

FROM MODELS TO PRODUCTION SYSTEMS (SYSTEMS THINKING)

Christian Kaestner

- Hulten, Geoff. "Building Intelligent Systems: A Guide to Machine Learning Engineering." (2018), Chapters 5, 7, and 8.

NON-UPDATE ON WAITLIST



memegenerator.net

- LTI MS All program: Now on Canvas. Assume you are enrolled.
- All others: Please be patient. If enrolling late, can submit this week's quizzes and homework 1 late (1 week later).

ADMINISTRATIVA

- CATME survey for team assignment today
- Homework 1 due on Wednesday
- Team assignments announced most likely on Wednesday
- Reminder on grading rubrics
- Reminder on collaboration policy
- #assignments channel on Slack

4: Q4: Lessons ▾

0 OF 3 GRADED

TOTAL POINTS
0.0 / 10.0 pts

Rubric Settings

Collapse View ▾

1 +10.0

Meets requirements as the solution identifies and describes two engineering lessons learned in the project. The lessons generalize from the case study and make specific recommendations for other projects. The lessons relate to engineering concerns of the production system, not data-science concerns in the initial model development.

2 +0.0

Doesn't meet requirements. No lessons were identified and described from the projects.

LEARNING GOALS

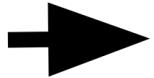
- Explain how machine learning fits into the larger picture of building and maintaining production systems
- Explain the consequences of the shift from deductive to inductive reasoning for abstraction and composition
- Explain the modularity implications of having machine-learning components without specifications
- Describe the typical components relating to AI in an AI-enabled system and typical design decisions to be made

ML MODELS AS PART OF A SYSTEM

EXAMPLE: IMAGE CAPTIONING PROBLEM

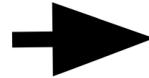


Image



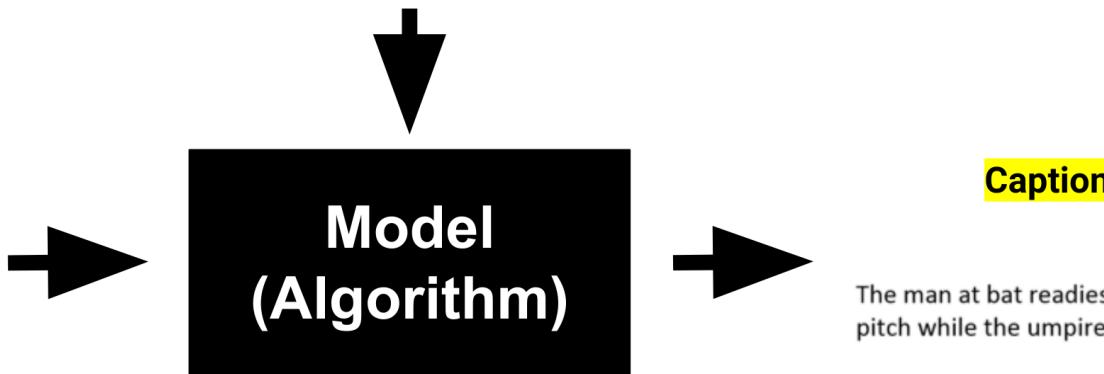
Algorithm

Caption



The man at bat readies to swing at the pitch while the umpire looks on.

EXAMPLE: IMAGE CAPTIONING PROBLEM

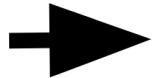


The man at bat readies to swing at the pitch while the umpire looks on.

WHY DO WE CARE ABOUT IMAGE CAPTIONING?



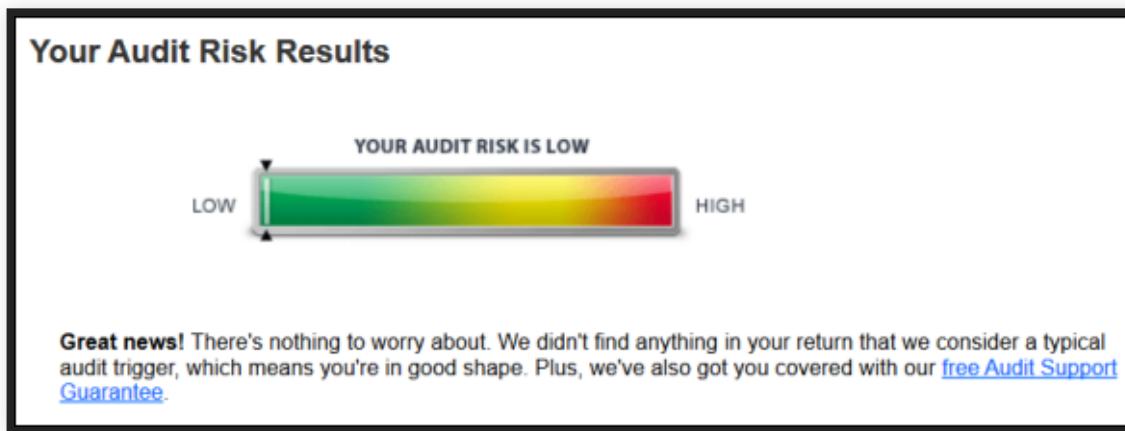
Algorithm



Caption

The man at bat readies to swing at the pitch while the umpire looks on.

MACHINE LEARNING AS (SMALL) COMPONENT IN A SYSTEM



Speaker notes

Traditional non-ML tax software, with an added ML component for audit risk estimation

