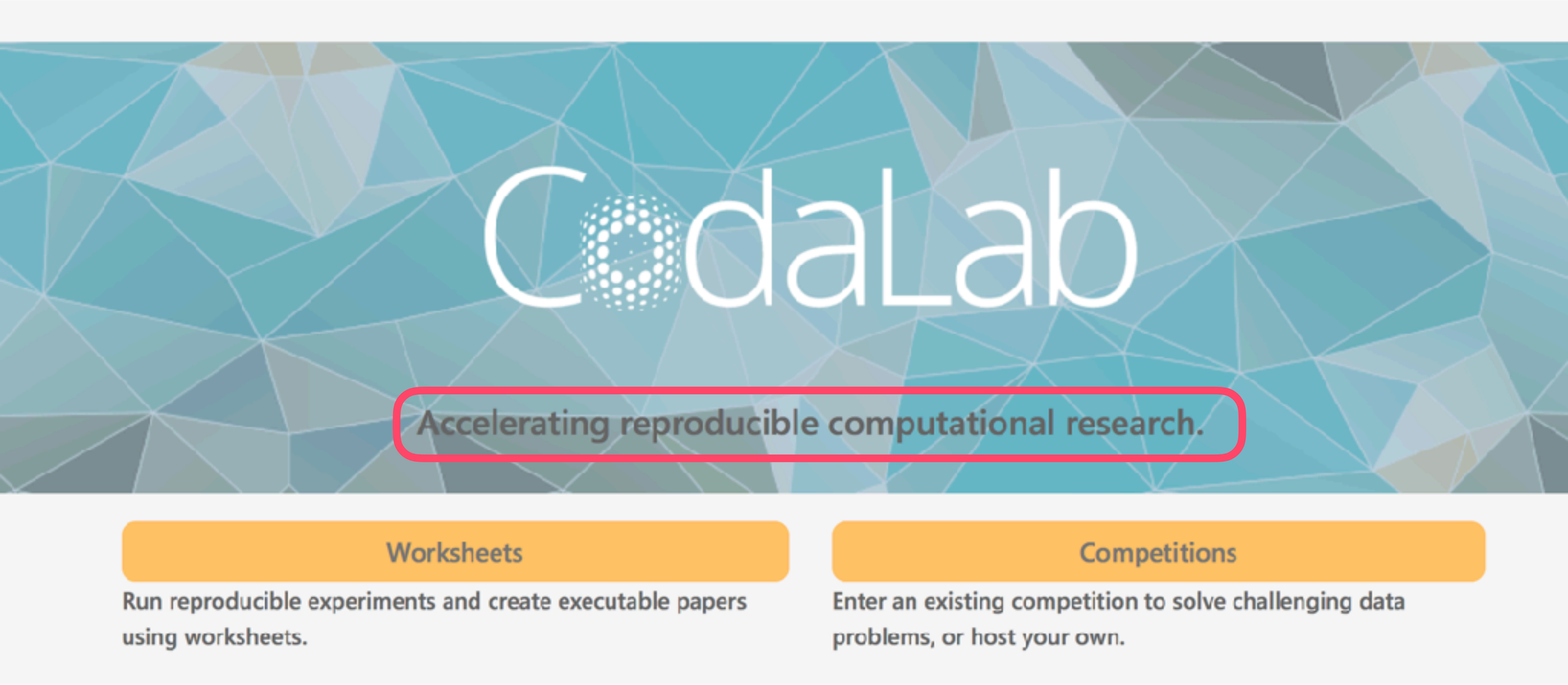
Computing Environment for Course Project

