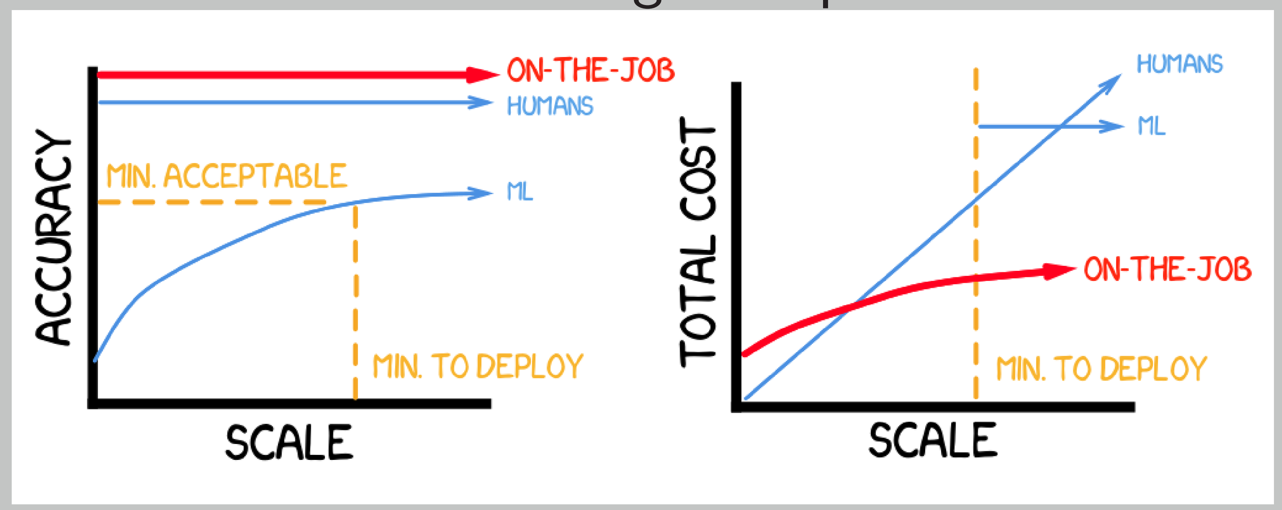
On-the-Job Learning with Bayesian Decision Theory

Keenon Werling   Arun Tejasvi Chaganty   Percy Liang   Christopher D. Manning

Department of Computer Science
Stanford University

Big picture

How do you deploy a high accuracy classifier starting with zero training examples?



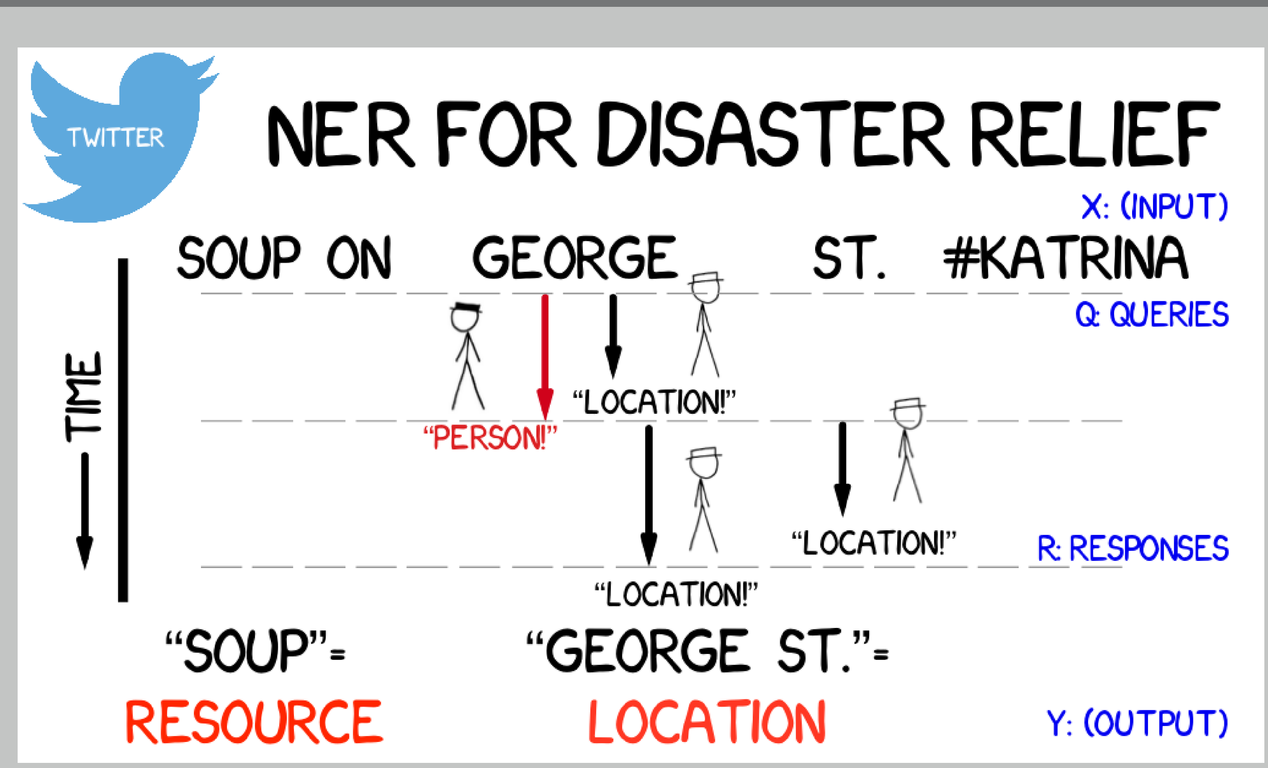
What is on-the-job learning?

