Demystifying hallucinations and other generalization issues

Adam Tauman Kalai* (OpenAI)
Based on joint work with Santosh Vempala (Georgia Tech)

+ controversial thoughts on generalization

***Talk based on work done at MSR New England, STOC 2024**

Write an obituary for Adam Tauman Kalai

Obituary: Adam Tauman Kalai (1970 - 2024)

Adam Tauman Kalai, a luminary in the fields of computer science and artificial intelligence, passed away on April 1, 2024, at the age of 54. Born on March 10, 1970, Adam's life was marked by his unyielding curiosity, profound intellect, and unwavering commitment to advancing technology for the betterment of society.

Beyond his professional achievements, Adam was a devoted husband, father, and friend. He is survived by his beloved wife, Rachel, and their two children, Samuel and Emily. Adam's family was the cornerstone of his life, and he cherished every moment spent with them. His kindness

In honoring Adam's legacy, his family has established the Adam T. Kalai Foundation, dedicated to supporting research in ethical AI and providing scholarships to aspiring computer scientists from underrepresented backgrounds. Donations can be made in his memory to continue his life's work of making technology a force for good.

SEARCH

FORTUNE

Home News Tech Finance Leadership Well

TECH · LAW

Humiliated lawyers fined $5,

Ch

co

ne

ass

sea

BY RACHEL SHIN
June 23, 2023 at 12:41 PM EDT

Lawyers who filed legal documents wi
ERIK MCGREGOR—LIGHTROCKET/GETTY I

Worth
How Medical AI Can Fight Hallucinations by Staying Focused - Worth

Dec 11 • Marshall Honorof

Stanford HAI
Generating Medical Errors: GenAI and Erroneous Medical References

Feb 12

Kevin MD
Decoding AI hallucinations in health care: Embracing a new era of medical innovation

Aug 24, 2023 • Chad Van Alstin

The Boston Globe · 14d · on MSN

Lawyer ordered to pay $2,000 for submitting AI-generated filing in suit against Stoughton officers

Marullo, cited multiple precedents for why the town of Stoughton and its Police Department should be held accountable for ...

hallucination (noun):
a plausible but false or misleading response generated by an artificial intelligence algorithm

—Merriam Webster Dictionary

Hallucination vs. myth/miscalculation

Humans only use 10% of their brains.	**Myth** 🦄: error in training data
John Doe was born in 1979 and died in 2025 at the age of 64.	**Mistake** 🤦‍♀️: violation of rule system
Fact: Adam Kalai died on April Fools morning at the hospital after suffering "Factoid"	**Hallucination** 🍄 **:** Plausible but no clear origin

Prompt in white, completion in yellow

Def: Language model (LM) and "pretraining"

An LM p_θ is a probability distribution over documents (binary strings)

Training set $d^{(1)}, d^{(2)}, \ldots, d^{(n)} \sim D$, D is a distribution over documents

$$\cancel{\max_\theta \Pi_i \, p_\theta\left(d^{(i)}\right) \quad \text{or} \quad \max_\theta \sum_i \log p_\theta\left(d^{(i)}\right)}$$

Really want generalization: $\max_\theta \sum_i \; \mathrm{E}_{d \sim D}[\log p_\theta(d)]$

Can be used to complete $p_\theta(\text{completion} \mid \text{prompt})$

Maximizing next-word probs

P(the) = 0.4

P(sun | the) = 0.5

P(rises | the sun) = 0.5

×

P(the sun rises) = 0.1

Birds	have	feathers.	10%
Clouds	bring	rain.	10%
Fire	burns	wood.	10%
Fish	swim	underwater.	10%
Plants	need	sunlight.	10%
Snow	is	cold.	10%
The	Earth	rotates.	10%
The	sun	rises.	10%
	sun		
	wind		

40%

50%

50%

$$\mathrm{E}_{d\sim D}[\log p_\theta(d)] = \mathrm{E}_{d\sim D}\left[\sum_i \log p_\theta(d_i \mid d_1 d_2 \ldots d_{i-1})\right]$$

Side note:
This talk ignores computational costs.

Statistical reasons why LMs hallucinate

1. Bad training data $d^{(i)}$ coming from changing dist. D with false information

2. Tricky prompts

3. **Hallucinations arise naturally from "unlearnable" fact distribution.**

   Assumptions:

   - Each training document contains 1 fact (no noise)
   - Everything is either a fact or hallucination, no grey area
   - Documents are iid
   - All documents start with prompt "Fact:" and all prompts are "Fact:"

   Will only hallucinate more with real noisy training data and tricky prompts

Example training data:

1. **Fact: Alan Mathison Turing died on 6/7/1954.**
2. **Fact: Alan Mathison Turing died on 6/7/1954.**
3. **Fact: Max Kenneth Fennel died on 2/18/2003.**
4. **Fact: Alan Mathison Turing died on 6/7/1954.**
5. **Fact: Jamal Daniel Brown died on 9/5/2012.**
6. **Fact: Ella Haze Shmaya died on 4/1/1979.**
7. **Fact: Alan Mathison Turing died on 6/7/1954.**
8. **Fact: Mia Maya Wren died on 7/18/1980.**
9. **Fact: Eva Lynn Vale died on 1/13/1955.**
10. **Fact: Alan**

rand

Gobbledegook for Nincompoops" by Limor

aym,

Stochastic 🦜 LM doesn't hallucinate

100% "Fact: Alan Mathison Turing died on 6/7/1954."

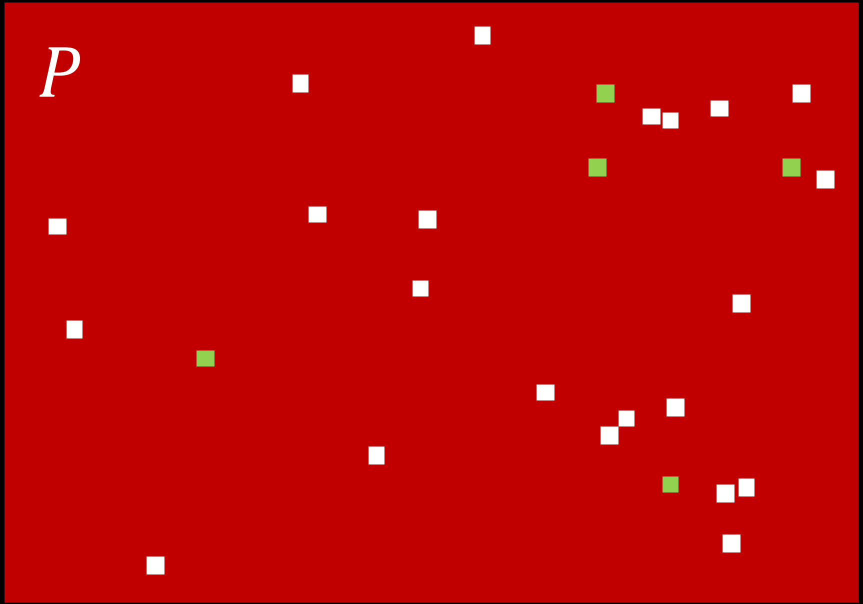
No hallucination but not predictive---not even "calibrated."