MACHINE LEARNING AS (SMALL) COMPONENT IN A SYSTEM



MACHINE LEARNING AS (CORE) COMPONENT IN A SYSTEM

the-changelog-318
← Dashboard | Quality: High ⓘ

Last saved a few seconds ago ... Share

00:00 ⚡ Offset 00:00 01:31:27

Play Back 5s 1x Speed Volume

NOTES
Write your notes here

Speaker 5 ► 07:44

Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

Speaker 5 ► 08:38

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript? ☆☆☆☆☆

Speaker notes

Transcription service, where interface is all built around an ML component

MACHINE LEARNING AS (CORE) COMPONENT IN A SYSTEM

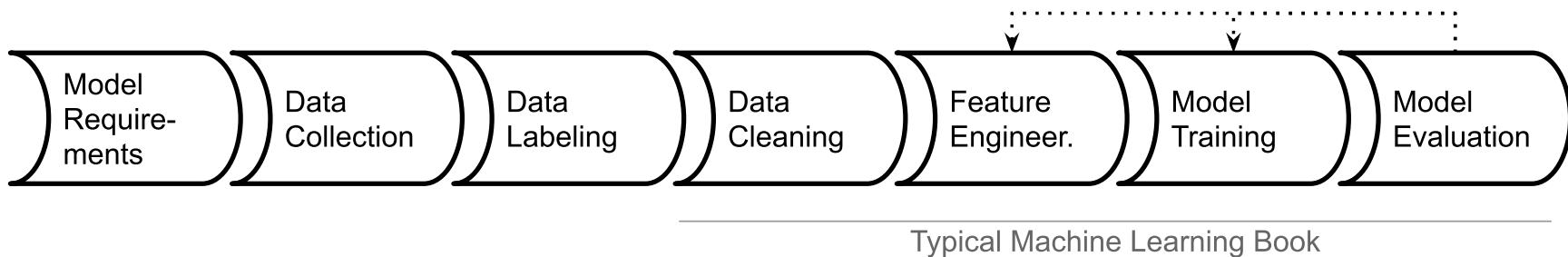


MANY MORE EXAMPLES:

- Product recommendations on Amazon
- Surge price calculation for Uber
- Inventory planning in Walmart
- Search for new oil fields by Shell
- Adaptive cruise control in a car
- Smart app suggestion in Android
- Fashion trends prediction with social media data
- Suggesting whom to talk to in a presidential campaign
- Tracking and predicting infections in a pandemic
- Adaptively reacting to network issues by a cell phone provider
- Matching players in a computer game by skill
- ...
- Some for end users, some for employees, some for expert users
- Big and small components of a larger system

MODEL-CENTRIC VS SYSTEM-WIDE FOCUS

TRADITIONAL MODEL FOCUS (DATA SCIENCE)



Focus: building models from given data, evaluating accuracy

AUTOMATING PIPELINES AND MLOPS (ML ENGINEERING)



Focus: experimenting, deploying, scaling training and serving, model monitoring and updating

MLOPS INFRASTRUCTURE



From: Sculley, David, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. "Hidden technical debt in machine learning systems."

Advances in neural information processing systems 28 (2015): 2503-2511.

Speaker notes

Figure from Google's 2015 technical debt paper, indicating that the amount of code for actual model training is comparably small compared to lots of infrastructure code needed to automate model training, serving, and monitoring. These days, much of this infrastructure is readily available through competing MLOps tools (e.g., serving infrastructure, feature stores, cloud resource management, monitoring).

ML-ENABLED SYSTEMS (ML IN PRODUCTION)



Focus: Interaction of ML and non-ML components, system requirements, user interactions, safety, collaboration, delivering products

MODEL VS SYSTEM GOALS

CASE STUDY: SELF-HELP LEGAL CHATBOT



**Welcome to the future of legal services.
Always available and ready to help.**

**Get legal assistance online. Chat with a lawyer for
free!**

Start Free Chat

Based on the excellent paper: Passi, S., & Sengers, P. (2020). [Making data science systems work](#). Big Data & Society, 7(2).

Speaker notes

Screenshots for illustration purposes, not the actual system studied

CASE STUDY: SELF-HELP LEGAL CHATBOT



PREVIOUS SYSTEM: GUIDED CHAT



Image source: <https://www.streamcreative.com/chatbot-scripts-examples-templates>

PROBLEMS WITH GUIDED CHATS

Non-AI guided chat was too limited

- Cannot enumerate problems
- Hard to match against open entries ("I want to file for bankruptcy" vs "I have no money")

Involving human operators very expensive

Old-fashioned



INITIAL GOAL: BETTER CHATBOT

- Help users with simple task
- Connect them with lawyers when needed
- Modernize appearance; "future of digital marketing"

BUY OR BUILD?

Botsify

Products ▾ Partner ▾ Resources ▾ Pricing Features Sign in [signup](#)

[Update](#) Instagram Chatbots are available on Botsify now

Premium Chatbot Platform For Everyone

Botsify is a managed chatbot platform that provide unified chat automation for your business. Get omnichannel live-chat service connected with multiple platforms to set autoresponses

1K Chatbots
20K Bot Responses
9K Hours Saved

9:45 TamimiMarkets Active Now

Hello ! This is Tamimi Market's chatbot, please choose your preferred language to use the services.

1 Arabic
2 English

Select your country by replying number:
1 Saudi Arabia
2 Bahrain

Select the required service by replying number:
1 Weekly Offers
2 Rewards Program
3 Store locations and timings
4 Complaints / Feedback

We're open 7 AM – 11 PM, Everyday. Click Here - <https://tinyurl.com/yjnu7vhd>

Type a message

Get Free Trial [Explore Platform Now](#)

Book Demo Now [Free 1-1 Product Tour](#)

+307 trials started in last 7 days

Speaker notes

One of many commercial frameworks for building AI chatbots

DATA SCIENCE CHALLENGES

- **Infrastructure:** Understand chat bot infrastructure and its capabilities
- **Knowing topics:** Identify what users talk about, train/test concepts with past chat logs
 - *"We fed VocabX a line deliberately trying to confuse it. We wrote, 'I am thinking about chapter 13 in Boston divorce filing.' VocabX figured out the two topics: (1) business and industrial/company/bankruptcy (2) society/social institution/divorce."*
- **Guiding conversations:** Supporting open-ended conversations requires detecting what's on topic and finding a good response; intent-topic modeling
 - *Is talk about parents and children on topic when discussing divorce?*
 - Data gathering/labeling very challenging -- too many corner cases

STEPPING BACK: SYSTEM GOALS

What are the goals of the system?



STATUS MEETING WITH (INHOUSE) CUSTOMER

The chatbot performed better than before but was far from ready for deployment. There were “too many edge cases” in which conversations did not go as planned.

Customer: "Maybe we need to think about it like an 80/20 rule. In some cases, it works well, but for some, it is harder. 80% everything is fine, and in the remaining 20%, we try to do our best."

Data science lead: The trouble is how to automatically recognize what is 80 and what is 20.

Customer: I agree. Let us focus on that. We just want value. Tech is secondary.