- **On-the-job learning** allows a system to query the crowd for labels on the uncertain parts of an input as it arrives **before** making a prediction.
- Can **maintain accuracy on difficult examples** by asking the crowd for assistance.
- **Reduces costs** on simpler examples by learning a better prediction model online (on-the-job).
- User specifies a base prediction model and how to trade off accuracy, cost and latency.
- System optimizes for utility using ideas from game playing and Bayesian decision theory.

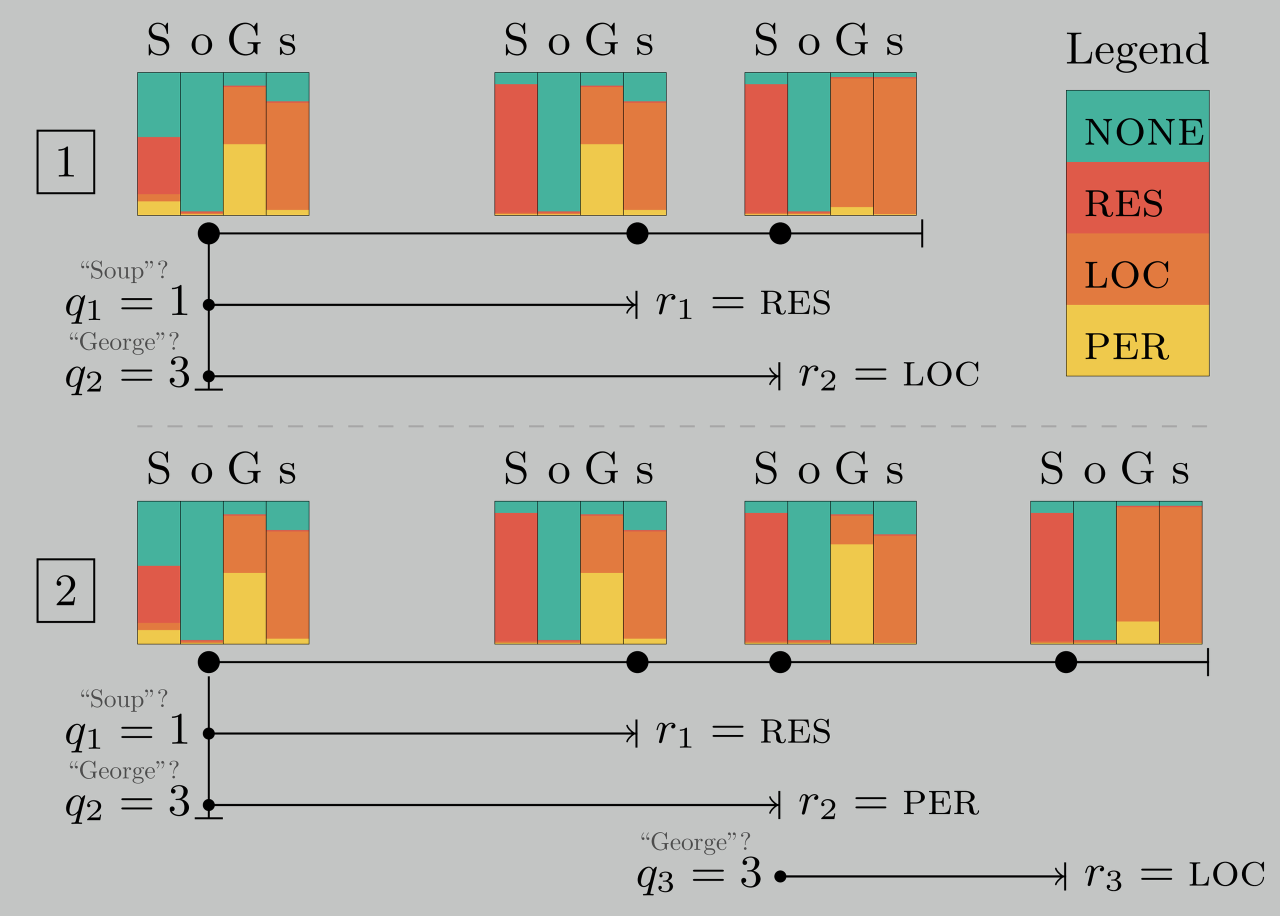
Related work

Area	Paradigm
Online active learning chooses the most informative examples to label *after* classification. Impossible to maintain high accuracy initially.	INPUT · PREDICT · LABEL · LEARN
Active classification learns a static policy from a labelled dataset to choose features to query at test time.	INPUT · LABEL* · PREDICT