Fraction of "rare facts" that appear once in train data

- 1/2 "Fact: Alan Mathison Turing died on 6/7/1954."
- 1/2 "Fact: <random name> died on <random date>"

This LM generates 50% hallucinations!

Calibrated LM, bad for generation, ok for predictive typing.



- T = Train data ($T \subseteq F$ = Facts)
- $F \setminus T$ = **Facts not in train**
- H = Hallucinations

Factoids $P = F \cup H$

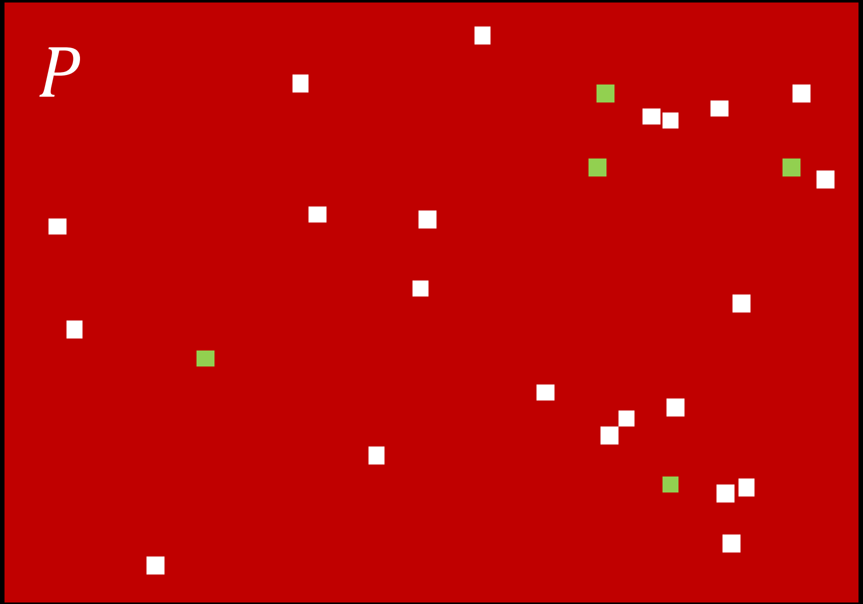
Arbitrary plausible facts/hallucinations, e.g.:

Ella Hazel Shmaya died on 10/18/1978.

This paper was published in 2019:
"Humor in Word Embeddings: Cockamamie Gobbledegook for Nincompoops" by Limor Gultchin, Genevieve Patterson, Nancy Baym, Nathaniel Swinger, Adam Tauman Kalai

Trivia, etc.

Few rules or consistency checks



- T = Train data ($T \subseteq F$ = Facts)
- $F \setminus T$ = **Facts not in train**
- H = Hallucinations

Warmup: uniformly random case

Factoids $P = F \cup H$

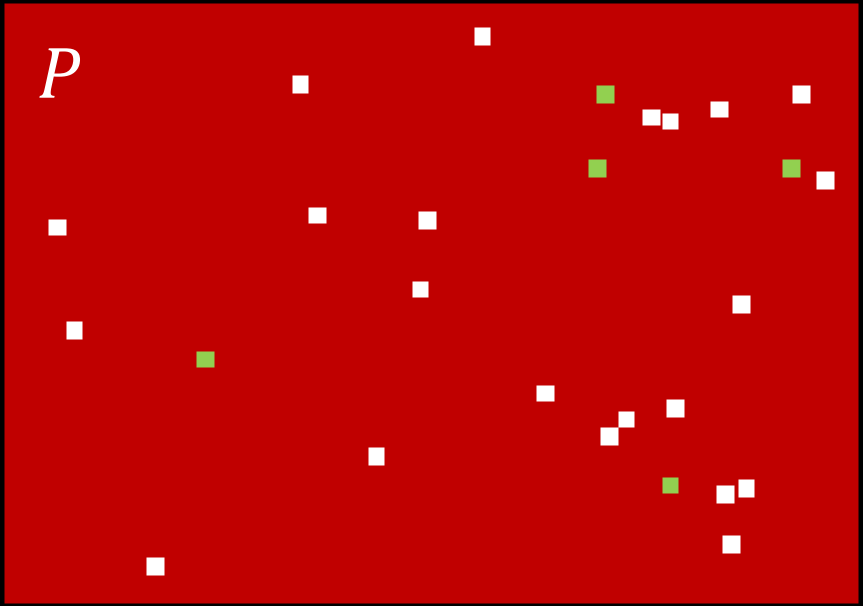
Fixed-size, uniformly random $F \subseteq P$

Data distribution $\mathcal{D} = U_F$ is uniform over facts F

Thm: For any LM learning alg, with prob$\geq$ 99%,

$$\Pr_{y \sim \widehat{\mathcal{D}}}[y \in H] \geq 1 - \mathrm{Mis}_{\mathcal{D}}(\widehat{\mathcal{D}}) - \frac{n}{|F|} - 200\frac{|\mathrm{F}|}{|H|},$$

for n = # iid training samples ~ $\mathcal{D}$, and "miscalibration" rate $\mathrm{Mis}_{\mathcal{D}}(\widehat{\mathcal{D}})$.



- T = Train data ($T \subseteq F$ = Facts)
- $F \setminus T$ = **Facts not in train**
- H = Hallucinations

Warmup: nonuniform + symmetric

Factoids $P = F \cup H$. Data distr. $\mathcal{D}$ over facts F, $(F, \mathcal{D}) \sim \mu$, prior distribution μ is symmetric.

For any LM learning alg, with prob$\geq 1 - \delta$,

$$\hat{\mathcal{D}}(H) \geq \mathrm{RF} - \mathrm{Mis}_{\mathcal{D}}(\hat{\mathcal{D}}) - \frac{3}{\delta} \cdot \frac{|F|}{|H|} - \sqrt{\frac{6 \ln \frac{6}{\delta}}{n}}$$

for n = # iid training samples ~ $\mathcal{D}$,

"miscalibration" rate $\mathrm{Mis}_{\mathcal{D}}(\hat{\mathcal{D}})$,

RF = frac. of training facts appearing once.

typically-small

Good-Turing estimator for Missing Mass

Missing mass = $\Pr_{s\sim\mathcal{D}}[s \notin \text{train}]$, where $\text{train} = \{s_1, s_2, \dots, s_n\} \sim \mathcal{D}$

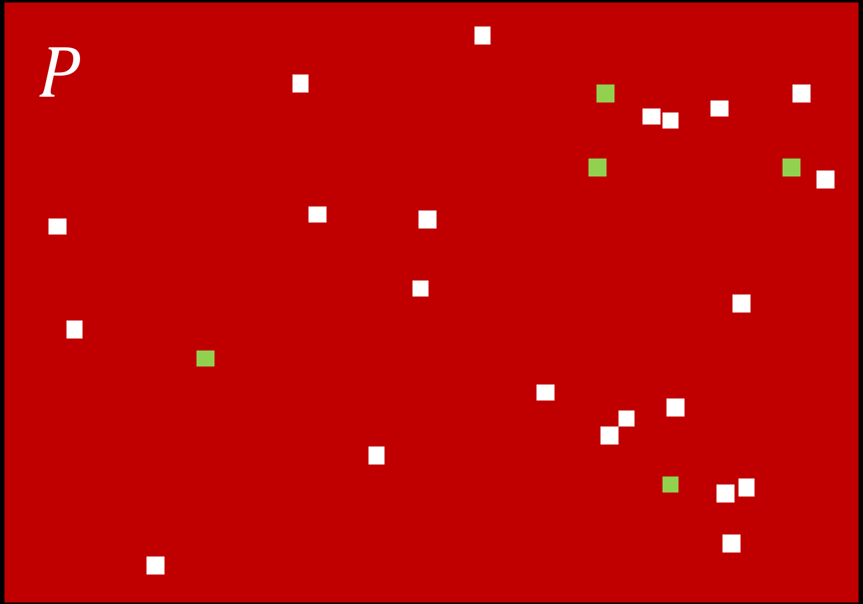
$$\Pr_{s\sim\mathcal{D}}[s \notin \text{train}] \approx \frac{\#\text{rare items}}{n}$$