Data scientist: It is harder than it sounds. One of the models is a matching model trained on pairs of legal questions and answers. 60,000 of them. It seems large but is small for ML.

Customer: That's a lot. Can it answer a question about say visa renewal?

Data scientist: If there exists a question like that in training data, then yes. But with just 60,000, the model can easily overfit, and then for anything outside, it would just fail.

Customer: I see what you are saying. Edge cases are interesting from an academic perspective, but for a business the first and foremost thing is value. You are trying to solve an interesting problem. I get it. But I feel that you may have already solved it enough to gain business value.

Speaker notes

Adapted from Passi, S., & Sengers, P. (2020). [Making data science systems work](#). Big Data & Society, 7(2).

SYSTEM GOAL FOR CHATBOT

- Collect user data to sell to lawyers
- Signal technical competency to lawyers
- Acceptable to fail: Too complicated for self-help, connect with lawyer
- Solving edge cases not important

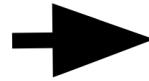
"Edge cases are important, but the end goal is user information, monetizing user data. We are building a legal self-help chatbot, but a major business use case is to tell people: 'here, talk to this lawyer.' We do want to connect them with a lawyer. Even for 20%, when our bot fails, we tell users that the problem cannot be done through self-help. Let us get you a lawyer, right? That is what we wanted in the first place."

Passi, S., & Sengers, P. (2020). [Making data science systems work](#). Big Data & Society, 7(2).

MODEL VS SYSTEM GOAL?



Algorithm



Caption

The man at bat readies to swing at the pitch while the umpire looks on.

MODEL VS SYSTEM GOAL?



MORE ACCURATE PREDICTIONS MAY NOT BE THAT IMPORTANT

- "Good enough" may be good enough
- Prediction critical for system success or just an gimmick?
- Better predictions may come at excessive costs
 - need way more data, much longer training times
 - privacy concerns
- Better user interface ("experience") may mitigate many problems
 - e.g. explain decisions to users
- Use only high-confidence predictions?

MACHINE LEARNING THAT MATTERS

- 2012 essay lamenting focus on algorithmic improvements and benchmarks in ML
 - focus on standard benchmark sets, not engaging with problem: Iris classification, digit recognition, ...
 - focus on abstract metrics, not measuring real-world impact: accuracy, ROC
 - distant from real-world concerns
 - lack of follow-through, no deployment, no impact
- Failure to *reproduce* and *productionize* paper contributions common
- Ignoring design choices in how to collect data, what problem to solve, how to design human-AI interface, measuring impact, ...
- Should focus on making impact -- requires building systems

Wagstaff, Kiri. "Machine learning that matters." In Proceedings of the 29 th International Conference on Machine Learning, (2012).

