Capabilities for Better ML Engineering

Chenyang Yang, Rachel Brower-Sinning, Grace A. Lewis, Christian Kästner, Tongshuang Wu

**Motivation**

Coarse-grained metrics like test accuracy often can not reveal potential (safety) issues in production. Existing work focuses on various model qualities and evaluation strategies but are largely scattered and unconnected.

**Capabilities**

A unifying framework for scattered work on ML specifications A useful abstraction to reason about in ML engineering, especially in safety-critical systems

---

**Example: Pedestrian Detection**

Capabilities unite existing efforts on model qualities and data augmentation.

Detect pedestrians...

- **Robustness**: in extreme weather using wheelchairs of different body sizes in rural area wearing costumes on a scooter of different skin colors
- **Generalizability**: Perturbation
- **Fairness**: Data slicing
- **Capabilities**: Counterfactual

**Model qualities**

- **Capabilities**
- **Augmentation strategies**

---

**Research Opportunities**

1. **Identification**
   - How to identify capabilities?
   - How to support more effective discovery and reuse of domain knowledge? When and how can we automate discovery?
   - How to support more efficient human-AI interaction in error analysis?
   - How to design a better process to help both experts and non-experts identify capabilities?

2. **Assessment**
   - How to assess capabilities' importance?
   - What is a good granularity for a capability?
   - How to evaluate or rank capabilities by context?

3. **Communication**
   - How to communicate capabilities?
   - How to develop a shared language or interface to facilitate capability communication?
   - How can capabilities support conflict resolution between different stakeholders?

4. **Instantiation**
   - How to instantiate capabilities to concrete examples?
   - How to select instantiation strategies in different scenarios? How to measure and trade off costs and benefits?
   - How do different instantiation strategies complement each other?

---

**Broad Usage Scenarios**

- **Model Debugging**
  - Use capabilities to generalize from individual mistakes to systematic problems.
  - **Stakeholders**: Data scientists
  - **Stages**: Model design, development

- **Model Maintenance**
  - Use capabilities to characterize data shift and build regression tests.
  - **Stakeholders**: Data scientists, end users...
  - **Stages**: Model deployment

- **External Quality Assurance**
  - Use capabilities to provide a holistic view of how models perform in different scenarios.
  - **Stakeholders**: External evaluators, regulators...
  - **Stages**: Model evaluation

- **Collaboration**
  - Use capabilities as a communication interface between different stakeholders.
  - **Stakeholders**: Data scientists, software engineers...
  - **Stages**: Model requirements, documentation

- **Data Documentation**
  - Use capabilities to provide abstractions for concrete data points.
  - **Stakeholders**: Data scientists, data collectors, data annotators....
  - **Stages**: Data curation, documentation

---

**Experiment Findings**

**Experiment setup**: We collected 8 capability test suites for sentiment analysis and measured models’ performance on capability test suites and out-of-distribution data.

**Finding 1**: Model performance on capability tests is a strong signal for model's generalizability.

**Finding 2**: Capability tests especially helps predict how well models generalize to further distributions.

**Finding 3**: Different capabilities add different amount of information.

**Finding 4**: Different capabilities add different kinds of information (from complementary, similar, to conflicting).

---

Check out our paper!