Learning How to Actively Learn: A Deep Imitation Learning Approach

Ming Liu

Joint Work with: Wray Buntine and Reza Haffari
Monash University, Australia

{ming.m.liu, wray.buntine, gholamreza.haffari} @monash.edu

Roadmap

- Introduction to active learning (AL)
- Markov decision process (MDP) for agent-based AL
- Deep imitation learning to train the AL policy
- Experiments & Analysis

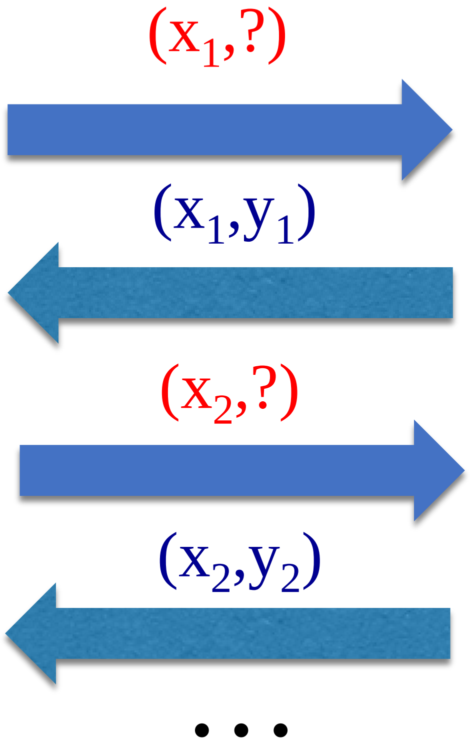
Roadmap

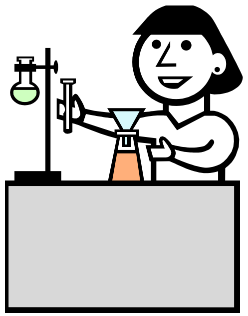
- Introduction to active learning (AL)
- Markov decision process (MDP) for agent-based AL
- Deep imitation learning to train the AL policy
- Experiments & Analysis

Introduction

Raw unlabeled data points $x_1, x_2, \ldots$

Classifier

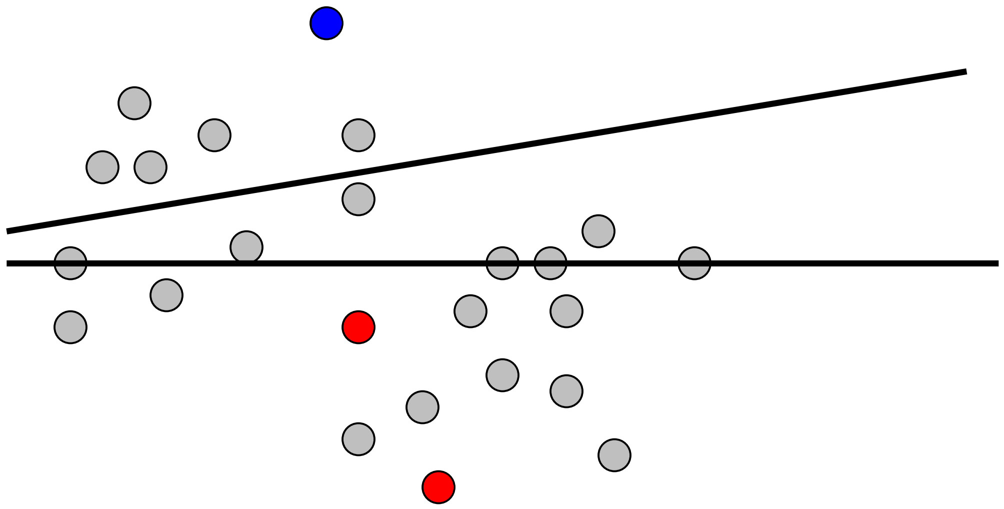




Oracle/Expert:
Provides labels for queries

Introduction

- At any time during the AL process, we have a current guess for the classifier



- AL Strategy: Query the point closest to the decision boundary

Introduction

Warnings:

- Not clear whether heuristics lead to optimal querying behavior
- Not clear which hard coded heuristic is good for a task at hand

AL Heuristics $(x_1,?)$

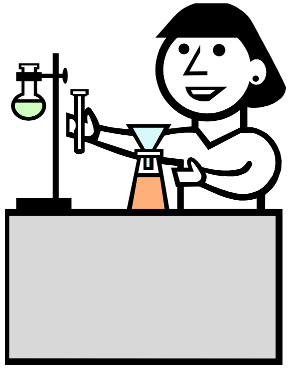
(x_1,y_1)

AL Heuristics $(x_2,?)$

(x_2,y_2)

. . .

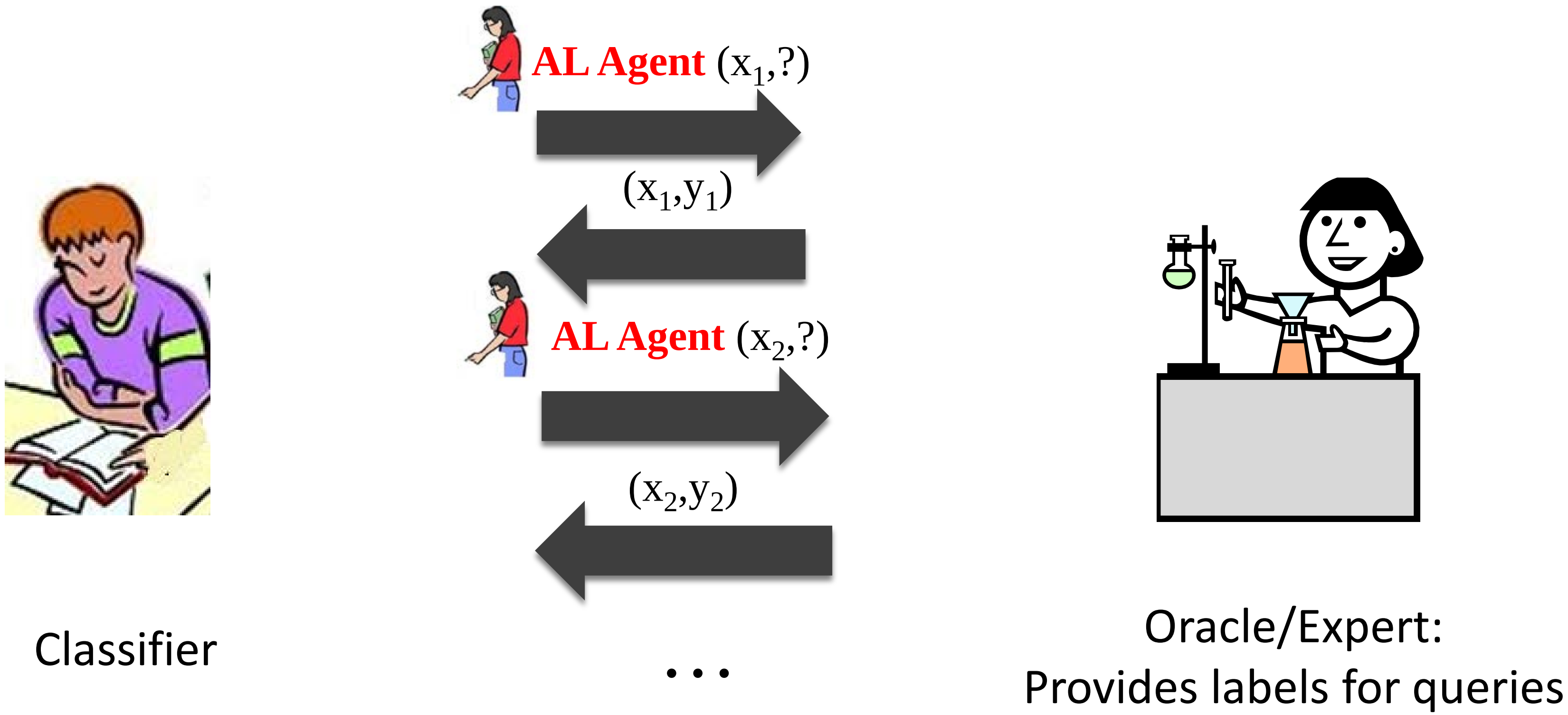
Classifier



Oracle/Expert:
Provides labels for queries

Introduction

Can we learn the best active learning strategy ?



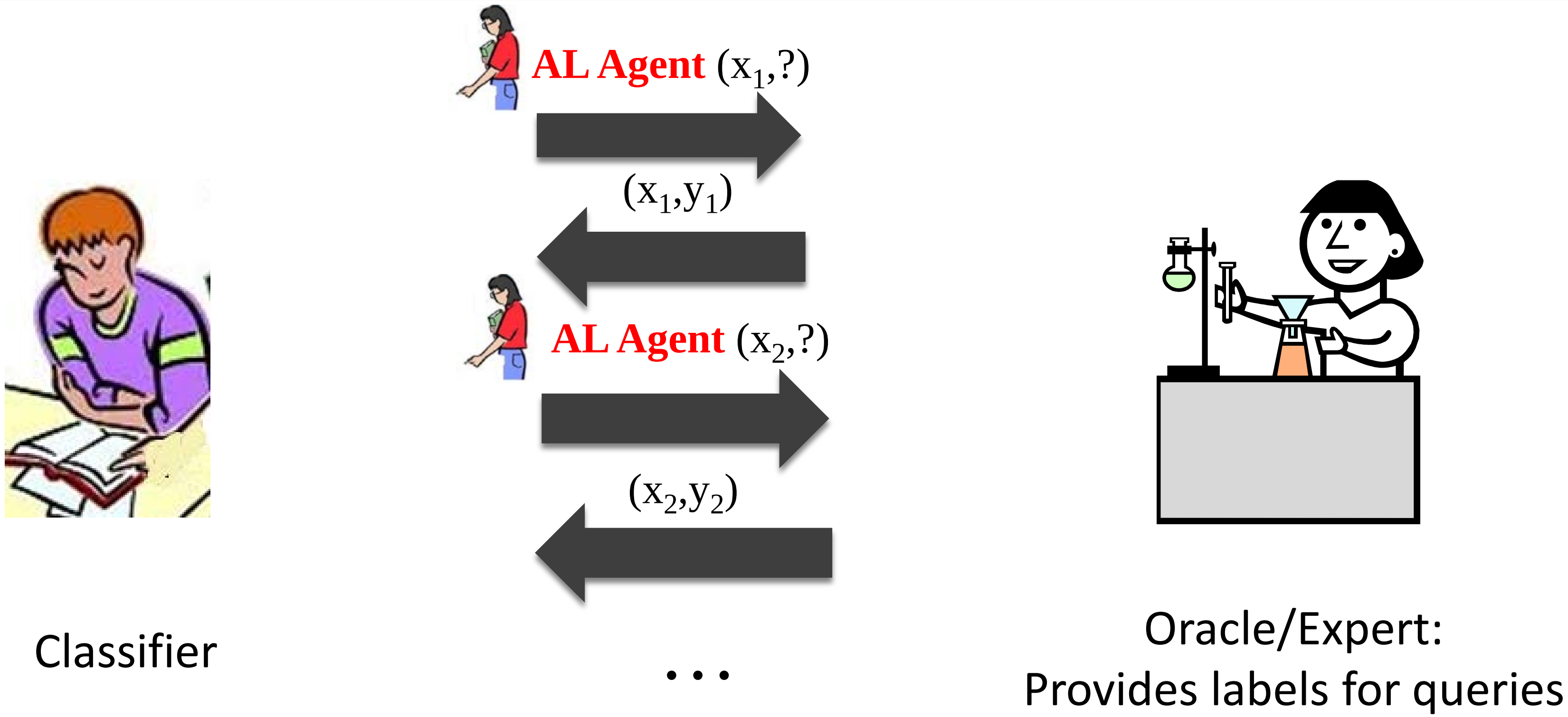
Roadmap

- Introduction to active learning (AL)
- Markov decision process (MDP) for agent-based AL
- Deep imitation learning to train the AL policy
- Experiments & Analysis