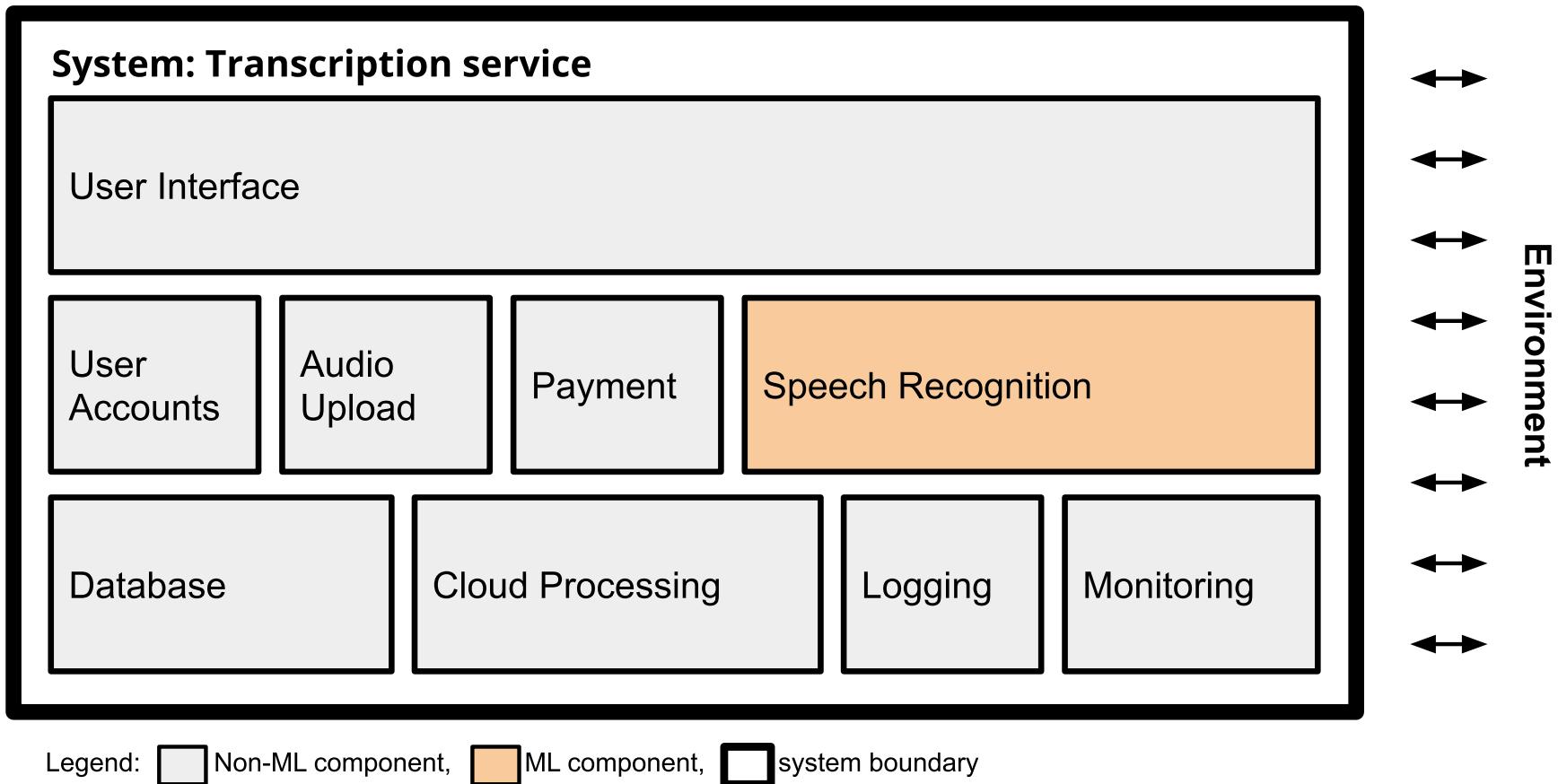
ON TERMINOLOGY

- There is no standard term for referring to building systems with AI components
- **ML-Enabled Systems**, *Production ML Systems*, **AI-Enabled Systems**, or *ML-Infused Systems*; *SE4AI*, *SE4ML*
- sometimes **AI Engineering** -- but usually used with a ML-pipeline focus
- **MLOps** ~ technical infrastructure automating ML pipelines
- sometimes **ML Systems Engineering** -- but often this refers to building distributed and scalable ML and data storage platforms
- "AIOps" ~ using AI to make automated decisions in operations; "DataOps" ~ use of agile methods and automation in business data analytics
- My preference: **Production Systems with Machine-Learning Components**

SYSTEMS THINKING



REPEAT: MACHINE LEARNING AS COMPONENT IN A SYSTEM



THE SYSTEM INTERACTS WITH USERS



Speaker notes

Audit risk meter from Turbo-Tax

THE SYSTEM INTERACTS WITH THE WORLD



THE SYSTEM INTERACTS WITH THE WORLD



- Model: Use historical data to predict crime rates by neighborhoods
- Used for predictive policing: Decide where to allocate police patrol

SYSTEM <-> WORLD = FEEDBACK LOOPS?





Artificial intelligence

Predictive policing algorithms are racist. They need to be dismantled.

Lack of transparency and biased training data mean these tools are not fit for purpose. If we can't fix them, we should ditch them.

by **Will Douglas Heaven**

July 17, 2020

ML PREDICTIONS HAVE CONSEQUENCES

- Assistance, productivity, creativity
- Manipulation, polarization, discrimination
- Feedback loops
- Need for **responsible engineering**

SAFETY IS A SYSTEM PROPERTY

- Code/models are not unsafe, cannot harm people
- Systems can interact with the environment in ways that are unsafe



SAFETY ASSURANCE IN THE MODEL/OUTSIDE THE MODEL

Goal: Ensure smart toaster does not burn the kitchen



SAFETY ASSURANCE IN THE MODEL/OUTSIDE THE MODEL

- In the model
 - Ensure maximum toasting time
 - Use heat sensor and past outputs for prediction
 - Hard to make guarantees
- Outside the model (e.g., "guardrails")
 - Simple code check for max toasting time
 - Non-ML rule to shut down if too hot
 - Hardware solution: thermal fuse



(Image CC BY-SA 4.0, C J Cowie)

MODEL VS SYSTEM PROPERTIES

- Similar to safety, many other qualities should be discussed at model **and** system level
 - Security
 - Privacy
 - Transparency, accountability
 - Maintainability
 - Scalability, energy consumption
 - Impact on system goals
 - ...

THINKING ABOUT SYSTEMS

- Holistic approach, looking at the larger picture, involving all stakeholders
- Looking at relationships and interactions among components and environments
 - Everything is interconnected
 - Combining parts creates something new with emergent behavior
 - Understand dynamics, be aware of feedback loops, actions have effects
- Understand how humans interact with the system

A system is a set of inter-related components that work together in a particular environment to perform whatever functions are required to achieve the system's objective --
Donella Meadows

SYSTEM-LEVEL CHALLENGES FOR AI-ENABLED SYSTEMS

- Getting and updating data, concept drift, changing requirements
- Handling massive amounts of data
- Interactions with the real world, feedback loops
- Lack of modularity of AI components, lack of specifications, nonlocal effects
- Deployment and maintenance
- Versioning, debugging and incremental improvement
- Keeping training and operating cost manageable
- Interdisciplinary teams
- Setting system goals, balancing stakeholders and requirements
- ...

DESIGNING INTELLIGENT EXPERIENCES

(Human-AI Interaction)

AI PREDICTIONS SHOULD INFLUENCE THE WORLD

- Smart toaster
- Automated slide design
- Product or music recommendations
- Feed curation in social media or news
- Recidivism prediction
- Health monitoring
- Transcription services
- Image search engine
- Self-driving cars
- Smart home
- Interact with the world through actuators (smart devices) or by influencing people

DESIGNING INTELLIGENT EXPERIENCES

- How to use the output of a model's prediction (for a objective)?
- Design considerations:
 - How to present prediction to a user? Suggestions or automatically take actions?
 - How to effectively influence the user's behavior toward the system's goal?
 - How to minimize the consequences of flawed predictions?
 - How to collect data to continue to learn from users and mistakes?
- Balancing at least three **system-level** outcomes:
 - Achieving objectives
 - Protection from mistakes
 - Collecting data for training

PRESENTING INTELLIGENCE

- Automate: Take action on user's behalf
- Prompt: Ask the user if an action should be taken
- Organize: Display a set of items in an order
- Annotate: Add information to a display
- Hybrids of these

FACTORS TO CONSIDER

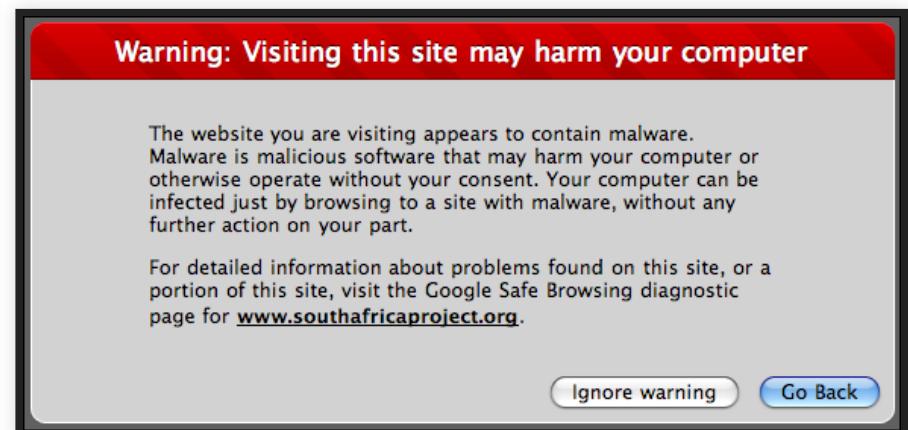
- **Forcefulness:** How strongly to encourage taking an action (or even automate it)?
- **Frequency:** How often to interact with the user?
- **Value:** How much does a user (think to) benefit from the prediction?
- **Cost:** What is the damage of a wrong prediction?

