Simulating Bandit Learning from User Feedback for Extractive QA

Ge Gao, Eunsol Choi, and Yoav Artzi

ACL 2022

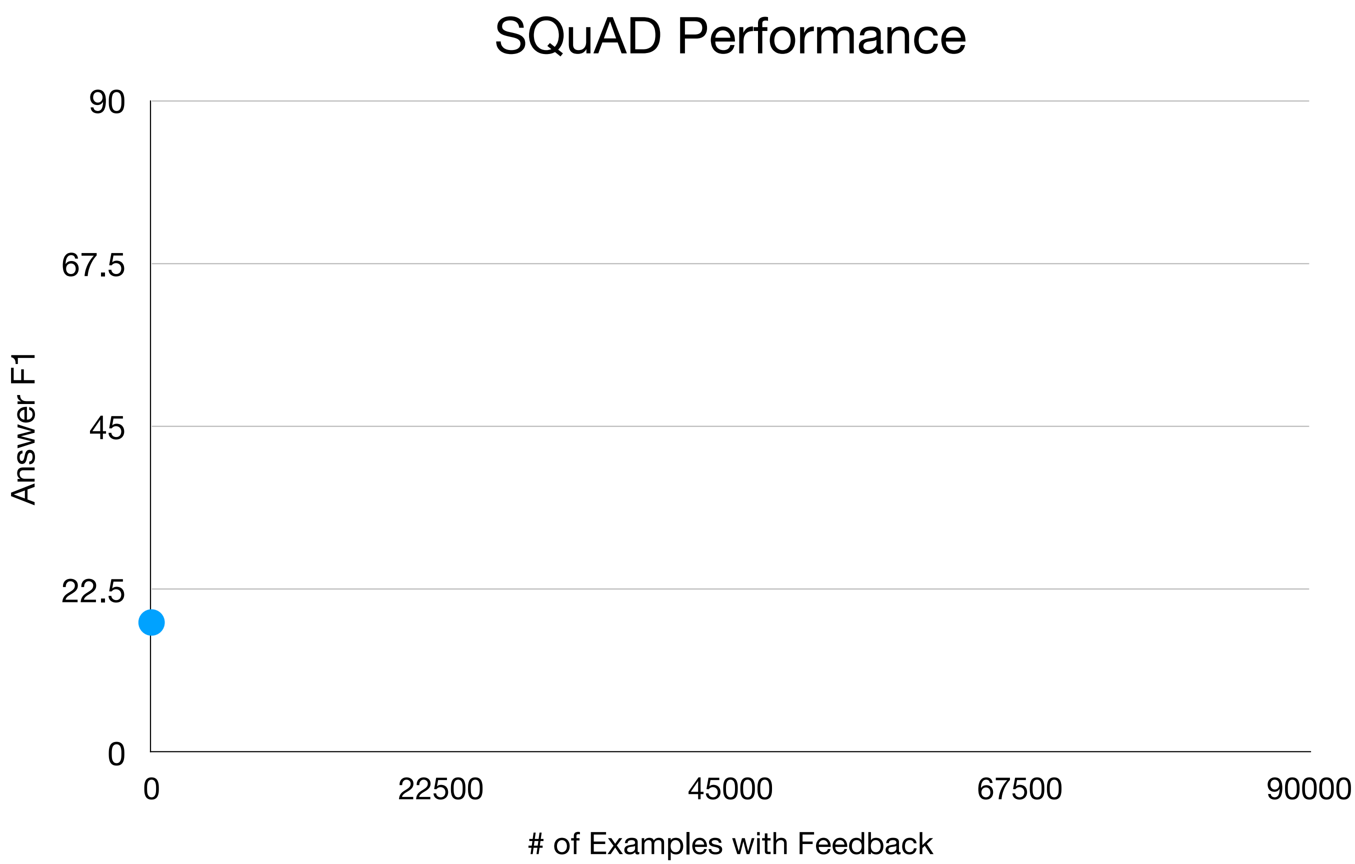
Cornell Bowers CIS Computer Science
CORNELL TECH
TEXAS The University of Texas at Austin

Research Question

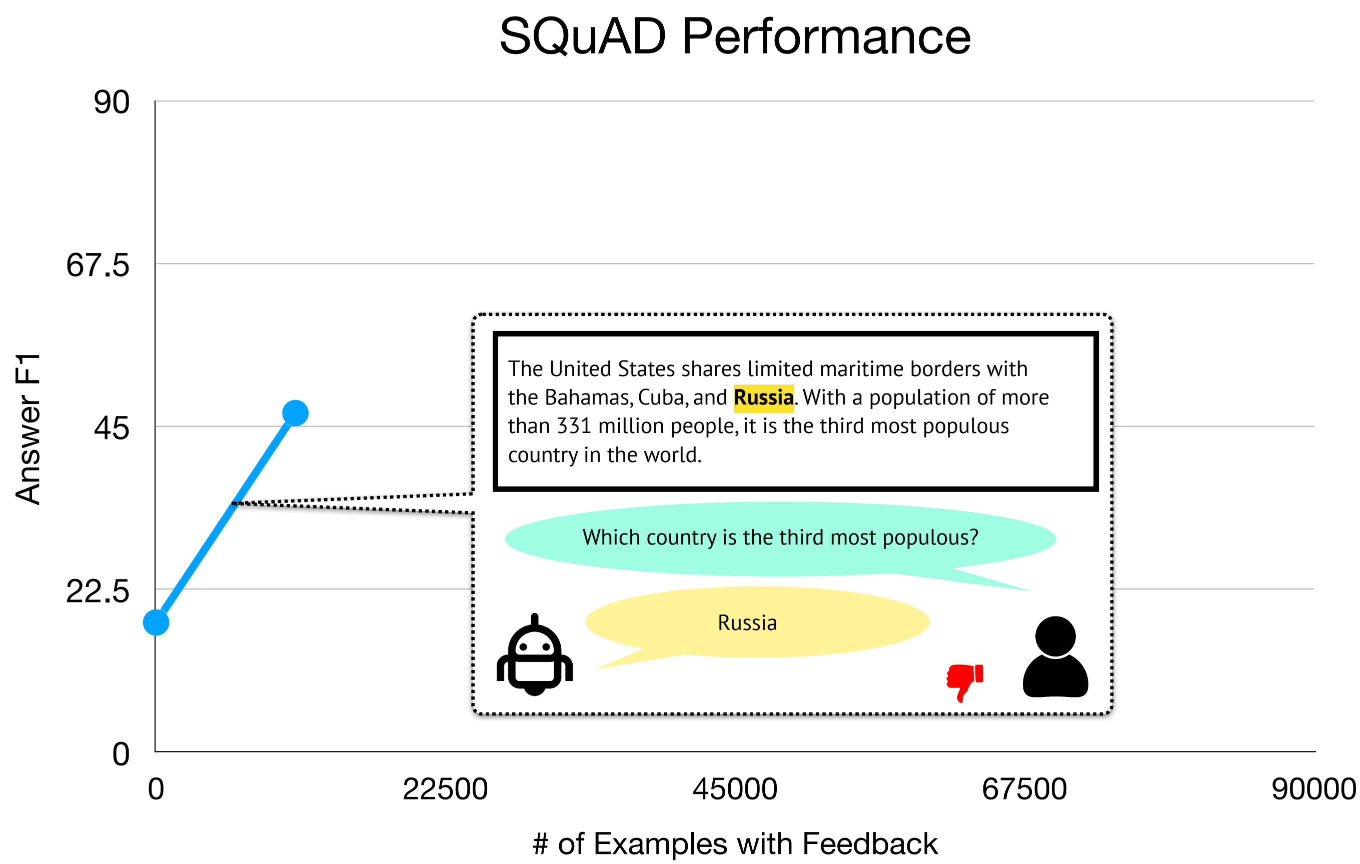
- **How to continually improve NLP systems by learning from interaction with users?**