Chen Zhao

CodaLab docker

CodaLab

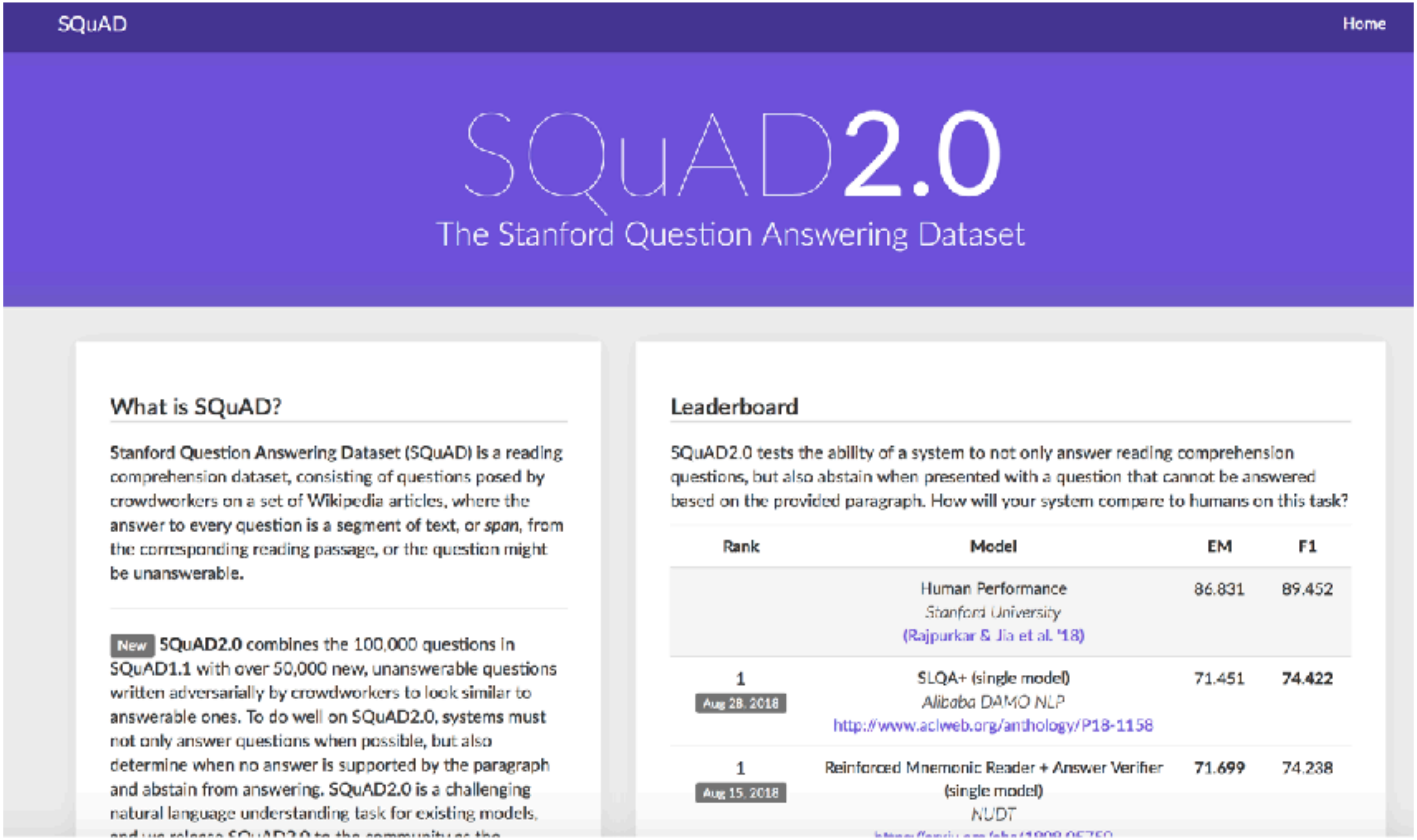


Accelerating reproducible computational research.

Run reproducible experiments and create executable papers using worksheets.

Enter an existing competition to solve challenging data problems, or host your own.

CodaLab



SQuAD2.0

The Stanford Question Answering Dataset

What is SQuAD?

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.

New **SQuAD2.0** combines the 100,000 questions in SQuAD1.1 with over 50,000 new, unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering. SQuAD2.0 is a challenging natural language understanding task for existing models, and we release SQuAD2.0 to the community as the

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph. How will your system compare to humans on this task?

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Aug 28, 2018	SLQA+ (single model) *Alibaba DAMO NLP* http://www.aclweb.org/anthology/P18-1158	71.451	**74.422**
1 Aug 15, 2018	Reinforced Mnemonic Reader + Answer Verifier (single model) *NUDT* https://arxiv.org/abs/1808.05759	**71.699**	74.238

CodaLab

- CodaLab set up:
    - Create your group account (both CodaLab and class)
    - Install command-line interface (Recommended)
    - Submit!