BREAKOUT DISCUSSION: EXPERIENCE DESIGN

Fall detection for elderly people:



Safe browsing: Blocking malicious web pages



- How do we present the intelligence to the user?
- Consider system goals, forcefulness, frequency, value of correct and cost of wrong predictions

Speaker notes

Devices for older adults to detect falls and alert caretaker or emergency responders automatically or after interaction.
Uses various inputs to detect falls. Read more: [How fall detection is moving beyond the pendant](#), MobiHealthNews, 2019

COLLECTING FEEDBACK

Report Incorrect Phishing Warning

If you received a phishing warning but believe that this is actually a legitimate page, please complete the form below to report the error to Google. Information about your report will be maintained in accordance with Google's [privacy policy](#).

URL:



I'm not a robot



reCAPTCHA
Privacy - Terms

Comments:
(Optional)

Submit Report

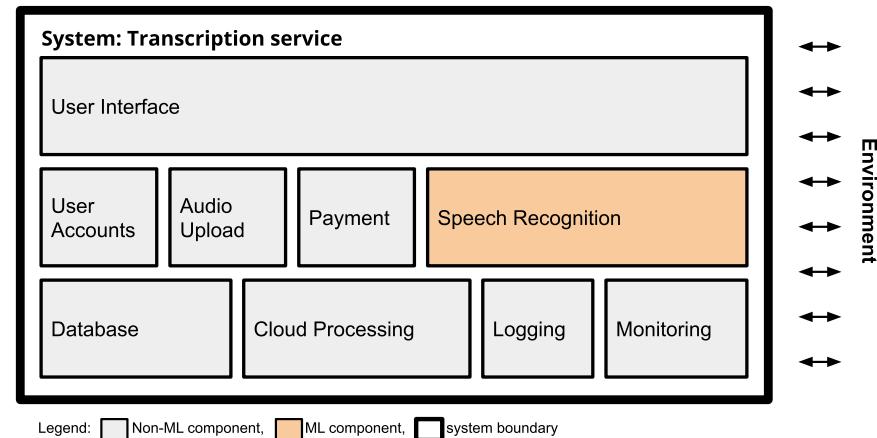
Google

OPERATING PRODUCTION ML SYSTEMS

(deployment, updates)

THINGS CHANGE...

- Newer better models released
 - Better model architectures
 - More training data
- Goals and scope change
 - More domains supported
 - Better recognition of dialects
- Model training due to drift
 - New terms (new products, new people) emerge in domain
 - Increased adoption in region with dialect
- Online experimentation



THINGS CHANGE...



Reasons for change in audit risk prediction?

Your Audit Risk Results



Great news! There's nothing to worry about. We didn't find anything in your return that we consider a typical audit trigger, which means you're in good shape. Plus, we've also got you covered with our [free Audit Support Guarantee](#).

MONITORING IN PRODUCTION

Design for telemetry

Report Incorrect Phishing Warning

If you received a phishing warning but believe that this is actually a legitimate page, please complete the form below to report the error to Google. Information about your report will be maintained in accordance with Google's [privacy policy](#).

URL:

I'm not a robot  reCAPTCHA
Privacy - Terms

Comments: (Optional)





MONITORING IN PRODUCTION



What and how to monitor in audit risk prediction?