- This work:
  - NLP system: **extractive QA**
  - Interaction with users: **simulated binary user feedback based on supervised data**

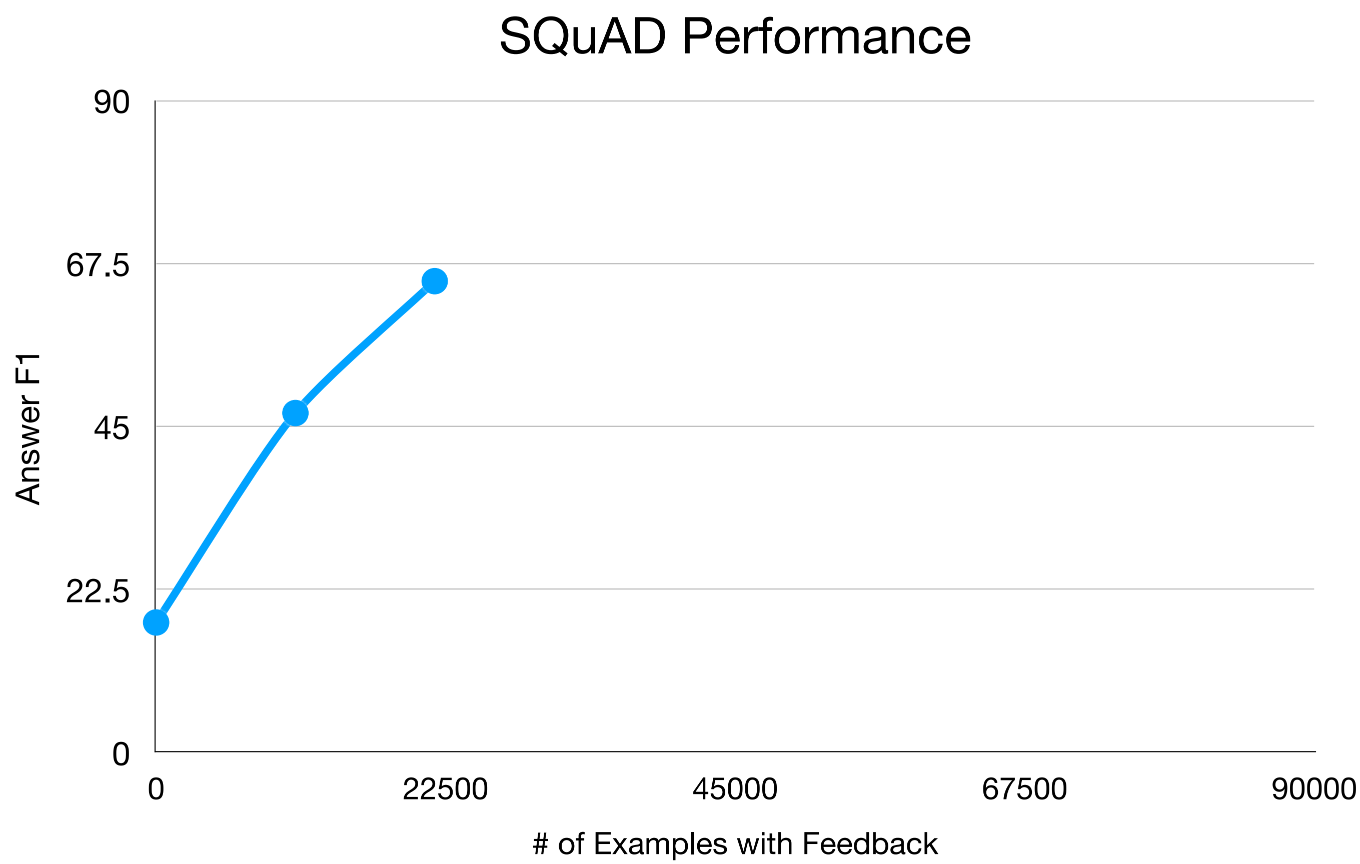
Learning from Feedback



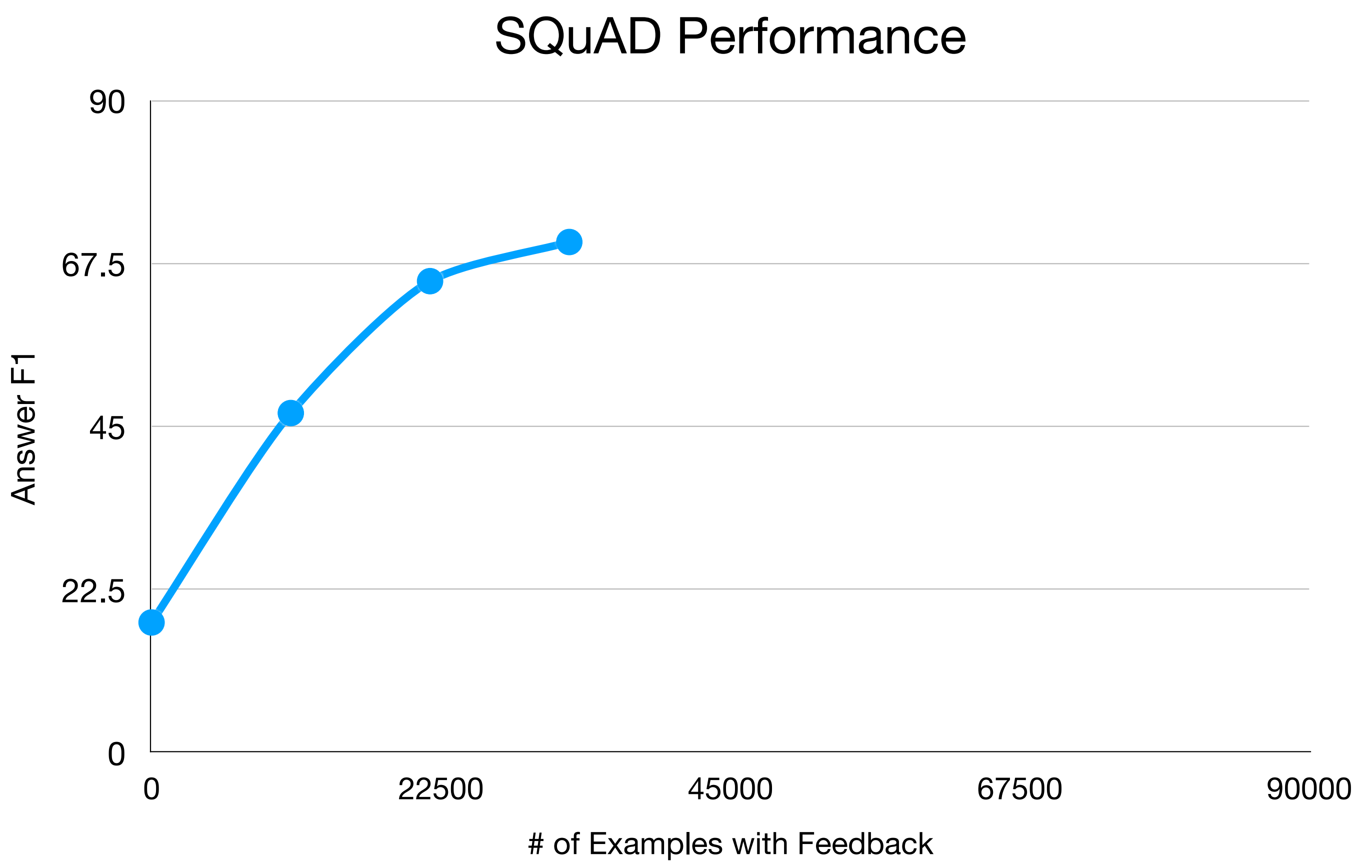