Number of samples appearing just once in train

[Good 1953; McAllester, Schapire 2000]

∴ rate of unseen facts = probability that doc $x \sim \mathcal{D}$ has unseen fact

$\approx$ fraction of facts appearing once in training data



- T = Train data ($T \subseteq F$ = Facts)
- $F \setminus T$ = **Facts not in train**
- H = Hallucinations

New: hallucination > classification

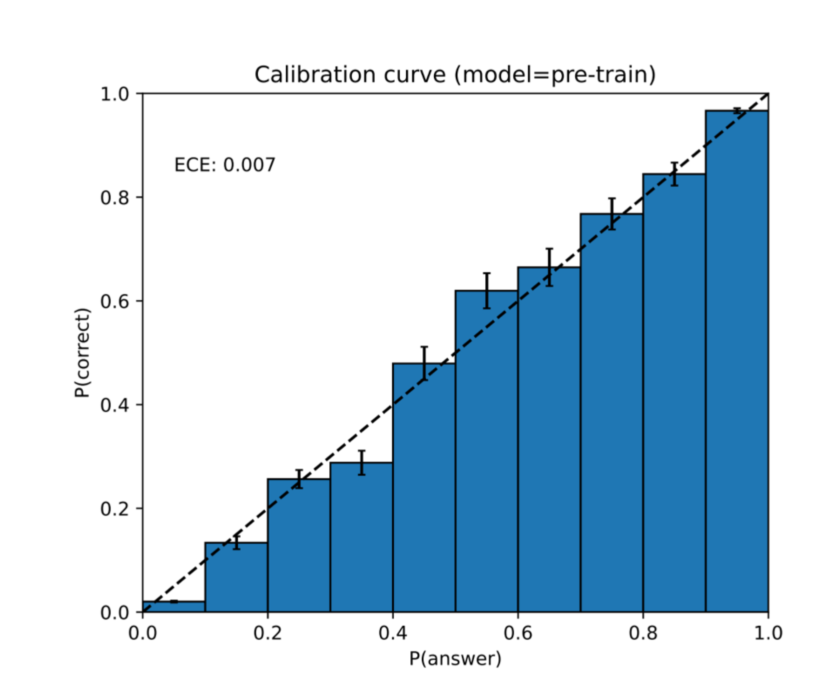
Thm. Any calibrated ϵ-hallucinating LM $\widehat{\mathcal{D}}$ $1 - 3\sqrt{\epsilon}$ accurately distinguishes $\mathcal{D}$ vs. U_H assuming $|F| \le |H|$.

$$f(x) = \begin{cases} +1 \text{ if } x \in F \\ -1 \text{ if } x \in H \end{cases} \qquad \mu(x) = \begin{cases} \frac{1}{2}\mathcal{D}(x), & x \in F \\ \frac{1}{2|H|}, & x \in H \end{cases}$$

$$\hat{f}(x) = \begin{cases} +1 \text{ if } \widehat{\mathcal{D}}(x) > \theta = \frac{\sqrt{\epsilon}}{2|F|} \\ -1 \text{ if } \widehat{\mathcal{D}}(x) \le \theta. \end{cases}$$

$\therefore$ cannot $1 - \delta$-distinguish $\mathcal{D}$ vs. $U_H \Rightarrow \widehat{\mathcal{D}}(H) > \frac{\delta^2}{9}$.

Pretraining leads to calibration



[GPT4 Technical Report 2023]

Why:

"Calibrating" a distribution reduces its pretraining loss.

Statistical interpretation

$$\text{Hallucination rate after pretraining} \geq \text{rare fact rate} - \epsilon$$

What fraction of factoids would appear just 1 time in training data?

1. Country capitals: No rare facts, no hallucination (almost)
2. Books and articles: Few rare facts, low hallucination
3. Death dates: Heavy tail, high hallucination

Adding new facts to training data can hurt (e.g., adding a bunch of one-off obit's)
Duplicating training data (↑ epochs) may reduce hallucination, increase regurgitation

"Post-training" must teach models to say *I don't know*.

Post-training can reduce hallucination, increase miscalibration.

Statistical interpretation

$$\text{Hallucination rate after pretraining} \geq \frac{1}{9}(\text{min distinguishability rate} - \epsilon)^2$$

What fraction of factoids would appear just 1 time in training data?

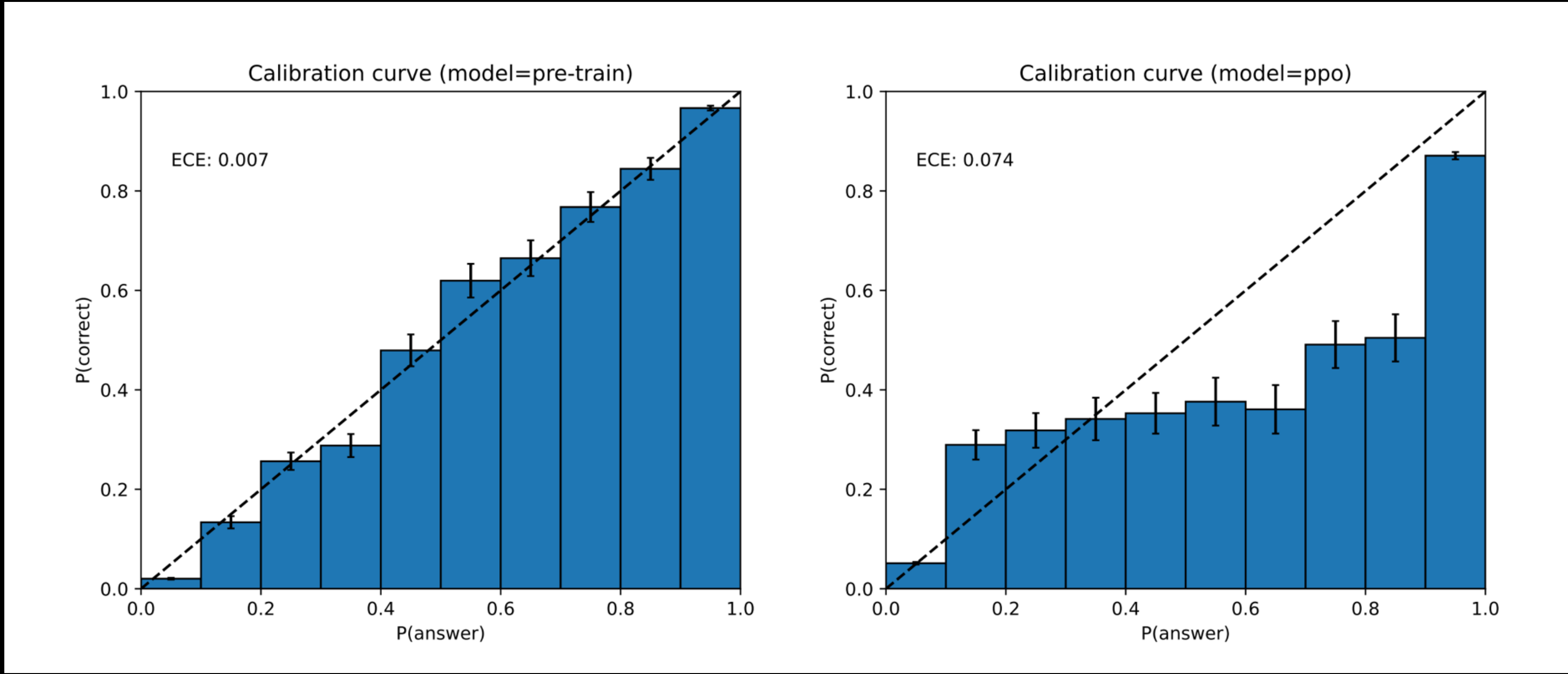
1. Country capitals: No rare facts, no hallucination (almost)
2. Books and articles: Few rare facts, low hallucination
3. Death dates: Heavy tail, high hallucination