Your Audit Risk Results

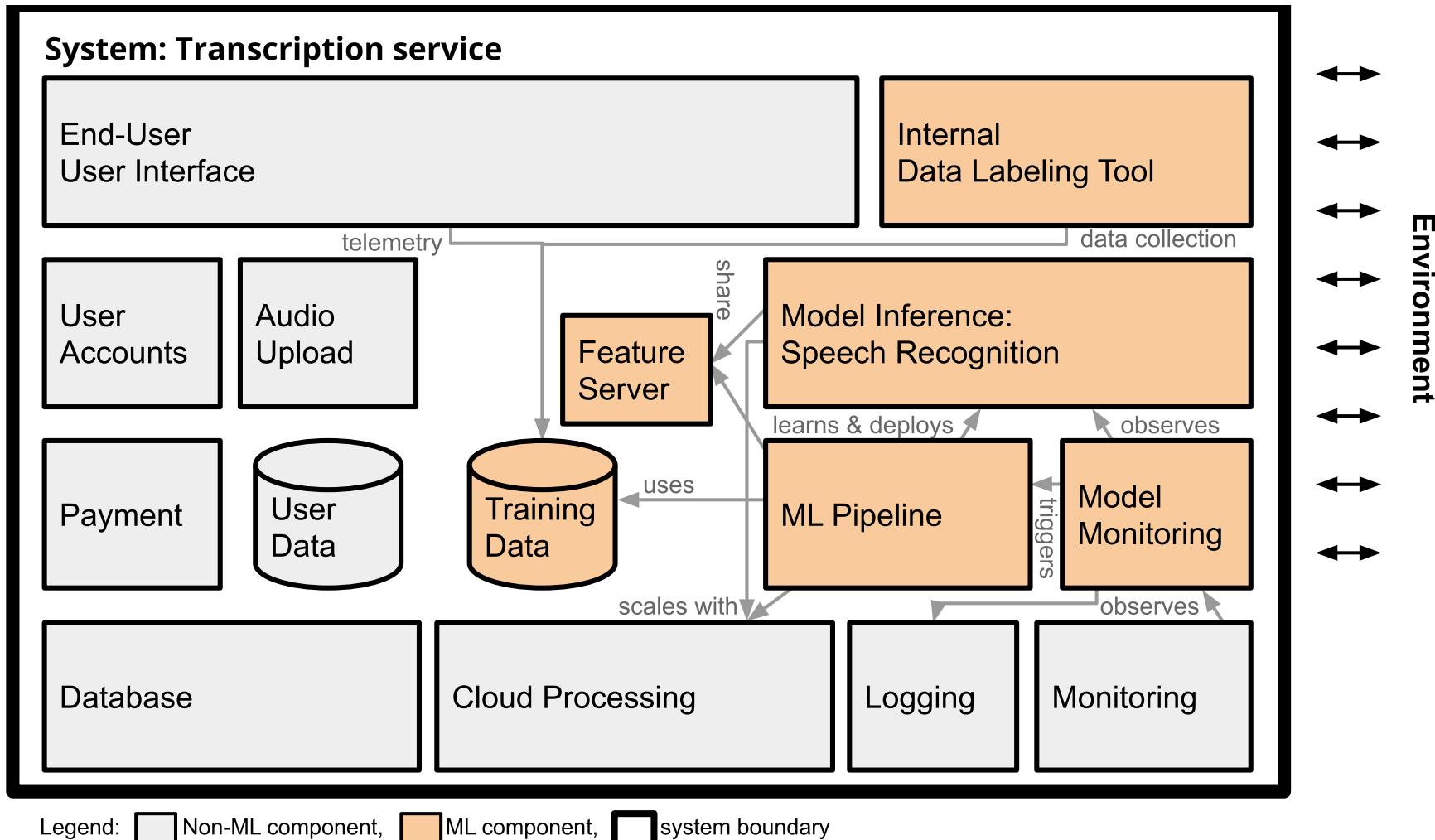


Great news! There's nothing to worry about. We didn't find anything in your return that we consider a typical audit trigger, which means you're in good shape. Plus, we've also got you covered with our [free Audit Support Guarantee](#).

PIPELINE THINKING



DESIGN WITH PIPELINE AND MONITORING IN MIND



SHIFTING FROM MODELS TO PIPELINES IS CHALLENGING

Across interviews with enterprise ML teams:

- Data scientists often focus on modeling in local environment, model-centric workflow
- Rarely robust infrastructure, often monolithic and tangled
- Challenges in deploying systems and integration with monitoring, streams etc
- Shifting to pipeline-centric workflow challenging
- Requires writing robust programs, slower, less exploratory
- Standardized, modular infrastructure
- Big conceptual leap, major hurdle to adoption

O'Leary, Katie, and Makoto Uchida. "[Common problems with Creating Machine Learning Pipelines from Existing Code](#)." Proc. Third Conference on Machine Learning and Systems (MLSys) (2020).

WHAT MAKES SOFTWARE WITH ML CHALLENGING?