Learning from Feedback



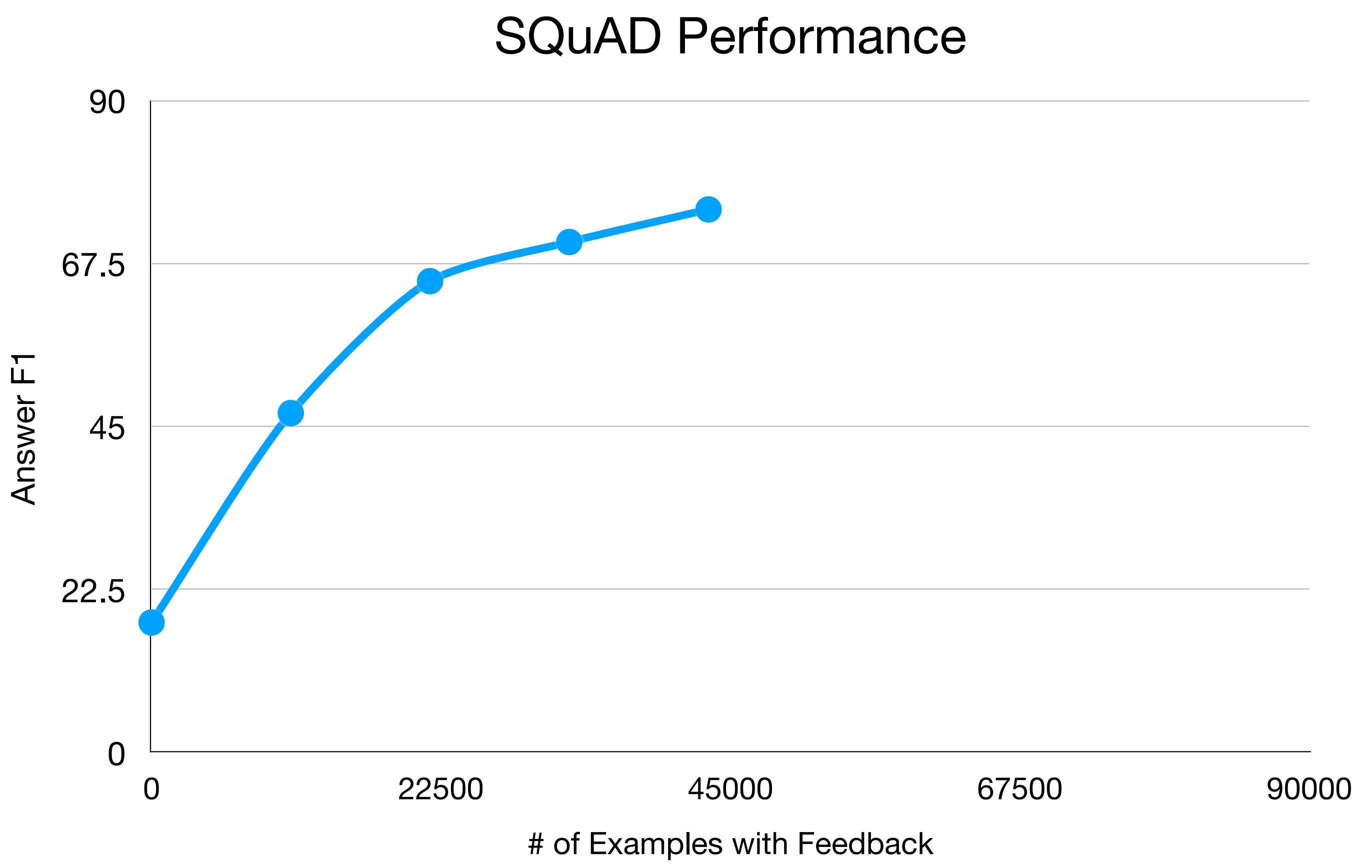
Learning from Feedback



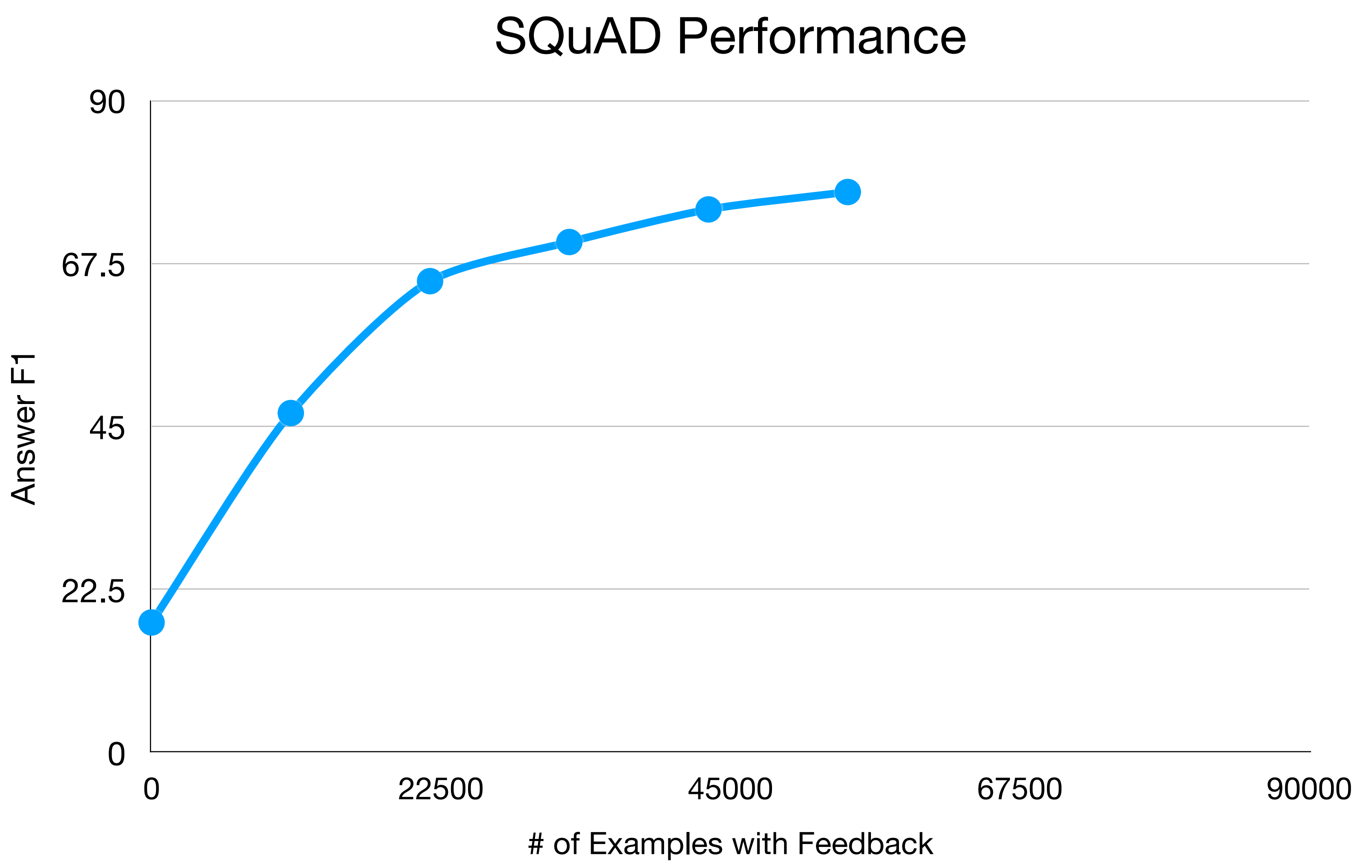
Learning from Feedback



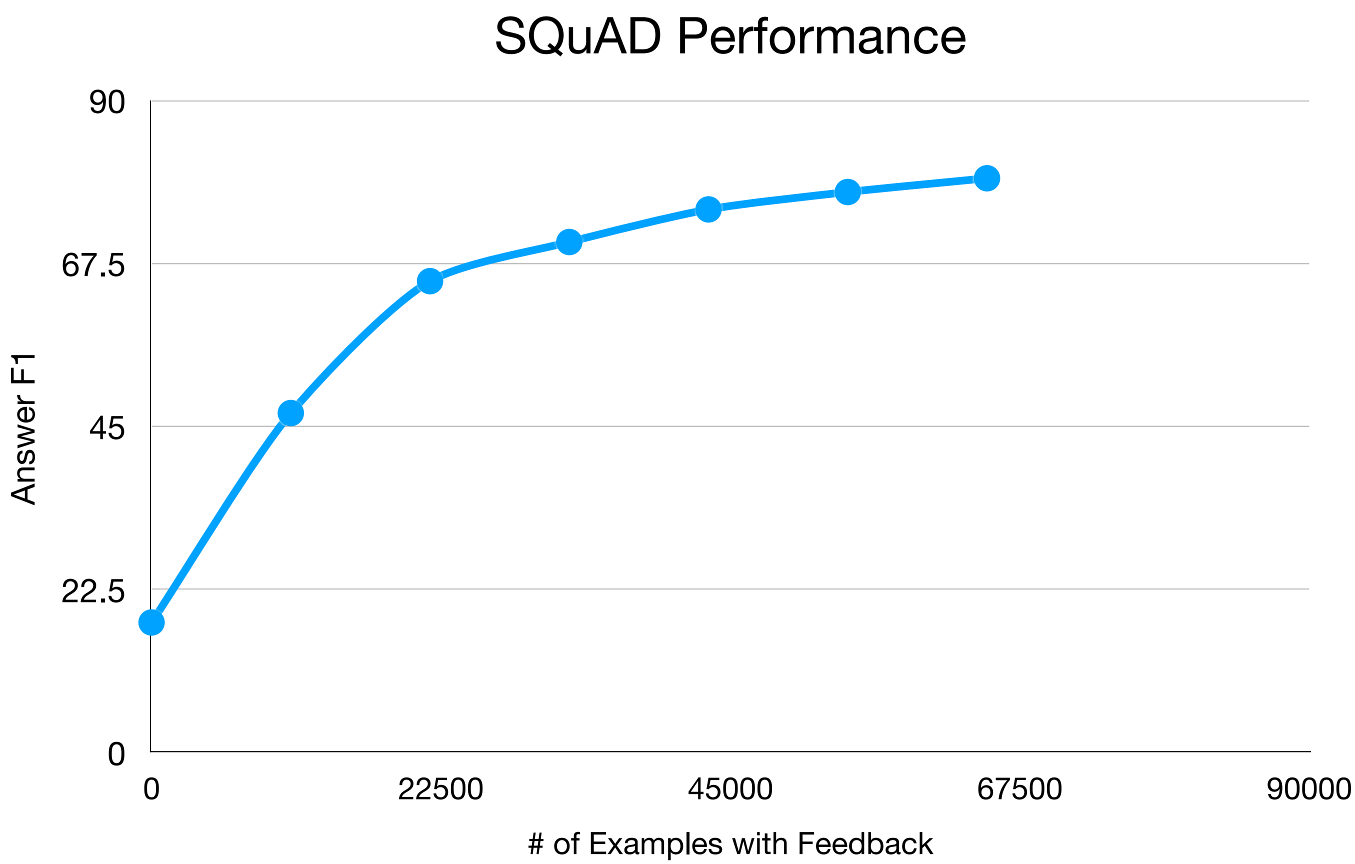
Learning from Feedback



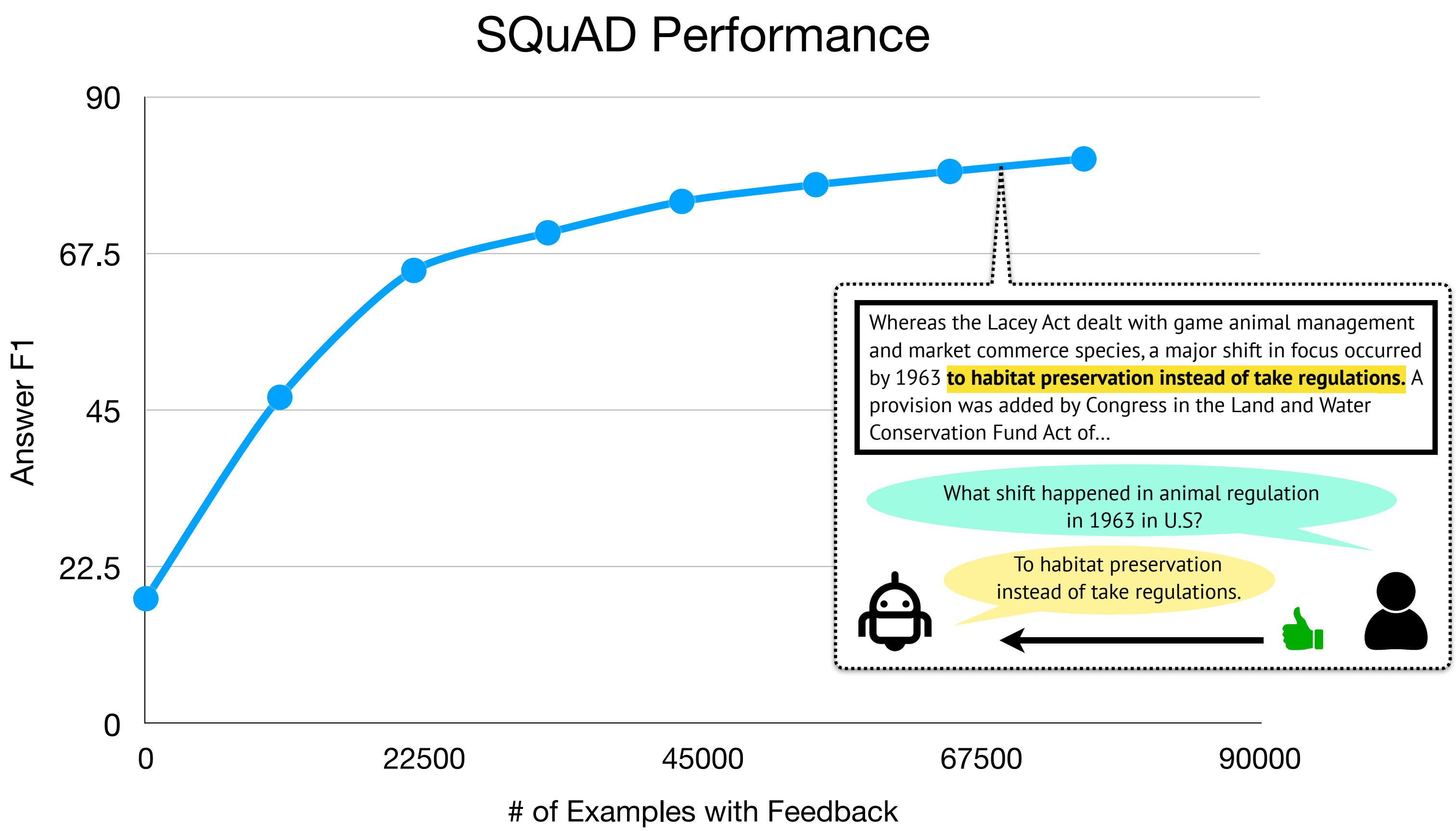