Agent-based Active Learning

Need to train an AL agent to tell what data to select next, given

- the previously selected data
- the pool of unlabeled data available
- the underlying classifier, learned so far



AL Query Strategy by an Agent

Raw unlabeled data points $x_1, x_2, \ldots$

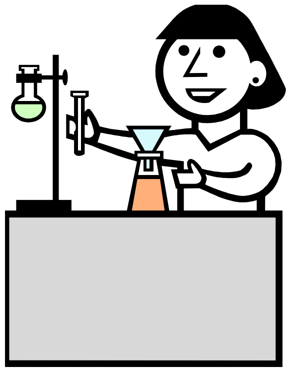
AL Agent $(x_1, ?)$

(x_1, y_1)

AL Agent $(x_2, ?)$

(x_2, y_2)

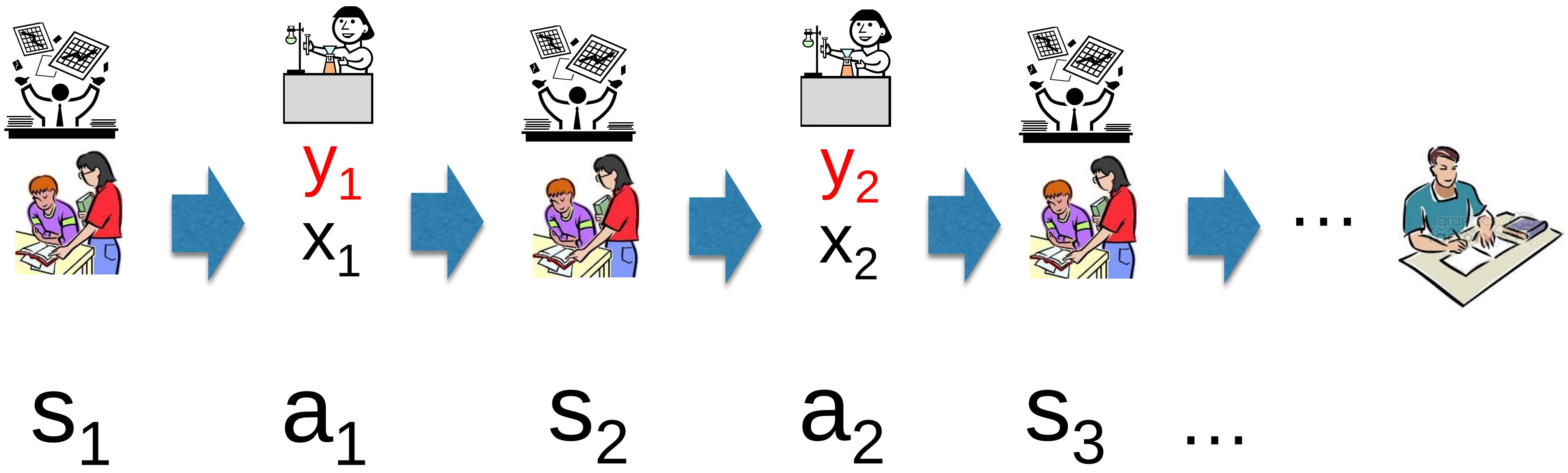
…



The Tutoring AL Agent & Learning Student (Classifier)

Oracle/Expert:
Provides labels for queries

Agent Operates in Markov Decision Process



Reward: Accuracy (

,

)