ML MODELS MAKE MISTAKES



NeuralTalk2: A flock of birds flying in the air

Microsoft Azure: A group of giraffe standing next to a tree

Image: Fred Dunn, <https://www.flickr.com/photos/gratapictures> - CC-BY-NC

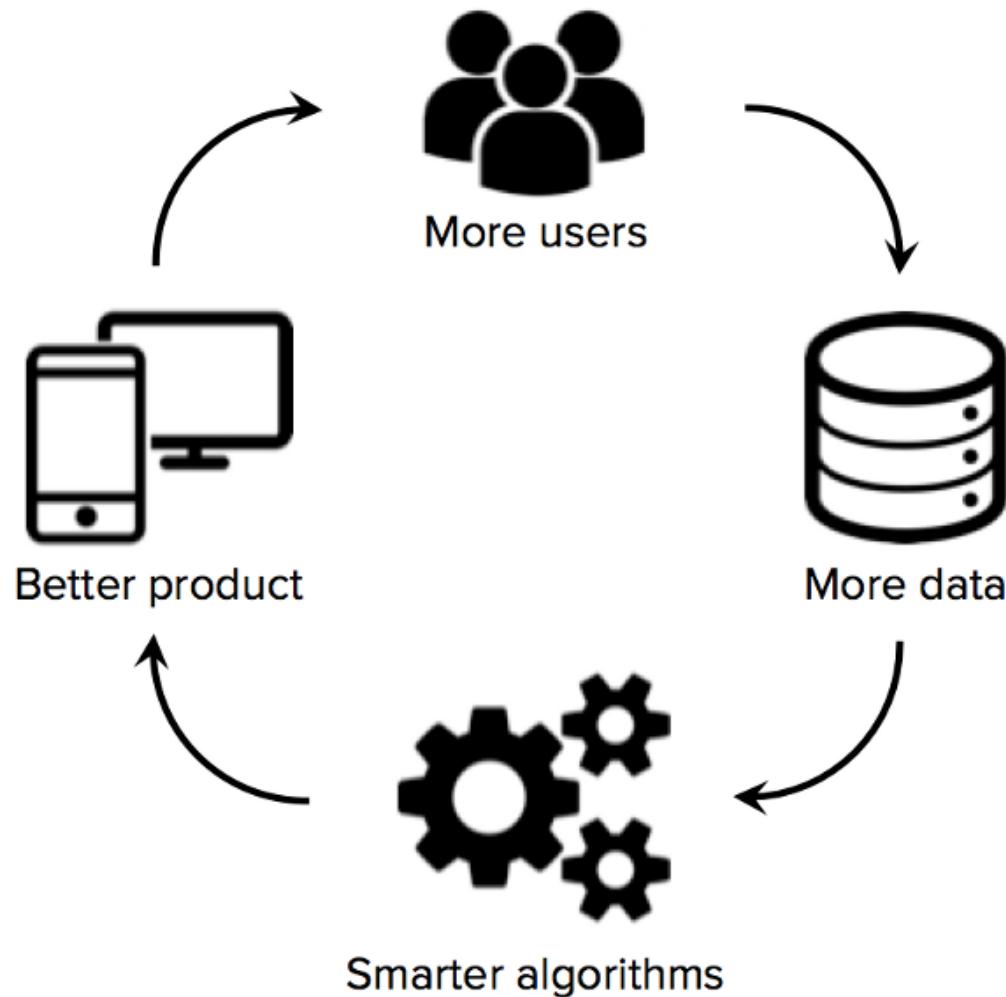
Speaker notes

Source: <https://www.aiweirdness.com/do-neural-nets-dream-of-electric-18-03-02/>

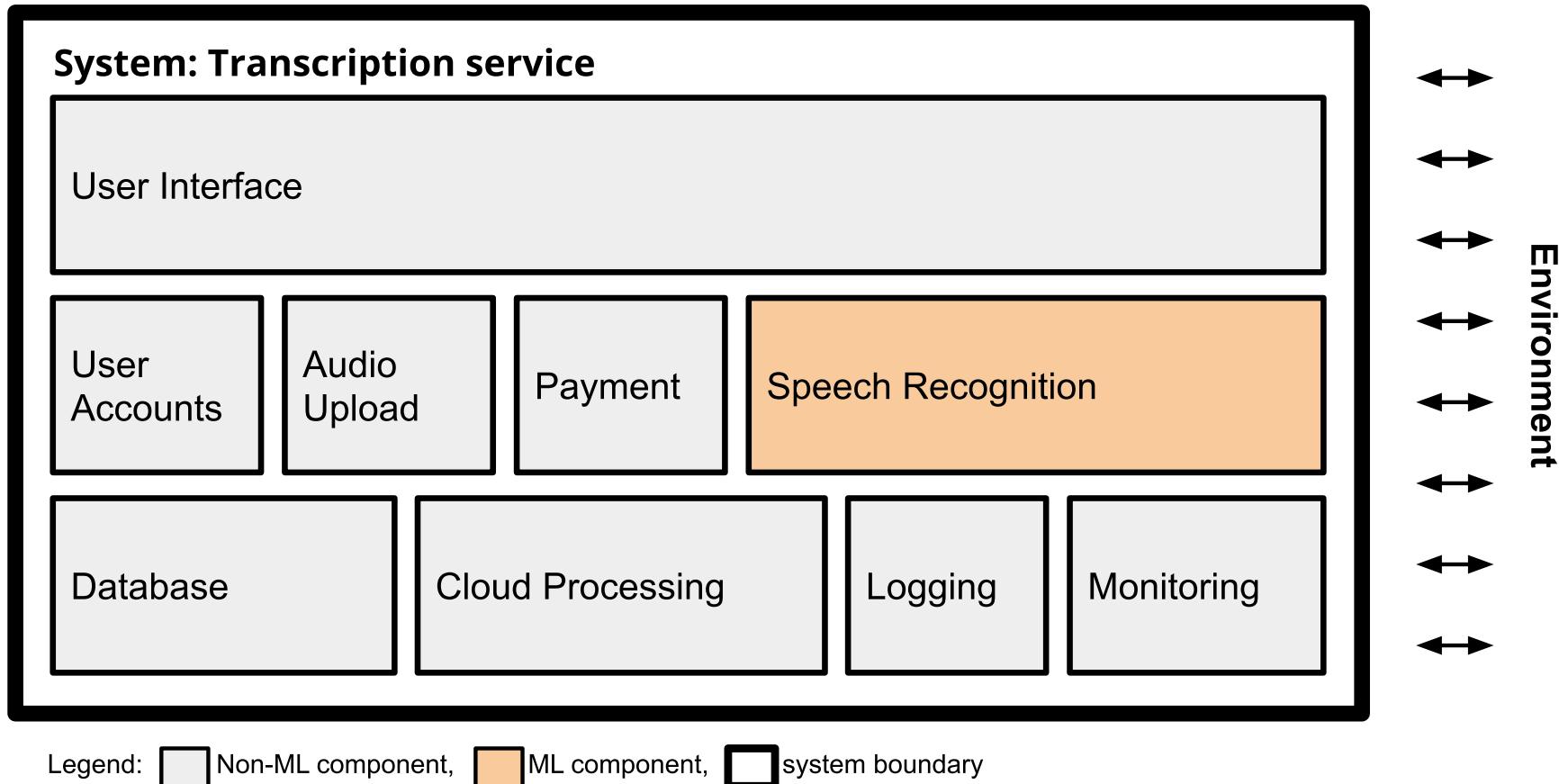
LACK OF SPECIFICATIONS

```
/**  
 * Return the text spoken within the audio file  
 * ????  
 */  
String transcribe(File audioFile);
```

DATA FOCUSED AND SCALABLE



INTERACTION WITH THE ENVIRONMENT



IT'S NOT ALL NEW

- Safe software with unreliable components
- Cyberphysical systems
- Non-ML big data systems, cloud systems
- "Good enough" and "fit for purpose" not "correct"

- We routinely build such systems
- ML intensifies our challenges

COMPLEXITY



ML CHALLENGES SYSTEM DECOMPOSITION

(deductive vs inductive reasoning)

COMPLEXITY IN ENGINEERED SYSTEMS



- Automobile: ~30,000 parts; Airplane: ~3,000,000 parts
- MS Office: ~ 40,000,000 LOCs; Debian: ~ 400,000,000 LOCs
- How do we build such complex systems?

MANAGING COMPLEXITY IN SOFTWARE

- **Abstraction:** Hide details & focus on high-level behaviors
- **Reuse:** Package into reusable libraries & APIs with well-defined *contracts*
- **Composition:** Build large components out of smaller ones



MANAGING COMPLEXITY IN SOFTWARE

- **Abstraction:** Hide details & focus on high-level behaviors
- **Reuse:** Package into reusable libraries & APIs with well-defined *contracts*
- **Composition:** Build large components out of smaller ones

```
/**  
 * compute deductions based on provided adjusted  
 * gross income and expenses in customer data.  
 *  
 * see tax code 26 U.S. Code A.1.B, PART VI  
 *  
 * Adjusted gross income must be positive;  
 * returned deductions are not negative.  
 */  
float computeDeductions(float agi, Expenses expenses) {  
    ...  
}
```

DIVIDE AND CONQUER

- Human cognitive ability is limited
- Decomposition of software necessary to handle complexity
- Allows division of labor
- Deductive reasoning, using logic



DEBUGGING AND ASSIGNING BLAME

- Each component has own specification
- For each input, specification indicates whether output correct

```
/**  
 * compute deductions based on provided adjusted  
 * gross income and expenses in customer data.  
 *  
 * see tax code 26 U.S. Code A.1.B, PART VI  
 */  
float computeDeductions(float agi, Expenses expenses);
```



STRICT CORRECTNESS ASSUMPTION

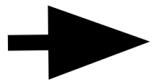
- Specification determines which outputs are correct/wrong
- Not "pretty good", "95% accurate", or "correct for 98% of all users"
- A single wrong result indicates a bug in the system



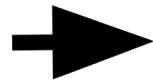
Speaker notes

A single wrong tax prediction would be a bug. No tolerance of occasional wrong predictions, approximations, nondeterminism.