Learning from Feedback



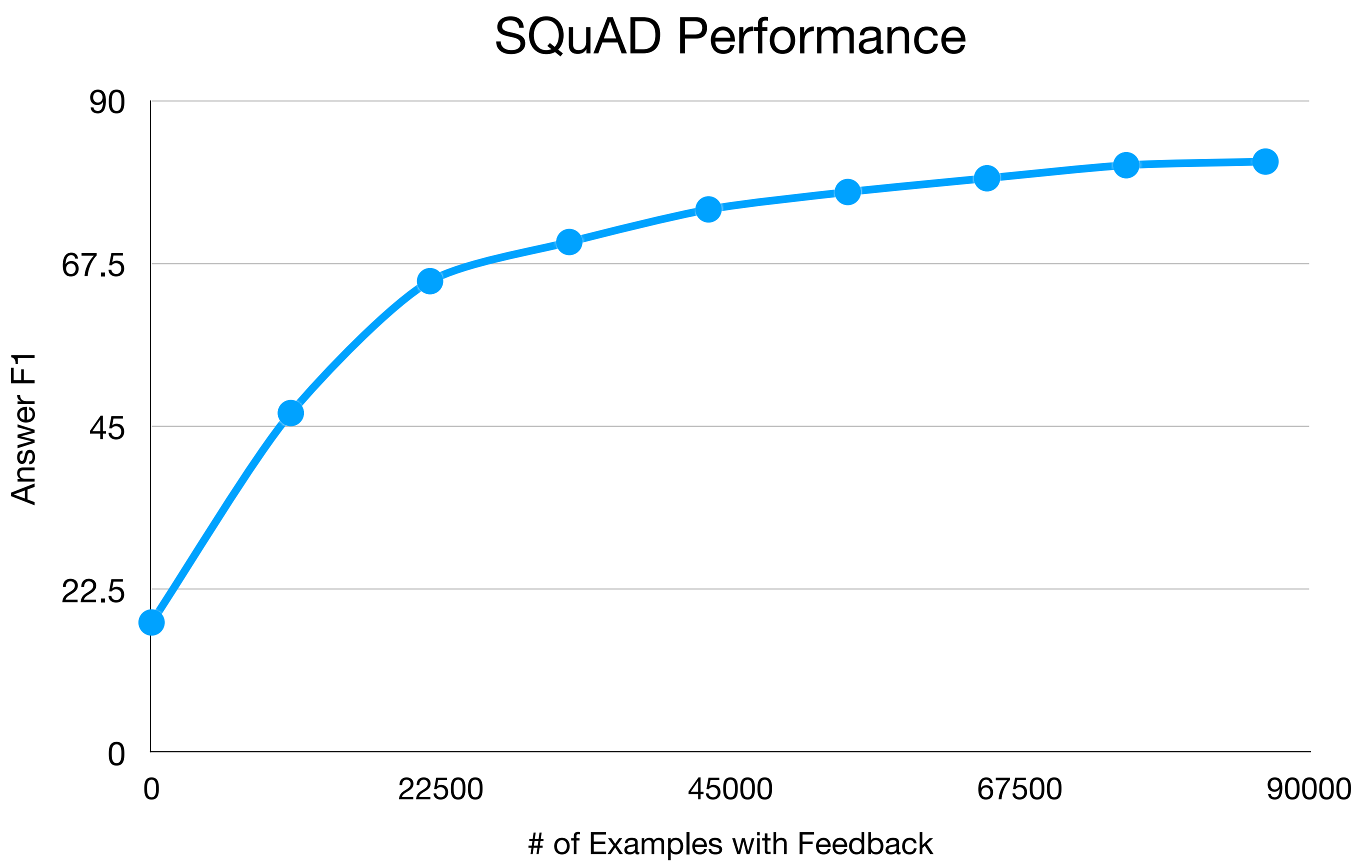
Learning from Feedback



Learning from Feedback



Learning from Feedback



Motivation

- **Why learn from user interactions?**

Training data

In-deployment Interaction Data