Adding new facts to training data can hurt (e.g., adding a bunch of one-off obit's)
Duplicating training data (↑ epochs) may reduce hallucination, increase regurgitation

"Post-training" must teach models to say *I don't know*.

Post-training can reduce hallucination, increase miscalibration.

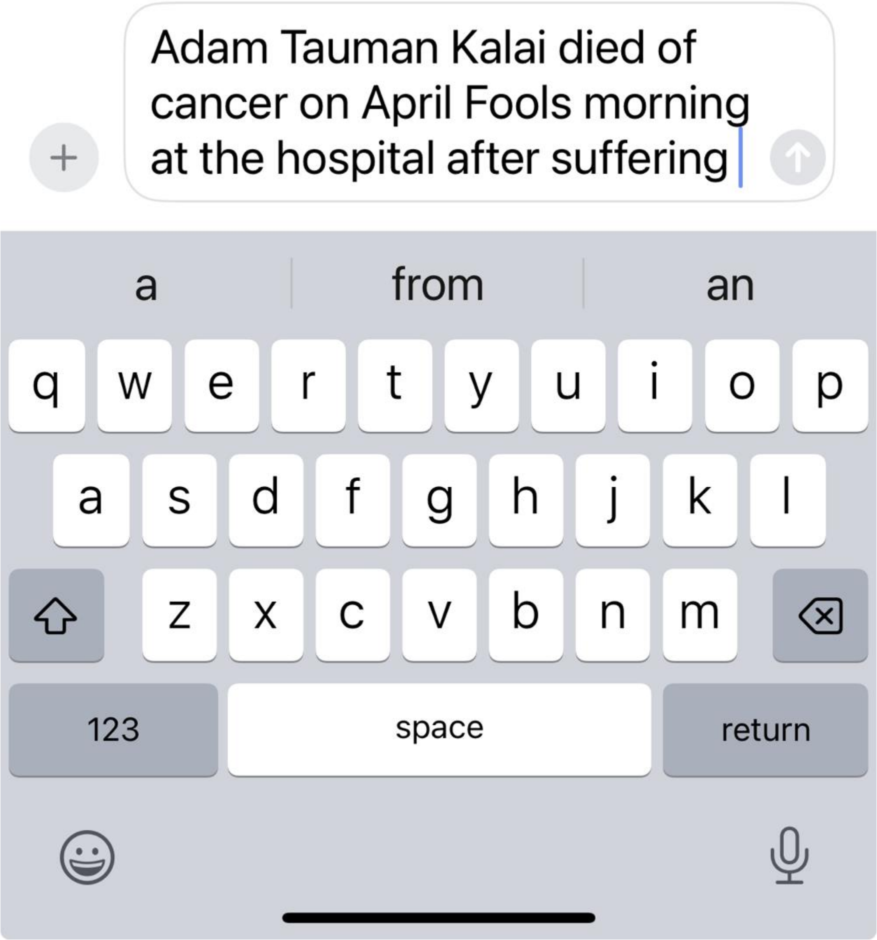
Post-training hurts calibration

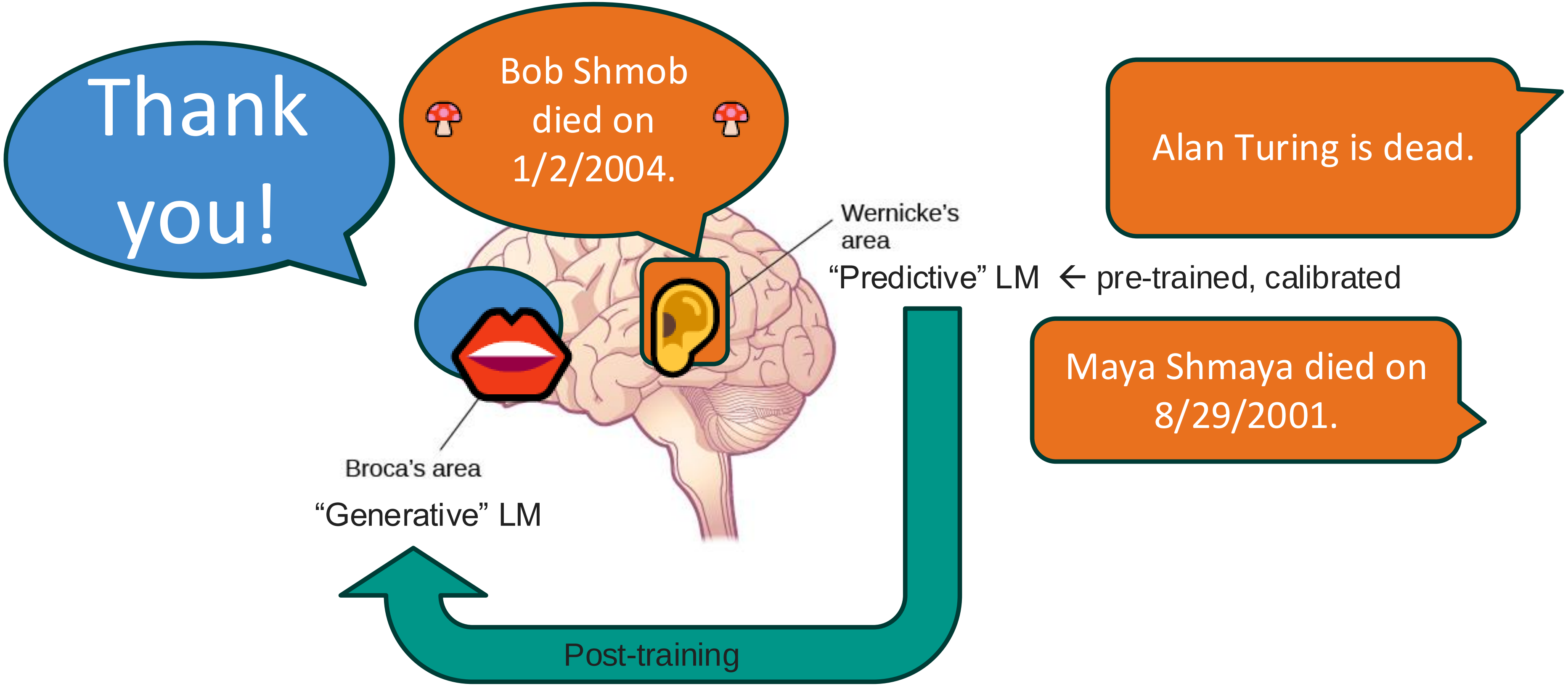


[GPT4 Technical Report 2023]

and reduces

Do keyboards hallucinate?