Learn the Optimal Query Policy

$$\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\sum_{t=1}^{\mathcal{B}} R(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})\right]$$

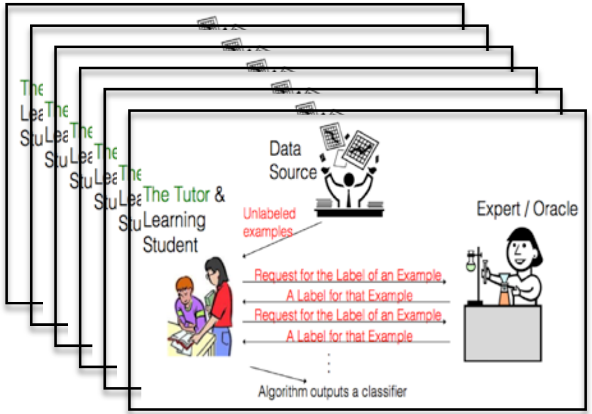
Roadmap

- Introduction to active learning (AL)
- Markov decision process (MDP) for agent-based AL
- Deep imitation learning to train the AL policy
- Experiments & Analysis

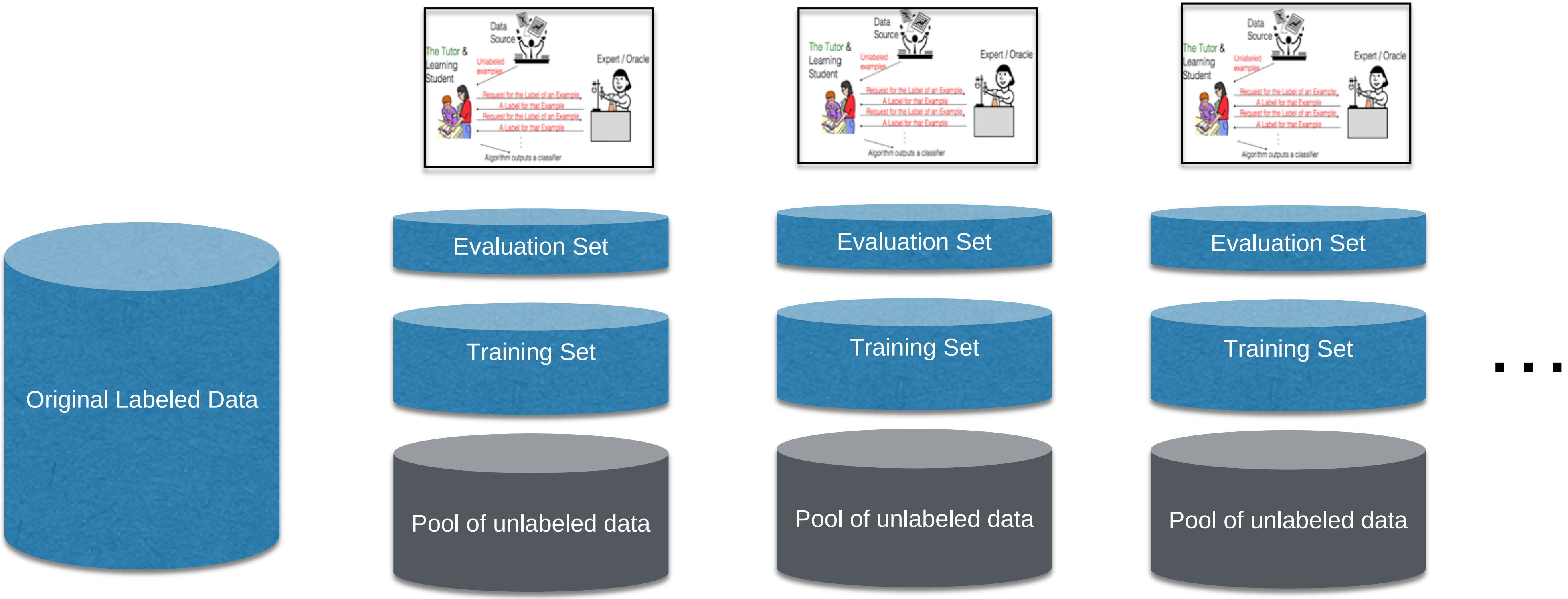
Training Agent’s Policy

- IDEA: Let’s train the agent based on AL simulation for a rich-data task and then transfer it to AL problem of interest