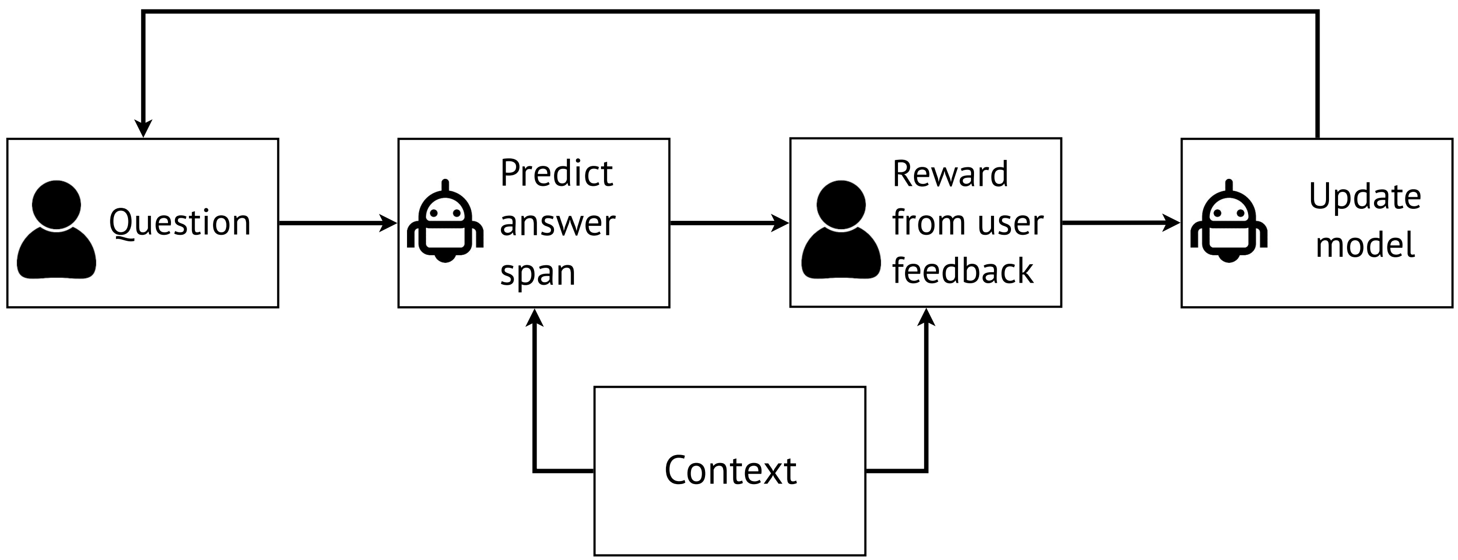
  - Reduce data collection costs and avoid artifacts
  - Enable improvement during deployment
  - No distributional shift between training and deployment
  - Systems evolves over time as the world changes