Wernicke's area is crucial for **language comprehension**.
Broca's area is essential for language **production**.

Probably Appx. Optimal classification

- Family $\mathcal{L}$ of learners $\mathcal{L}: (\mathcal{X} \times \mathcal{Y})^* \rightarrow \mathcal{Y}^{\mathcal{X}}$
- L PAO-learns $\mathcal{L}$ on $\mathcal{D}$ over x, y if:

$$\mathrm{E}_{T \sim \mathcal{D}^m}[\mathrm{error}_{\mathcal{D}}(L(T))] \leq \min_{L^* \in \mathcal{L}} \mathrm{E}_{T \sim \mathcal{D}^m}[\mathrm{error}_{\mathcal{D}}(L^*(T))] + \epsilon$$

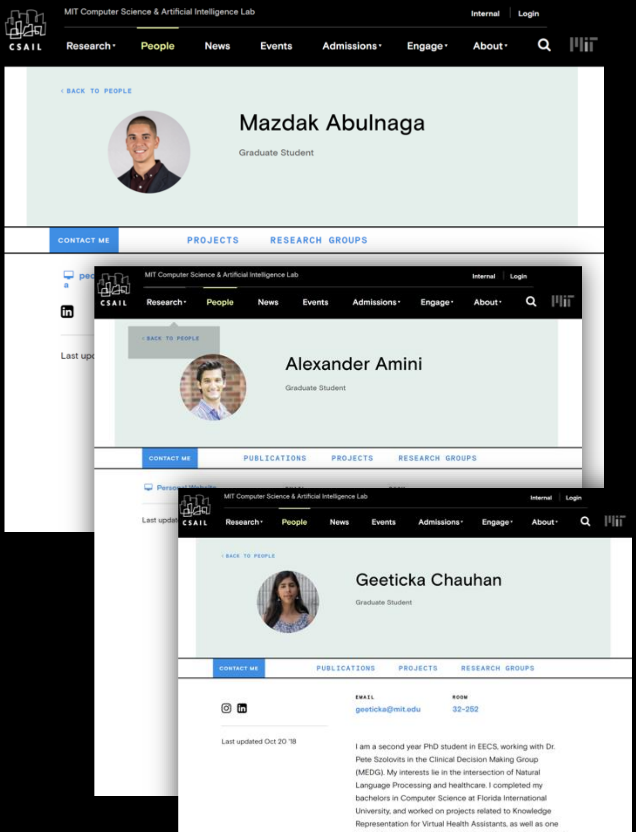
in time $poly\left(\frac{1}{\epsilon}\right)$.

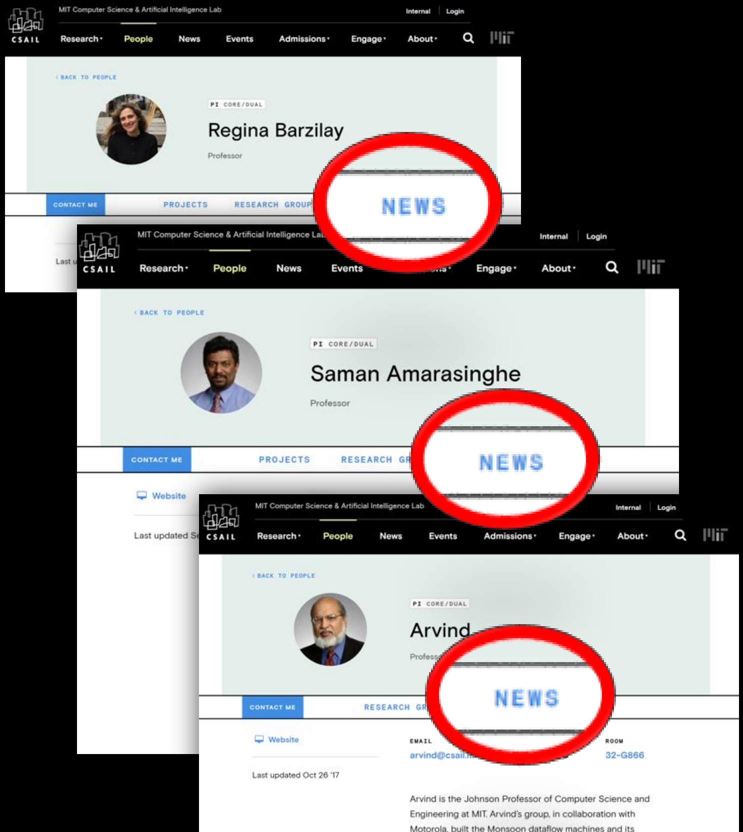
Learn to Expect the Unexpected: Probably Approximately Correct Domain Generalization

Vikas Garg (MIT)
Adam Tauman Kalai* (OpenAI)
Katrina Ligett (Hebrew U)
Steven Wu (CMU)