On-the-job learning combines advantages of both the above methods. Note, Legion:AR (?) studied the user interface aspects of on-the-job learning, while we study the machine learning aspects of it.	INPUT · LABEL · PREDICT · LEARN

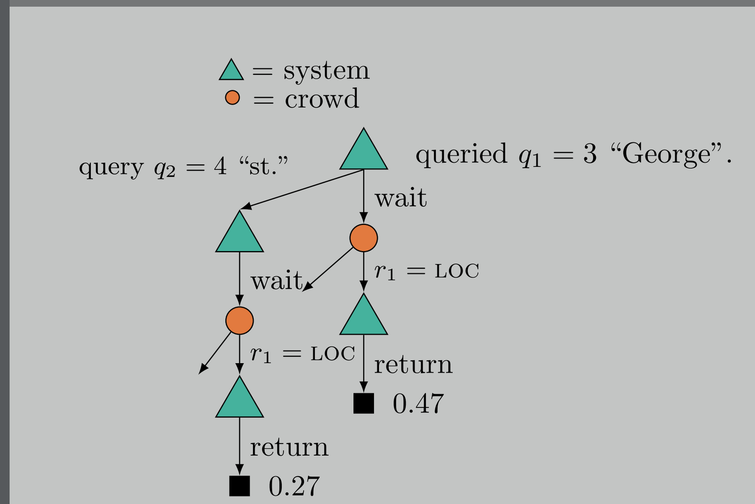
Example: named entity recognition on tweets



How marginals evolve after incorporating responses



Approximating utility with MCTS



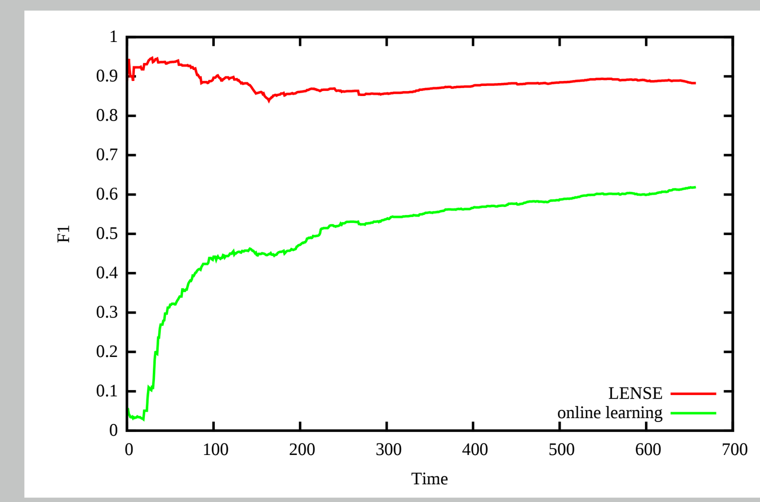
- **Stochastic game** between system and crowd.
- **States** capture time, queries in flight and received responses.
- **Actions** are querying for a label, waiting or returning current best guess.
- The system chooses actions that **maximize utility**.
- Approximated by Markov Chain Tree Search (MCTS) with progressive widening, using an environment model.