Motivation

- **Why learn from user interactions?**

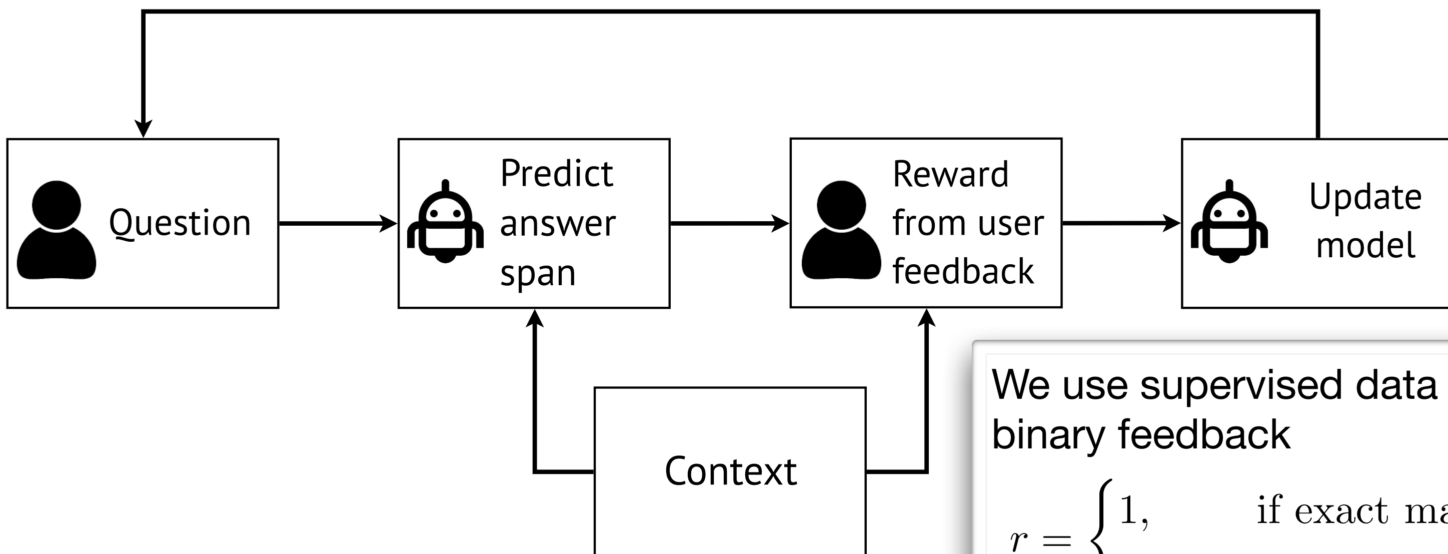
  - Reduce data collection costs and avoid artifacts
  - Enable improvement during deployment
  - No distributional shift between training and deployment
  - Systems evolves over time as the world changes

Contextual Bandit Learning
Online Setup



Objective: maximize total expected immediate reward

Contextual Bandit Learning
Online Setup



We use supervised data to simulate binary feedback

$$r = \begin{cases} 1, & \text{if exact match index-wise} \\ -0.1, & \text{otherwise} \end{cases}$$

Objective: maximize total expected <u>immediate</u> reward

Learning Algorithm

- We use **policy gradient**

QA model

Question

Context

$$r \nabla_\theta \log \pi_\theta(\hat{y} \mid \bar{q}, \bar{c})$$

Model-predicted answer

Reward

- Equivalent to REINFORCE except that we use <u>argmax</u> to predict answers instead of sampling

Experimental Setup

- Data: 6 English datasets using Wikipedia, news, and web texts [MRQA versions]
- Evaluation metric: token-level F1
- Model: SpanBERT-base [Joshi et al., 2020]
- Experiments:

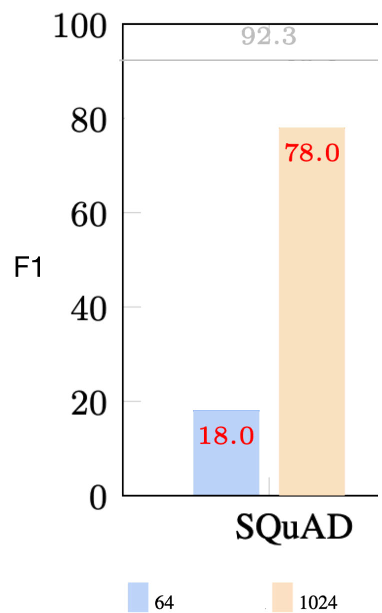
  1. **In-domain simulation**: Little in-domain supervised data

  2. **Domain adaptation**: Abundant out-of-domain supervised data

In-Domain Learning

Supervised training performance

Initial model performance



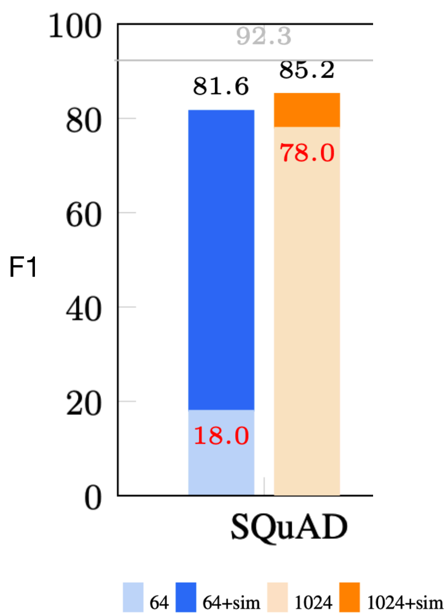
- Train an initial model on a small amount of supervised examples: 64 or 1024

In-Domain Learning

Supervised training performance

Simulation performance

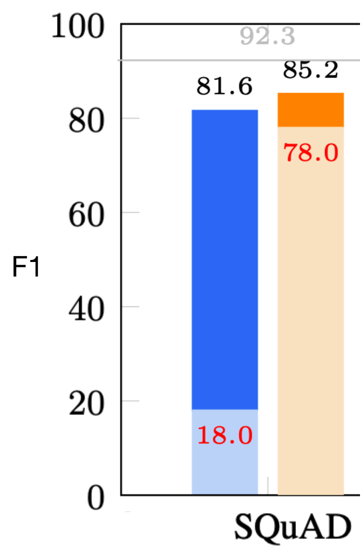
Initial model performance

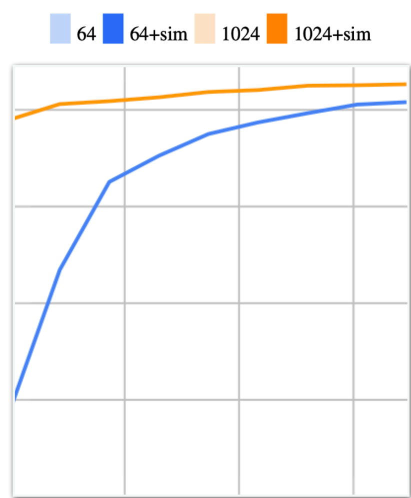


- Train an initial model on a small amount of supervised examples: 64 or 1024
- Simulation: receive rewards and update the model on the fly