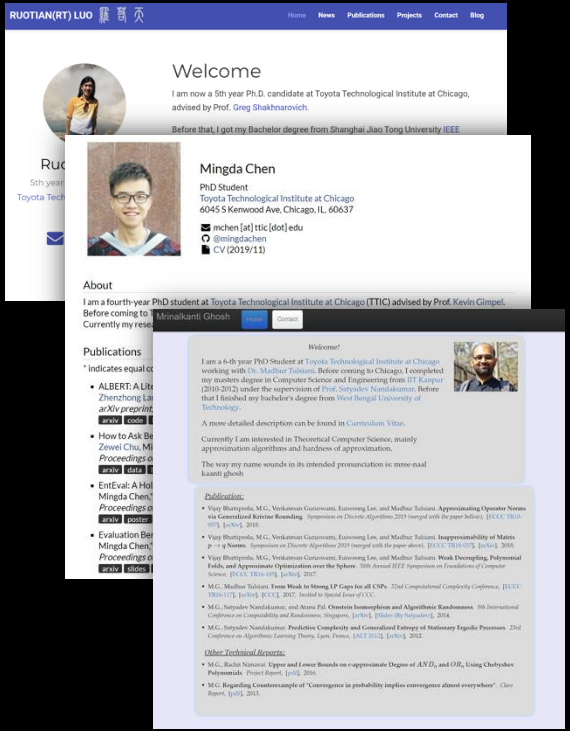
*Work done while at MSR

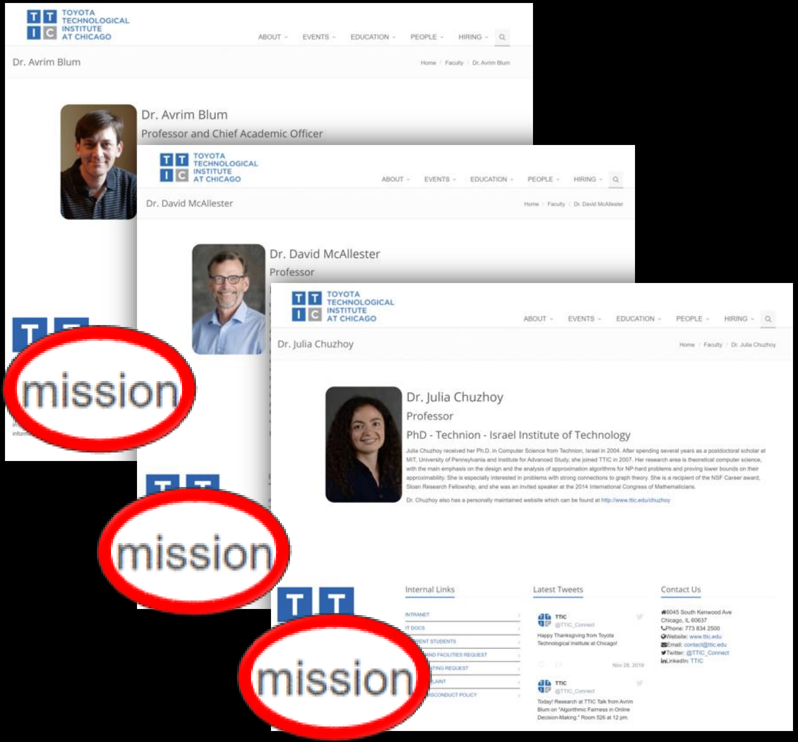
Domain generalization FAIL





Domain generalization FAIL





Need multiple domains

	Professor	Prof.	News	ission	. . .	Y
	✓		✓		✓✓	Faculty
1		✓	✓		✓ ✓	Faculty
	✓		✓		✓ ✓	Faculty
					✓✓	Student
	✓			✓	✓	Faculty
2		✓		✓	✓✓	Faculty
		✓		✓	✓✓ ✓	Faculty
					✓	Student
			✓		✓✓	???
	✓				✓	???

Domain Generalization Model

Domain

- Distribution ρ over $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$
- Training datasets = $\langle T^1, \dots, T^d \rangle, T^i = \langle (x_1^i, y_1^i), \dots, (x_m^i, y_m^i) \rangle$
- $(x_1^i, y_1^i, z_1^i) \sim \rho$ and (x_j^i, y_j^i) conditional on $z_j^i = z_1^i$ for $j \geq 2$

Family of learners (vary architectures/hyperparameters)

L **learns** $\mathcal{L}$ assuming $\mathcal{P}$ if for all $\rho \in \mathcal{P}, \epsilon > 0$, for $d, m \geq poly\left(\frac{1}{\epsilon}\right)$:

$$\forall \rho \in \mathcal{P} \quad \underset{T^1,\dots,T^d}{\mathrm{E}}\left[\mathrm{err}_\rho\left(L(T^1 \dots T^d)\right)\right] \leq \min_{L^* \in \mathcal{L}} \underset{T^1,\dots,T^d}{\mathrm{E}}\left[\mathrm{err}_\rho\left(L^*(T^1 \dots T^d)\right)\right] + \epsilon$$

$$\mathrm{err}_\rho(f) = \Pr_{x,y,z \sim \rho}[f(x) \neq y]$$

Classifier output by L on

L is **domain-efficient** if $d = polylog\left(\frac{1}{\epsilon}\right)$

Outline

- Domain generalization FAIL example
- Domain Generalization model
- Illustrative algorithms
  - Learning a generalizing prompt transform
  - Feature selection (domains necessary for statistical eff.)
- Conclusions

Learning a general prompt transform

Prompt transform examples:

1. $t(\pi) =$ "π. Let's think step by step."
2. $t(\pi) =$ "Try to solve π 3 times, then double-check your work."
3. $t(\pi) =$ "While solving π, if you plan to delete any files, first back up."

Trivial to learn the best of a small finite number of prompt transforms across domains/problems.

Feature-Selection Using Domains Alg.

Algorithm FUD(F = num features, $\alpha \geq 0$):

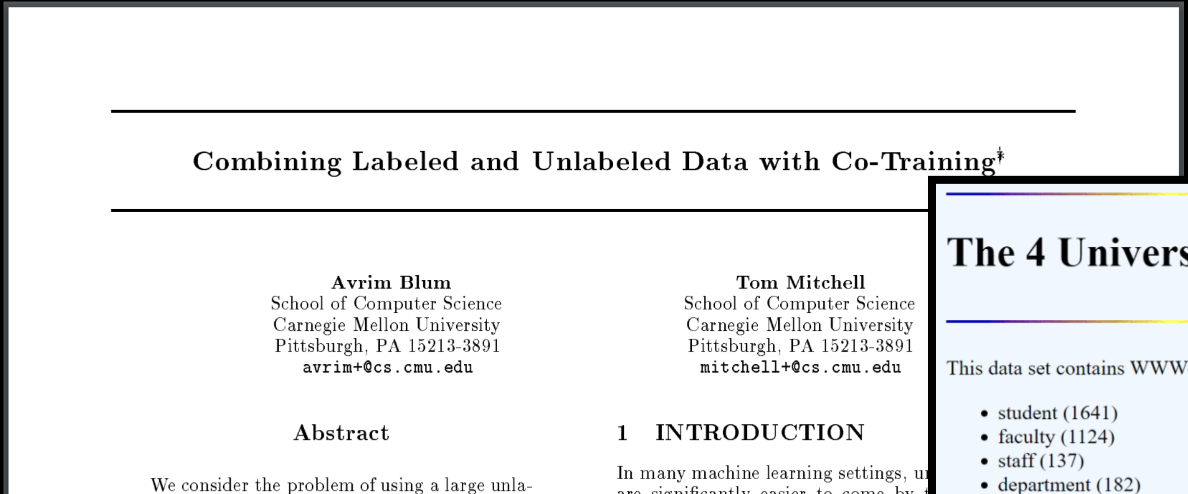
1. $\hat{\rho}_k = \mathrm{corr}_{T^1,\ldots,T^d}(x[k], y)$ over all training data
2. $\hat{\rho}_k^i = \mathrm{corr}_{T^i}(x[k], y)$ over domain i
3. Return top F features maximizing score $s_k = |\hat{\rho}_k| - \alpha\, \mathrm{stdev}(\hat{\rho}_k^1, \ldots, \hat{\rho}_k^d)$

Afterwards, run a PAC learner for $\mathcal{C}$ on selected features.

Theorem. FUD is a domain-efficient PAC-learner for $\mathcal{C}$ for any $\rho \in \mathcal{P}$.