- This is Meta-Learning: Learning to Actively Learn
  - Synthesize many AL problems
  - Use Imitation/Reinforcement Learning algorithms



Synthesizing AL Problems

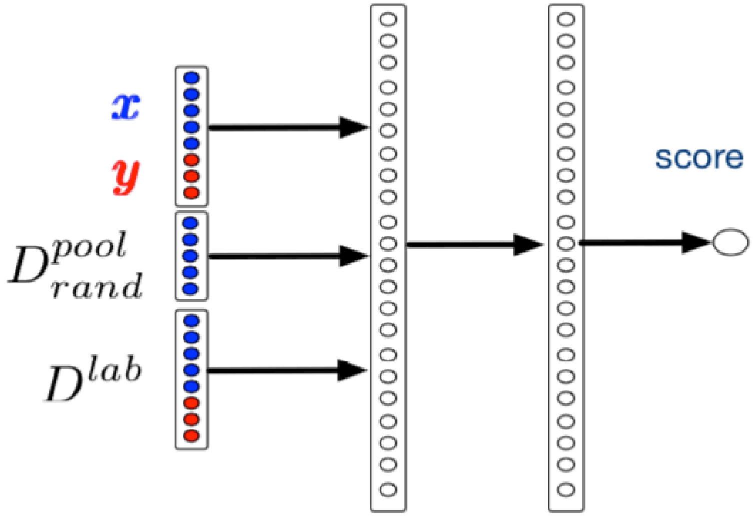


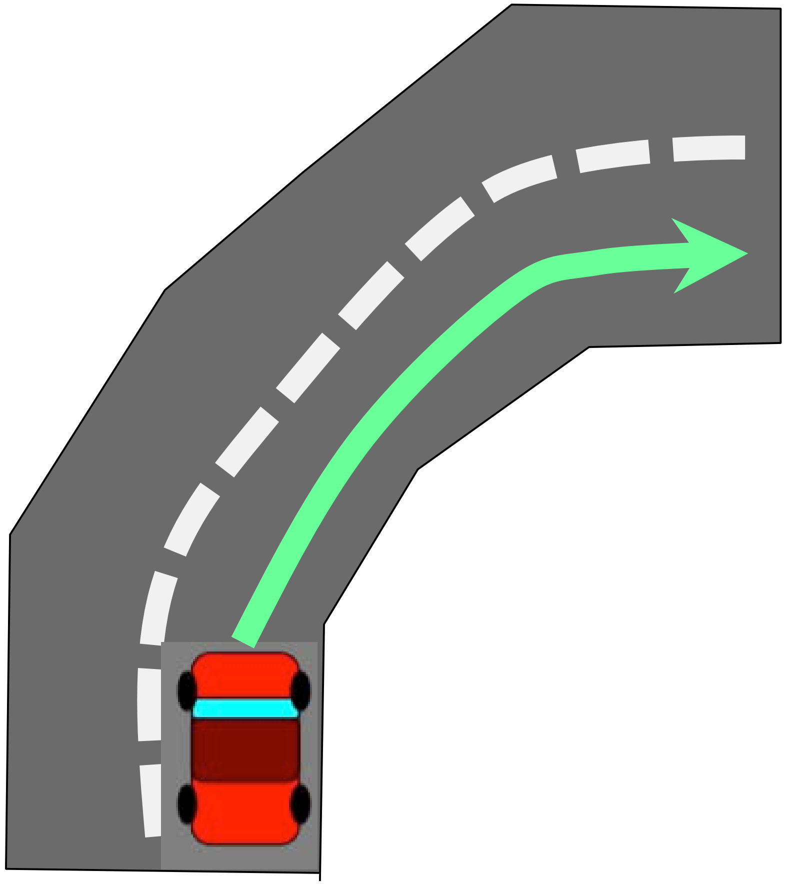
$$\mathbb{E}_{(D^{lab}, D^{unl}, D^{evl}) \sim \mathcal{D}} \left[\mathbb{E}_{\pi_{\boldsymbol{\theta}}} \left[\sum_{t=1}^{\mathcal{B}} R(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}) \right] \right]$$

