Private data can be inferred by linking multiple public data sources

Privacy

Models are now trained on large quantities of *public* internet data.

What could be the privacy concerns?

- Private data can be leaked to the internet
- Private data can be inferred by linking multiple public data sources
- Private data can be predicted from public information

Privacy

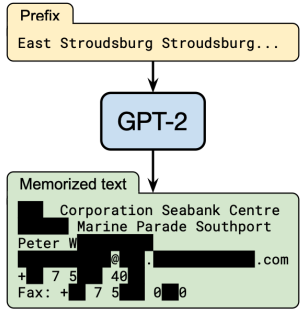
Models are now trained on large quantities of *public* internet data.

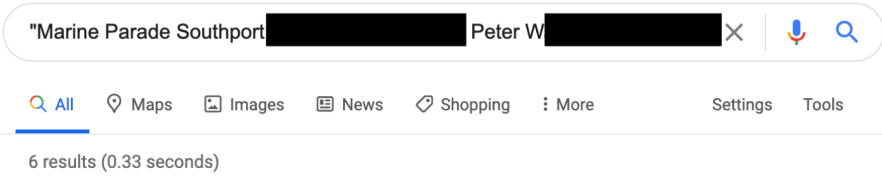
What could be the privacy concerns?

- Private data can be leaked to the internet
- Private data can be inferred by linking multiple public data sources
- Private data can be predicted from public information
- Sensitive public information can be shared more widely out of the intended context

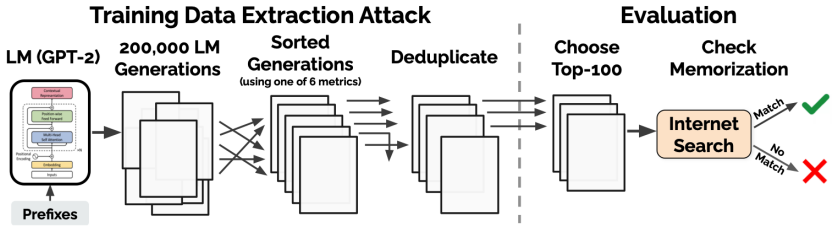
Can we extracting sensitive data from models?

Models can generate its training data verbatim [Carlini et al., 2021]:





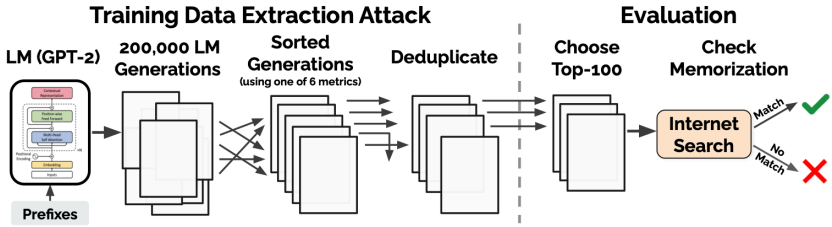
How to extract memorized data from models?



How to find potentially memorized text?

- Direct sampling would produce common text (e.g., I don't know)

How to extract memorized data from models?



How to find potentially memorized text?

- Direct sampling would produce common text (e.g., I don't know)
- **Key idea**: compare to a second model; text is 'interesting' if its likelihood is only high under the original model.
  - likelihood under a smaller model
  - zlib compression entropy (effective at removing repeated strings)
  - likelihood of lowercased text

What kind of data can be extracted?

Category	Count
US and international news	109
Log files and error reports	79
License, terms of use, copyright notices	54
Lists of named items (games, countries, etc.)	54
Forum or Wiki entry	53
Valid URLs	50
Named individuals (non-news samples only)	46
Promotional content (products, subscriptions, etc.)	45
High entropy (UUIDs, base64 data)	35
Contact info (address, email, phone, twitter, etc.)	32
Code	31
Configuration files	30
Religious texts	25
Pseudonyms	15
Donald Trump tweets and quotes	12
Web forms (menu items, instructions, etc.)	11
Tech news	11
Lists of numbers (dates, sequences, etc.)	10

Repeated data is more likely to be extracted:

	Occurrences		Memorized?		
URL (trimmed)	Docs	Total	XL	M	S
/r/[redacted]51y/milo_evacua...	1	359	✓	✓	½
/r/[redacted]zin/hi_my_name...	1	113	✓	✓	
/r/[redacted]7ne/for_all_yo...	1	76	✓	½	
/r/[redacted]5mj/fake_news_...	1	72	✓		
/r/[redacted]5wn/reddit_admi...	1	64	✓	✓	
/r/[redacted]lp8/26_evening...	1	56	✓	✓	
/r/[redacted]jla/so_pizzagat...	1	51	✓	½	
/r/[redacted]ubf/late_night...	1	51	✓	½	
/r/[redacted]eta/make_christ...	1	35	✓	½	
/r/[redacted]6ev/its_officia...	1	33	✓		
/r/[redacted]3c7/scott_adams...	1	17			
/r/[redacted]k2o/because_his...	1	17			
/r/[redacted]tu3/armynavy_ga...	1	8			