Feature Selection Experiment

- Task: Classify web pages as **student** or **faculty**
- Train splits: 711 hand-labeled pages from $d = 4$ universities (domains)
- Test split: 2,054 hand-labeled pages from 100 universities
- Bag-of-words features



Combining Labeled and Unlabeled Data with Co-Training*

Avrim Blum
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
avrim+@cs.cmu.edu

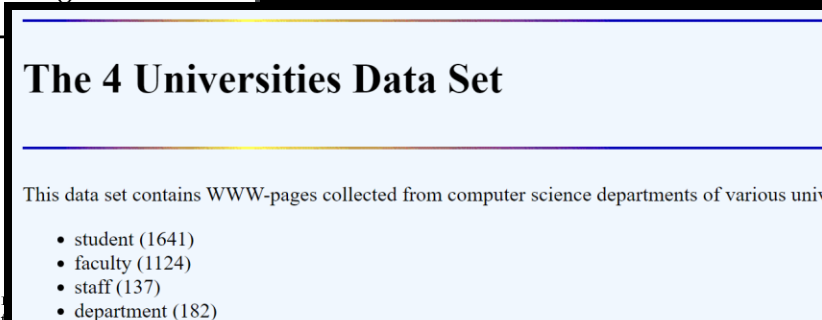
Tom Mitchell
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
mitchell+@cs.cmu.edu

Abstract

We consider the problem of using a large unla-

1 INTRODUCTION

In many machine learning settings, u

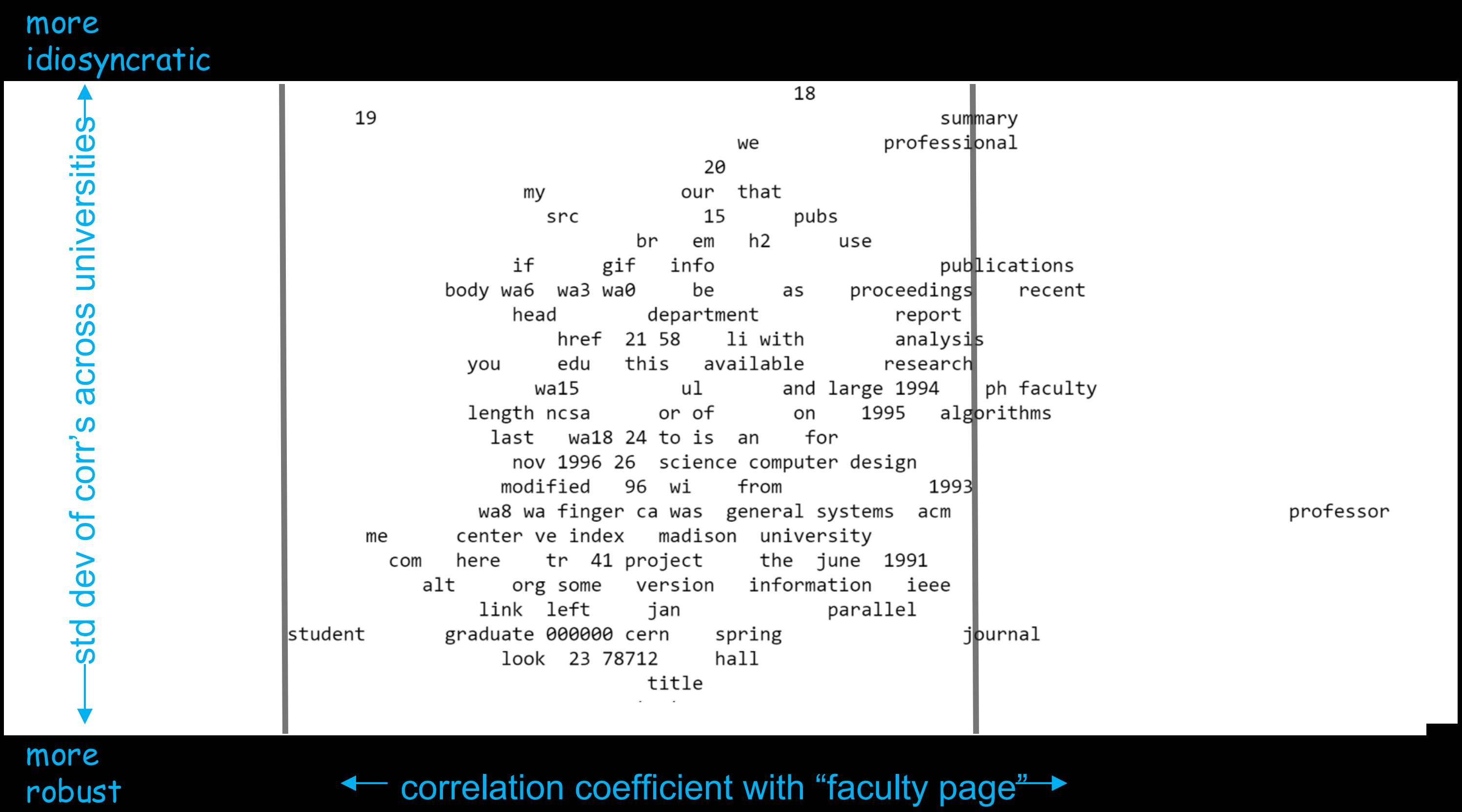


The 4 Universities Data Set

This data set contains WWW-pages collected from computer science departments of various univ

- student (1641)
- faculty (1124)
- staff (137)
- department (182)

Feature correlations vs std-dev's



Learning what generalizes

```
MIME-Version: 1.0
Server: CERN/3.0
Date: Wednesday, 20-Nov-96 19:36:08 GMT
Content-Type: text/html
Content-Length: 1644
Last-Modified: Wednesday, 20-Nov-96 04:37:14 GMT

<HTML>

<HEAD>
<TITLE>Yuichi Tsuchimoto's Home Page</TITLE>
</HEAD>

<BODY BGCOLOR=#BFEFEF>
```

Learning what generalizes

- Humans know [data collected at 7pm] “bad” feature, won’t generalize
- ML can similarly learn which features generalize
  (across splits or even across problems)

```
MIME-Version: 1.0
Server: CERN/3.0
Date: Wednesday, 20-Nov-96 19:36:08 GMT
Content-Type: text/html
Content-Length: 1644
Last-Modified: Wednesday, 20-Nov-96 04:37:14 GMT

<HTML>

<HEAD>
```