Imitation Learning

- The algorithmic oracle gives the correct action in each world state

- Train the agent (policy network) to prefer the "correct" action compared to "incorrect" ones (i.e. classification)





Algorithmic Oracle

- It computes the correct action in each world state
  - Re-train the underlying model using all possible queries/actions
  - Mark the one leading to the most accurate prediction on the evaluation set

$\text{argmax}_{(xi,yi)\ \text{in Pool}}$ Accuracy (Retrain( , x_i,y_i) , Evaluation Set)

- Too slow for typical large pools of data
- IDEA: Randomly sample a subset and maximize over it
  - Leads to efficient training and effective learned policies

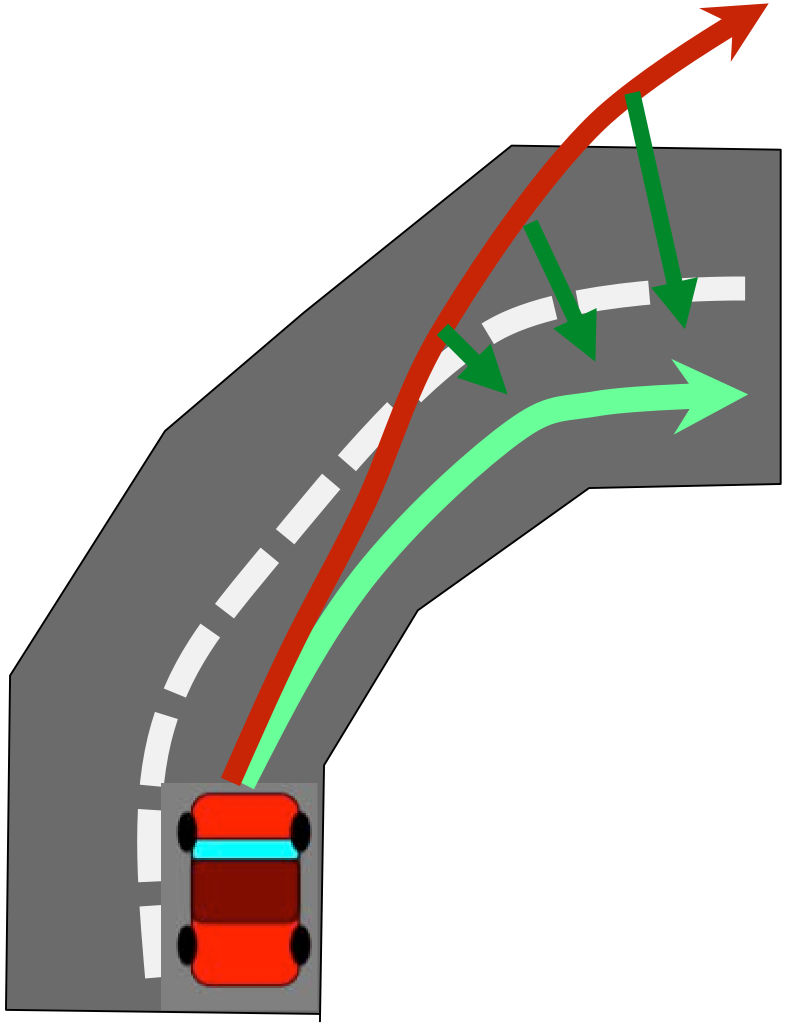
Imitation Learning DAGGER

- The collected state-action pairs are not i.i.d. hence problematic for classifier learning