IMAGE CAPTIONING ALGORITHM



Algorithm



Caption

The man at bat readies to swing at the pitch while the umpire looks on.

```
/**  
 * ????  
 */  
String getCaption(Image img);
```

Speaker notes

We do not know how to program this or specify this. No way of saying whether caption is "correct" for input, but defer to human judgement.

LEARNING IMAGE CAPTIONING ALGORITHM



Learning rules by fitting to examples, no specification, inductive reasoning

Speaker notes

"Rules"/algorithm learned from data. Still no specification. Best fit to given training data.

CORRECTNESS OF MODEL?



1. A **blender**
sitting on top
of a cake.

*All models are wrong, but
some are useful. -- George
Box*

Image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: troubleshooting integrative machine learning systems." In Proc. AAAI. 2017.

Speaker notes

Human judgment needed. Furthermore, a single bad example is not a problem.

WEAK CORRECTNESS ASSUMPTIONS

- Often no reliable ground truth (e.g. human judgment)
- Accepting that mistakes will happen, hopefully not too frequently; "95% accuracy" may be pretty good
- More confident for data similar to training data



1. A **blender**
sitting on top
of a cake.

SPECIFICATIONS IN MACHINE LEARNING?

- Usually clear specifications do not exist -- we use machine learning exactly because we do not know the specifications
- Can define correctness for some data, but not general rules; sometimes can only determine correctness after the fact
- Learning for tasks for which we cannot write specifications
 - Too complex
 - Rules unknown
- ML will learn rules/specifications (inductive reasoning), often not in a human-readable form, but are those the right ones?
- Usually goals used instead --> maximize a specific objective



(Daniel Miessler, CC SA 2.0)

DEDUCTIVE REASONING

- Combining logical statements following agreed upon rules to form new statements
- Proving theorems from axioms
- From general to the particular
- *mathy reasoning, eg. proof that π is irrational*
- Formal methods, classic rule-based AI systems, expert systems

INDUCTIVE REASONING

- Constructing axioms from observations
- Strong evidence suggests a rule
- From particular to the general
- *sciency reasoning, eg. finding laws of nature*
- Most modern machine learning systems, statistical learning

CONSEQUENCES FROM LACK OF SPECIFICATIONS



Speaker notes

Breaks many traditional assumptions and foundations for compositional reasoning and divide and conquer

Poorly understood interactions between models: Ideally, develop models separately & compose together. In general, must train & tune together.

DECOMPOSING THE IMAGE CAPTIONING PROBLEM?



Speaker notes

Using insights of how humans reason: Captions contain important objects in the image and their relations. Captions follow typical language/grammatical structure

STATE OF THE ART DECOMPOSITION (IN 2015)



Example and image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "[On human intellect and machine failures: troubleshooting integrative machine learning systems](#)." In Proc. AAAI. 2017.

BLAME ASSIGNMENT?



Visual Detector

- | | |
|--------------------|------|
| 1. teddy | 0.92 |
| 2. on | 0.92 |
| 3. cake | 0.90 |
| 4. bear | 0.87 |
| 5. stuffed | 0.85 |
| ... | |
| 15. blender | 0.57 |

Language Model

1. A teddy bear.
2. A stuffed bear.
- ...
108. A **blender** sitting on top of a cake.

Caption Reranker

1. A **blender** sitting on top of a cake.
2. A teddy bear **in front of** a birthday cake.
3. A cake sitting on top of a **blender**.

Example and image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "[On human intellect and machine failures: troubleshooting integrative machine learning systems.](#)" In Proc. AAAI. 2017.

NONMONOTONIC ERRORS



Visual Detector

teddy	0.92
computer	0.91
bear	0.90
wearing	0.87
keyboard	0.84
glasses	0.63

1. A teddy bear
sitting **on top**
of a computer.

Fixed Visual Detector

teddy	1.0
bear	1.0
wearing	1.0
keyboard	1.0
glasses	1.0

1. **a person wearing
glasses and holding
a teddy bear sitting
on top of a keyboard.**

Example and image from: Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "[On human intellect and machine failures: troubleshooting integrative machine learning systems.](#)" In Proc. AAAI. 2017.

TAKEAWAY: SHIFT IN DESIGN THINKING?

Breaking traditional decomposition and reasoning strategies...

From deductive reasoning to inductive reasoning...

From clear specifications to goals...

From guarantees to best effort...

What does this mean for software engineering?

For decomposing software systems?

For correctness of AI-enabled systems?

For safety?

For design, implementation, testing, deployment, operations?

These problems are not new, but are exacerbated by the increasing use of ML!

SUMMARY

- Production AI-enabled systems require a *whole system perspective*, beyond just the model or the pipeline
- Distinguish system goals from model goals
- Quality at a *system level*: safety beyond the model, beyond accuracy
- Plan for operations (telemetry, updates)
- Large design space for user interface (intelligent experience): forcefulness, frequency, telemetry

RECOMMENDED READINGS

- Passi, S., & Sengers, P. (2020). [Making data science systems work](#). *Big Data & Society*, 7(2).
- Wagstaff, Kiri. "[Machine learning that matters.](#)" In Proceedings of the 29th International Conference on Machine Learning, (2012).
- Sculley, David, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. "[Hidden technical debt in machine learning systems.](#)" In *Advances in neural information processing systems*, pp. 2503-2511. 2015.
- O'Leary, Katie, and Makoto Uchida. "[Common problems with Creating Machine Learning Pipelines from Existing Code.](#)" Proc. Third Conference on Machine Learning and Systems (MLSys) (2020).
- Nushi, Besmira, Ece Kamar, Eric Horvitz, and Donald Kossmann. "[On human intellect and machine failures: troubleshooting integrative machine learning systems.](#)" In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1017-1025. 2017.
- Nahar, Nadia, Shurui Zhou, Grace Lewis, and Christian Kästner. "[Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process.](#)" In *Proceedings of the 44th International Conference on Software Engineering (ICSE)*, May 2022.
- Yang, Qian. "[The role of design in creating machine-learning-enhanced user experience.](#)" In *2017 AAAI Spring Symposium Series*. 2017.
- Sambasivan, Nithya, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M. Aroyo. "[“Everyone wants to do the model work, not the data work”: Data Cascades in High-Stakes AI](#)". In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1-15. 2021.