In-Domain Learning

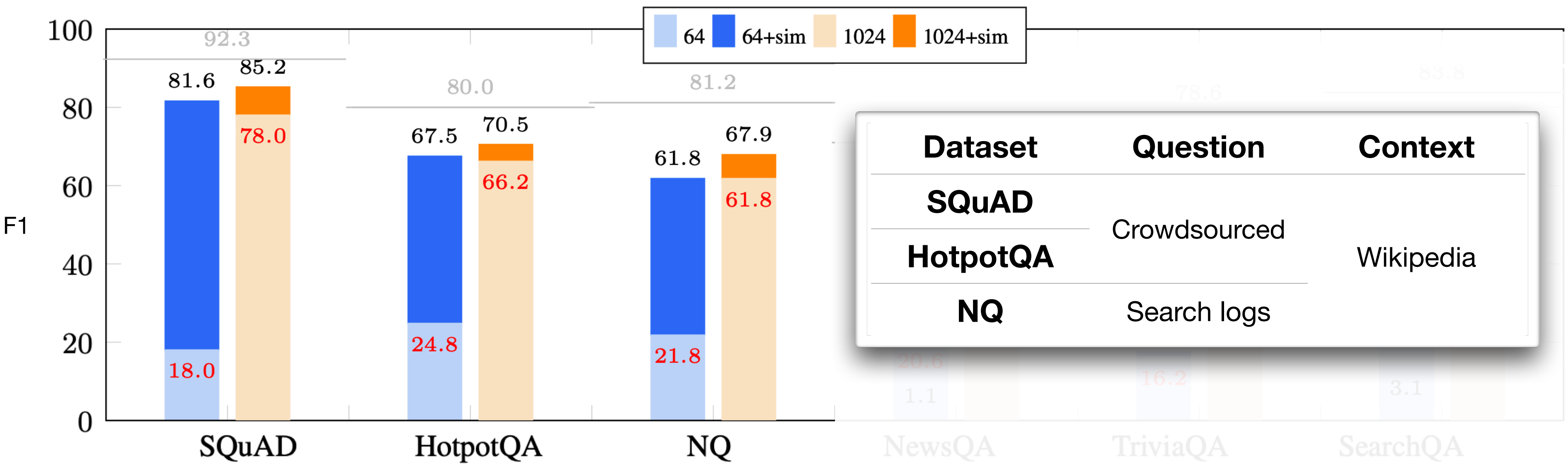
- Works well on SQuAD: performance gains
- Stable learning progression with much of the learning happening early





In-Domain Learning

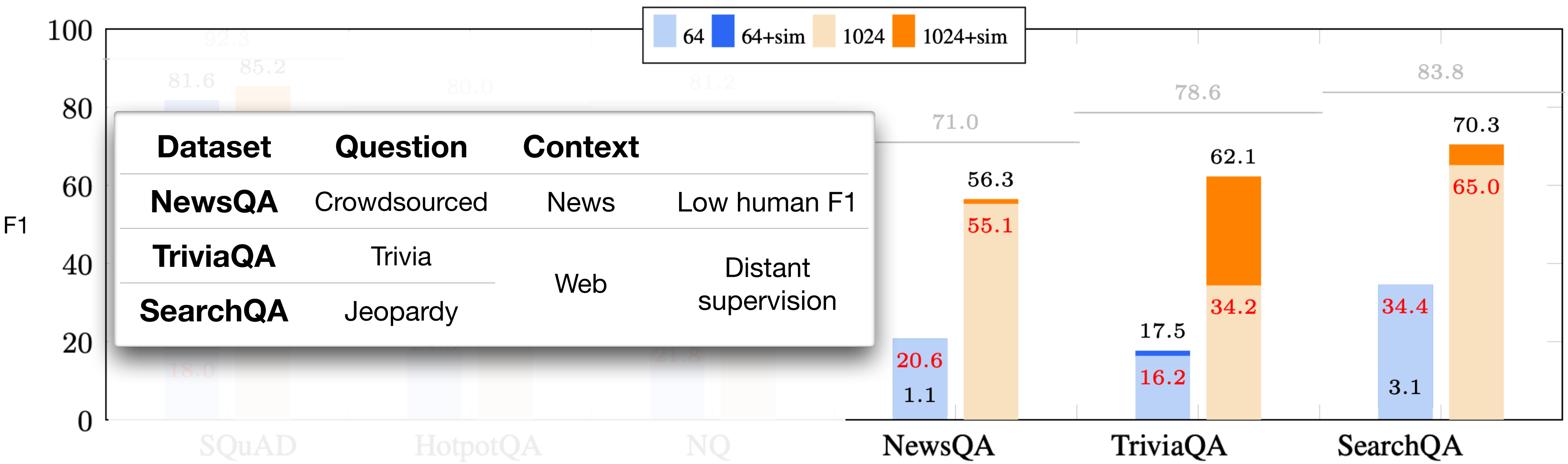
- Consistent performance gains on Wikipedia datasets
- Large gains with weaker initial models



Dataset	Question	Context
SQuAD	Crowdsourced	Wikipedia
HotpotQA	Crowdsourced	Wikipedia
NQ	Search logs	Wikipedia

In-Domain Learning

- Consistent performance gains on Wikipedia datasets
- Inconsistent with weaker initial models on challenging/noisy datasets



Dataset	Question	Context	
NewsQA	Crowdsourced	News	Low human F1
TriviaQA	Trivia	Web	Distant supervision
SearchQA	Jeopardy		

Domain Adaptation

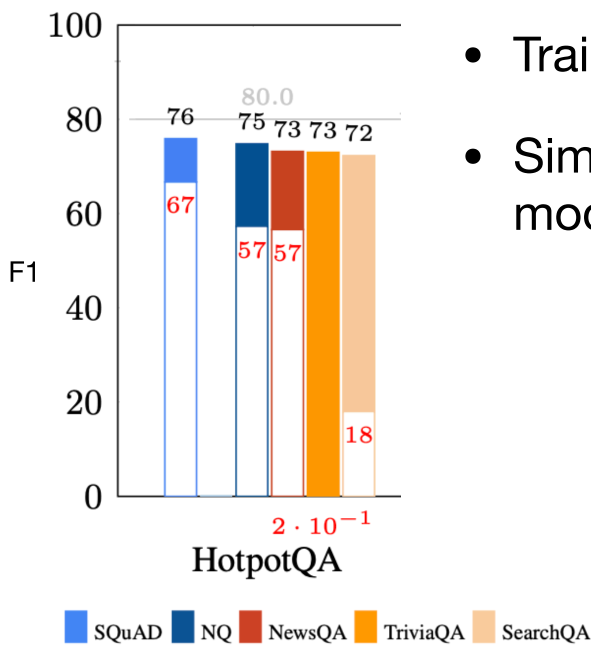
- So far: little in-domain data for initialization and continual bandit learning
- But: what if there is no data for the target domain at all?

Domain Adaptation

Supervised training performance

Simulation performance

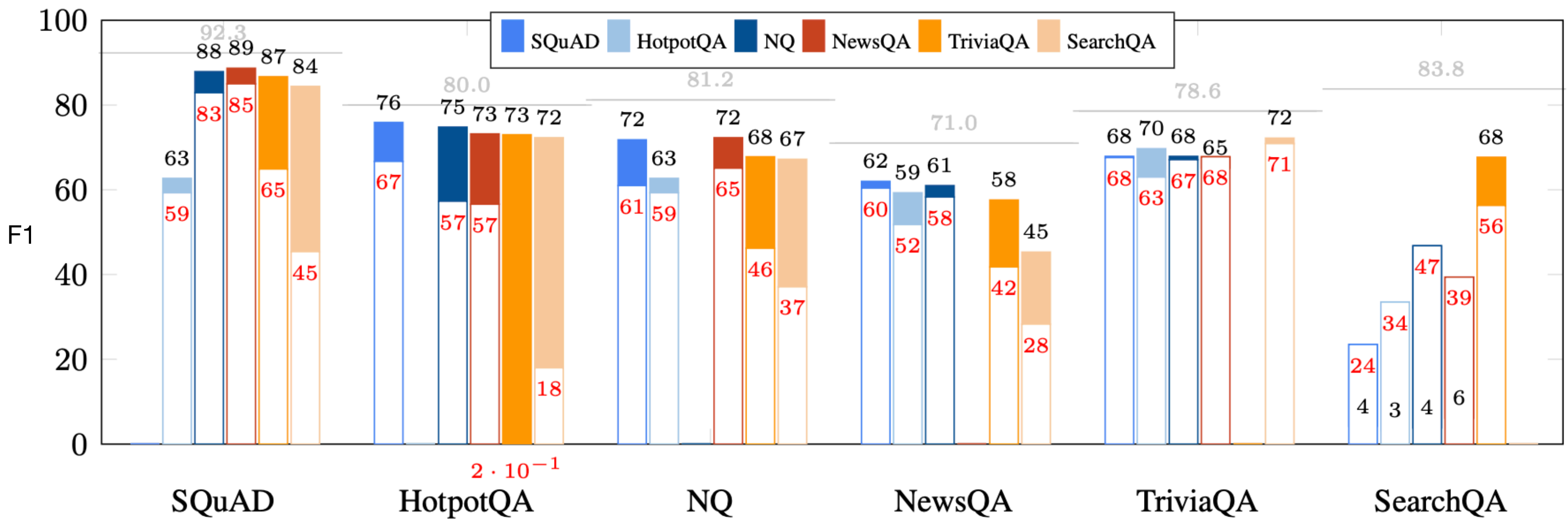
Initial model performance



- Train an initial model on <u>an existing dataset</u>
- Simulation: receive rewards and update the model on the fly <u>in the target domain</u>