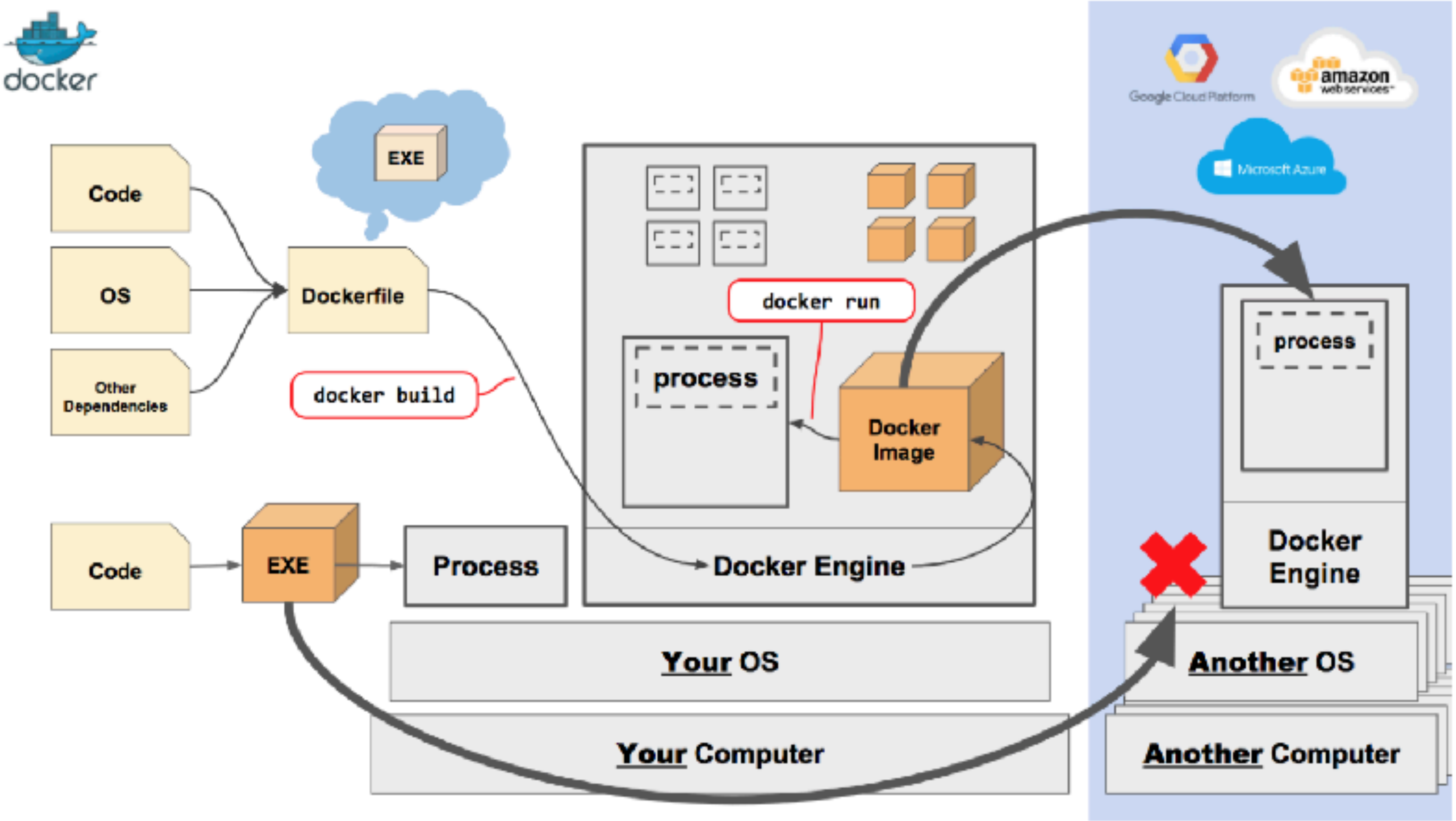
CodaLab

- Run official evaluation locally
  - Use the evaluation script we provided
  - Run the trained model on dev set
- Submit your model to the leaderboards
  - Docker submission
  - We test the model on private test set, then update the leaderboard

Docker

- A modern platform for AI applications
  - Automate the deployment of software applications inside **containers**
  - Allow users to package an application with all of its dependencies into the container
  - Applications run on **different OS** via a layer of indirection

Docker


docker
Code
OS
Other Dependencies
EXE
Dockerfile
docker build
Code
EXE
Process
docker run
process
Docker Image
Docker Engine
Your OS
Your Computer
Google Cloud Platform
amazon web services
Microsoft Azure
process
Docker Engine
Another OS
Another Computer

Docker

- We will provide a basic docker image
    - All dependencies for the baseline model
    - Include other dependencies in your code
    - We also provide some practice for using Docker
- You can customize your docker image from scratch(not recommended)

Course Project

- Overview
- Input/Output
- Dataset
- Baseline
- Evaluation

Course Project - Overview

- Incremental
  - We have several sub-tasks as homework (Deep Learning, Sequence Tagging..)
  - Final project will benefit from the previous sub-task code/results
  - Sub-task will use docker for submission
    - We will provide some hands on practice for getting used to Docker in sub-tasks
  - Final course project will use both docker and CodaLab

Course Project - I/O

- Example I/O
  - Input: {112; At its premiere, the librettist of this opera portrayed a character who asks for a glass of wine with his dying wish; False; WAIT}
  - Output: {The Magic Flute; BUZZ }
- Input:
  - Index (character position)
  - Question text up to this position
  - Is_new_sent
  - Opponent Action (probably BLANK)
- Output

Course Project - I/O

- Input:
- Output:
    - Top-1 guess
    - Action (buzz or wait)

Course Project - Dataset

- Data locates at https://github.com/Pinafore/qb#downloading-data
- Training/dev/test
    - Training (questions, correct answer w/wiki mapping)
    - Two evaluation datasets
        - Held-out data (from different tournaments)
        - Adversarial questions (more challenging)
            - Avoid clues that are easy for computers to answer

Course Project - Baseline

- Baseline system (2017 NIPS competition)
  - Guesser
    - IR system (TF-IDF) keyword-matching
    - 31862 QB questions, 6991 entities
    - Given a query(question), compare to previously asked questions, sort the answer entities based on similarity
    - TOP1 Accuracy 0.55
  - Buzzer

Course Project - Baseline

- Baseline system (2017 NIPS competition)
  - Guesser
  - Buzzer
    - Based on the guesser score
    - Threshold
      - Top-score exceeds the threshold(0.3)

Course Project - Evaluation

- Evaluation
  - Accuracy (Based on top-1 guess)
  - Head-to-head competition
    - Evaluate on same output, but like real competition
    - Check the answer of model that buzz first, if not correct, check the other model's answer at the end
    - 10 points for correct answer, -5 for incorrect if not at the end of question