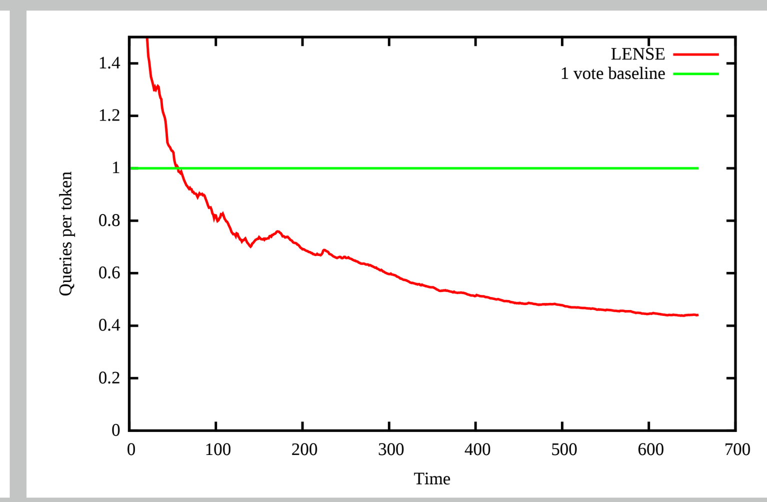
$$\underbrace{p(\mathbf{y}, \mathbf{r}, \mathbf{t} \mid \mathbf{x}, \mathbf{q}, \mathbf{s})}_{\text{environment model}} \triangleq \underbrace{p_\theta(\mathbf{y} \mid \mathbf{x})}_{\text{prediction}} \prod_{i=1}^{k} \underbrace{p_{\mathrm{R}}(r_i \mid y_{q_i})}_{\text{annotator noise}} \underbrace{p_{\mathrm{T}}(t_i \mid s_i)}_{\text{latency}}.$$

- Use human-labelled examples as training data to learn the model.

Named Entity Recognition (CoNLL 2003)

System	Latency/tok	Qs/tok	PER F_1	LOC F_1	ORG F_1	F_1
1-vote	467 ms	1.0	90.2	78.8	71.5	80.2
3-vote	750 ms	3.0	93.6	85.1	74.5	85.4
5-vote	1350 ms	5.0	**95.5**	87.7	78.7	87.3
Online	n/a	n/a	56.9	74.6	51.4	60.9
Threshold	414 ms	0.61	95.2	**89.8**	79.8	88.3
LENSE	**267 ms**	**0.45**	95.2	89.7	**81.7**	**88.8**

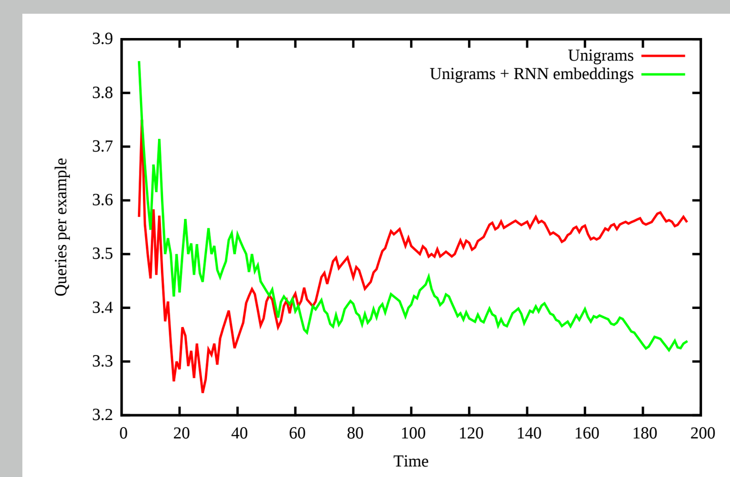




Takeaway

On-the-job learning is capable of making consistently accurate predictions while reducing annotation costs.

Sentiment



System	Latency	Qs/ex	Acc.
5-vote	13.5 s	5.00	98.7
UNIGRAMS			
Online	n/a	n/a	78.1
Threshold	10.9 s	2.99	95.9
LENSE	11.7 s	3.48	98.6
RNN			
Online	n/a	n/a	85.0
Threshold	11.0 s	2.85	96.0
LENSE	11.0 s	3.19	98.6

Takeaway

On-the-job learning will maintain accuracy even if the model lacks the capacity to.

Conclusions and Future Work

- Consider on-the-job learning to get accurate labels on your next project for cheap.
- Easy to use open-source implementation, LENSE, available!
- Future directions include improving confidence estimation, learning from measurements and more applications.