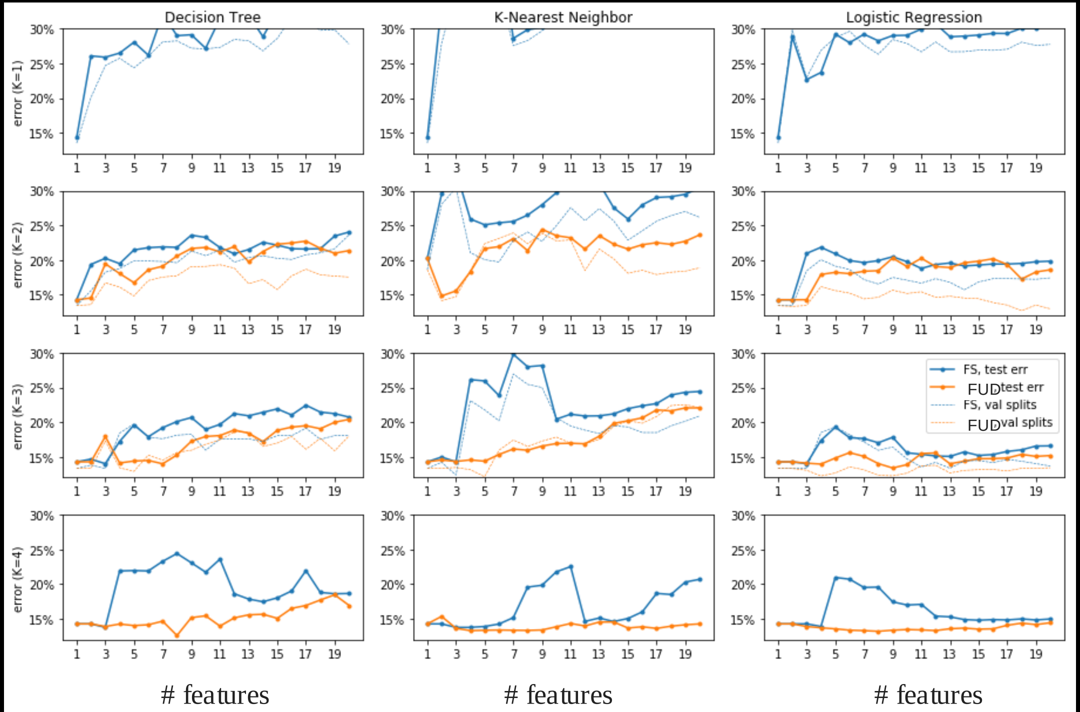
Results

$d = 1$

$d = 2$

$d = 3$

$d = 4$



Finding Useful Train Splits

- With data from only one city…
  - Divide data by North/South

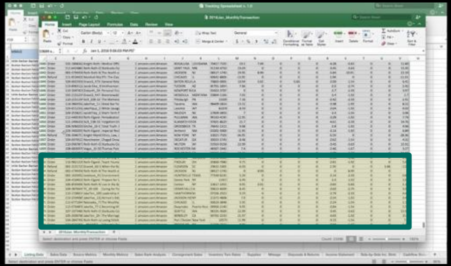
- With only MNIST handwritten digit data…
  - Split thick vs thin strokes

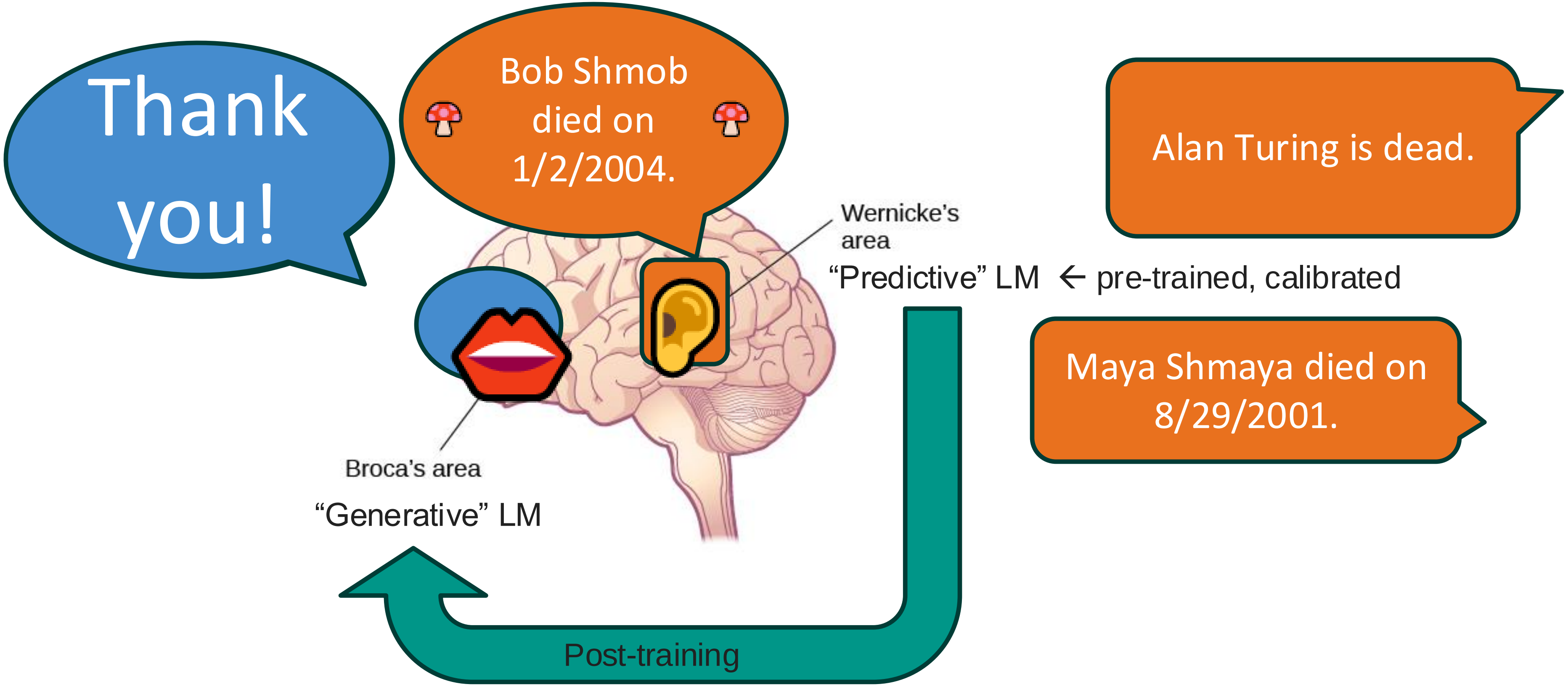
- With data from  iversi
  - Split by department

DEPARTMENT OF COMPUTER SCIENCE

Department of Physics

- With any dataset $X, Y \in (\mathcal{X} \times \mathcal{Y})^m$…
  - Split by example order in dataset





Wernicke's area is crucial for **language comprehension**.
Broca's area is essential for language **production**.

Conclusions

- ML does not naturally generalize to other domains
- Must “learn to generalize” to new domains
- Take-away: maintain split provenance
- New model for domain generalization

Alignment as a bandit problem

- Alignment procedures $A_0, A_1, \dots, A_k : V \to \Theta$
  - Vague description $v \in V$ (e.g., text, human annotators, NN simulators)
  - Outputs params $\theta \in \Theta$ (e.g., generative LM, system prompt)
  - Maximize true eventual utility $u: \Theta \to [0,1]$ only observed after the fact
  - Impossible if "we only get 1 shot"

- Multi-armed bandit setting. For $t = 1,2, \dots$:
  - Humans picks problem (u_t, v_t) but reveal only v_t
  - Learner chooses θ_t
  - Learner sees reward $u_t(\theta_t)$ and optional additional observations/feedback survey
  - Given $A_1, A_2, \dots, A_k : V \to \Theta$, can achieve $O\left(\sqrt{k/t}\right)$ avg regret relative to best A_{i^*}

Open problem

- Goal: provably safe ASI chess player (*not* ASI chess player trained on whole internet)

- Assumptions:
  - Benevolent humans
  - Train ASI chess player on only chess games and RL
  - Additional assumptions?

- To prove:
  - ASI chess player is “safe”
  - Won’t embed a computer virus that when LMs train on will do bad stuff?

- Safe to connect it to a small model, a “stupid aligned AI” that talks?