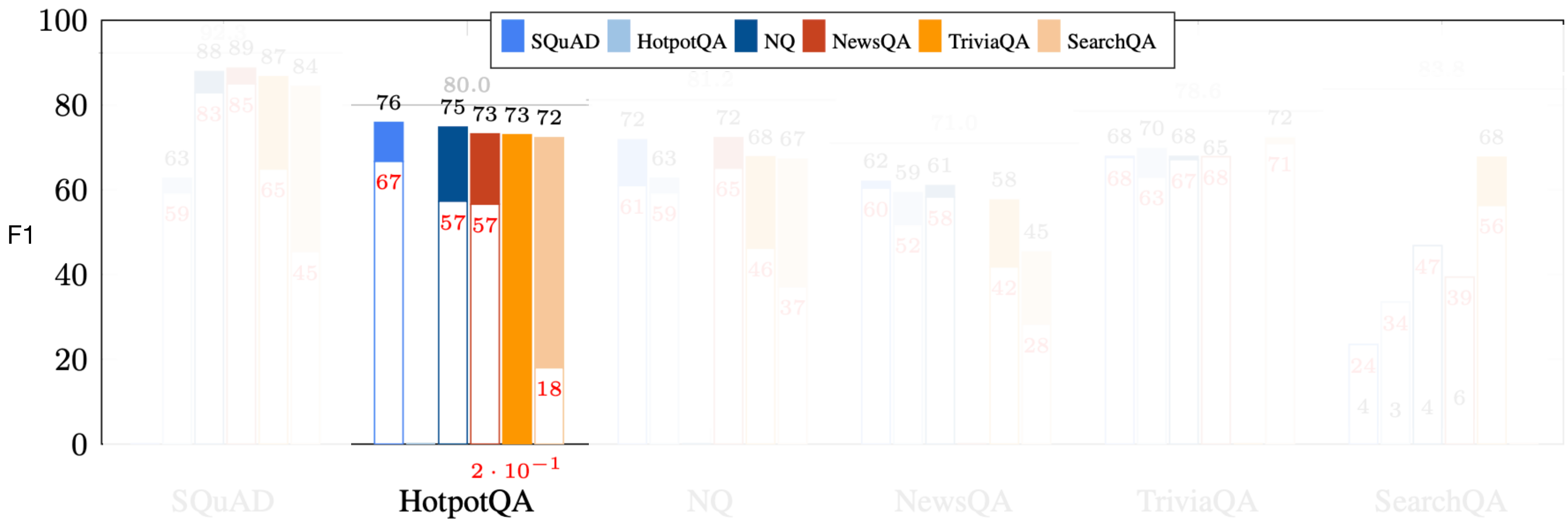
Domain Adaptation

- Performance gains on 22/30 configurations



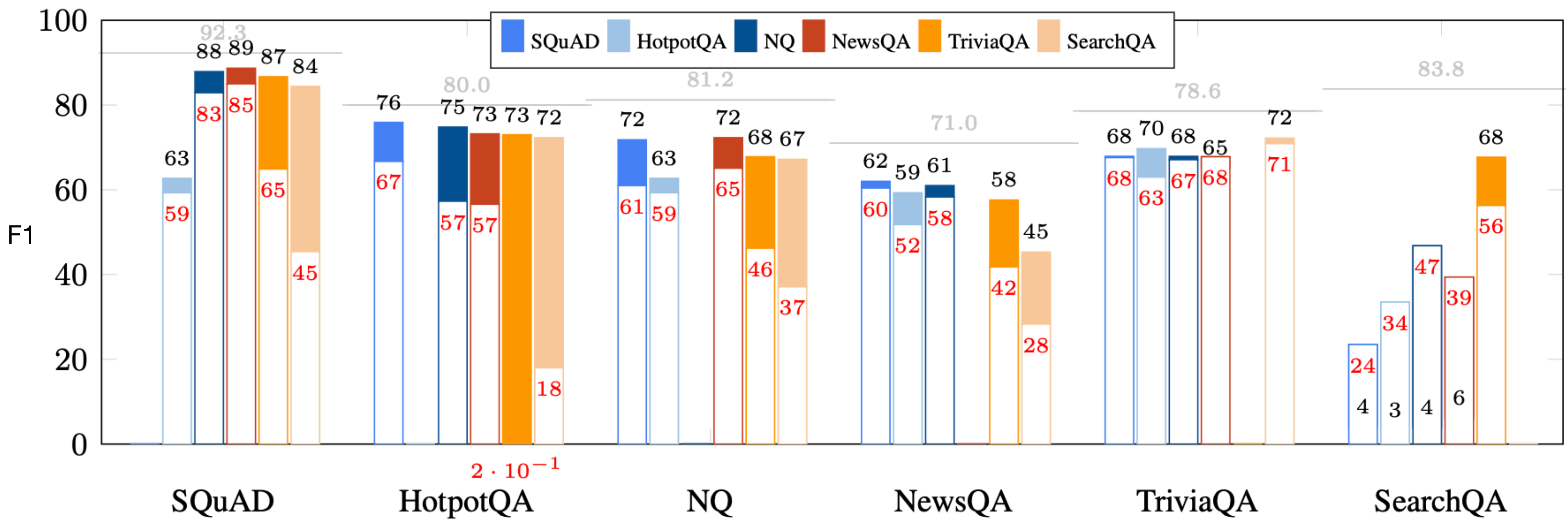
Domain Adaptation

- Performance gains on 22/30 configurations
- Extrapolate well particularly on HotpotQA from TriviaQA



Domain Adaptation

- Performance gains on 22/30 configurations
- Extrapolate well particularly on HotpotQA from TriviaQA
- Less consistent adaptation to NewsQA, TriviaQA, and SearchQA



Related Work

- **Bandit learning for NLP**
  Structured prediction [Sokolov et al., 2016], semantic parsing [Lawrence and Riezler, 2018], machine translation [Skolov et al., 2017; Kreutzer et al., 2018a,b; Mendoncca et al., 2021], summarization [Gunasekara et al., 2021], intent recognization [Falke and Lehnen, 2021]

- **Alternative forms of supervision for QA**
  Fine-grained information [Dua et al., 2020; Khashabi et al., 2020a], binary feedback [Kratzwald et al., 2020; Campos et al., 2020]

- **Domain Adaptation for QA**
  Data augmentation [Yue et al., 2021], adversarial training [Lee et al., 2019], back-training [Kulshreshtal et al., 2021], exploiting small lottery subnetworks [Zhu et al., 2021]

Conclusion

- Formulate learning from user feedback for extractive QA as a contextual bandit problem
- Demonstrate the effectiveness of the learning signal through simulation studies, including for domain adaptation
- Much more in the paper: offline learning, noise sensitivity analysis, regret analysis, and more experiments and analysis of domain adaptation

[fin]