- Data Aggregation (DAGGER): Once in a while, use the predicted action by the policy network during training (Ross et al 2011)

$$\pi_\tau = \beta_\tau \tilde{\pi}^* + (1 - \beta_\tau)\hat{\pi}_\tau$$

- This is to make sure the policy sees bad states and the correct action to recover from them in the training time



Roadmap

- Introduction to active learning (AL)
- Markov decision process (MDP) for agent-based AL
- Deep imitation learning to train the AL policy
- Experiments & Analysis

Experiments (Task 1: text classification)

- Sentiment Classification: Positive/Negative sentiment of a review
  - Train the AL policy on one product, and apply to the reviews of another

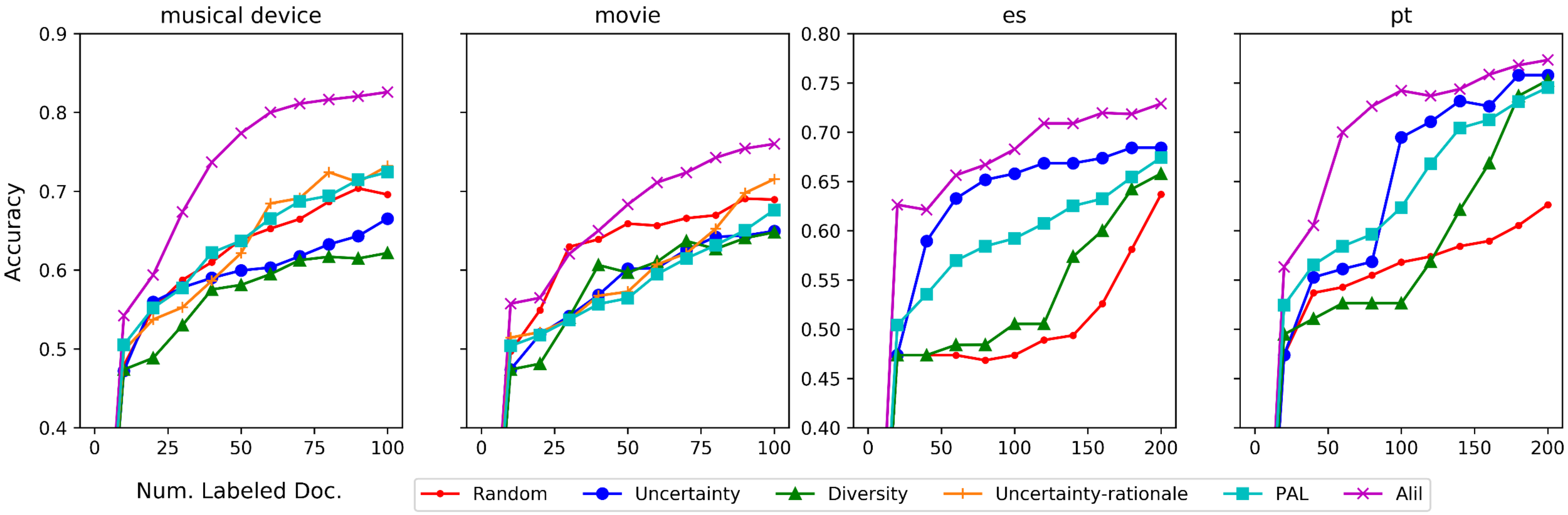
- Authorship Profiling: Gender of the author of a tweet
  - Train the AL policy on one language, and apply to another

		doc. (src/tgt)	
src	**tgt**	**number**	**avg. len. (tokens)**
elec.	music dev.	27k/1k	35/20
book	movie	24k/2k	140/150
en	sp	3.6k/4.2k	1.15k/1.35k
en	pt	3.6k/1.2k	1.15k/1.03k

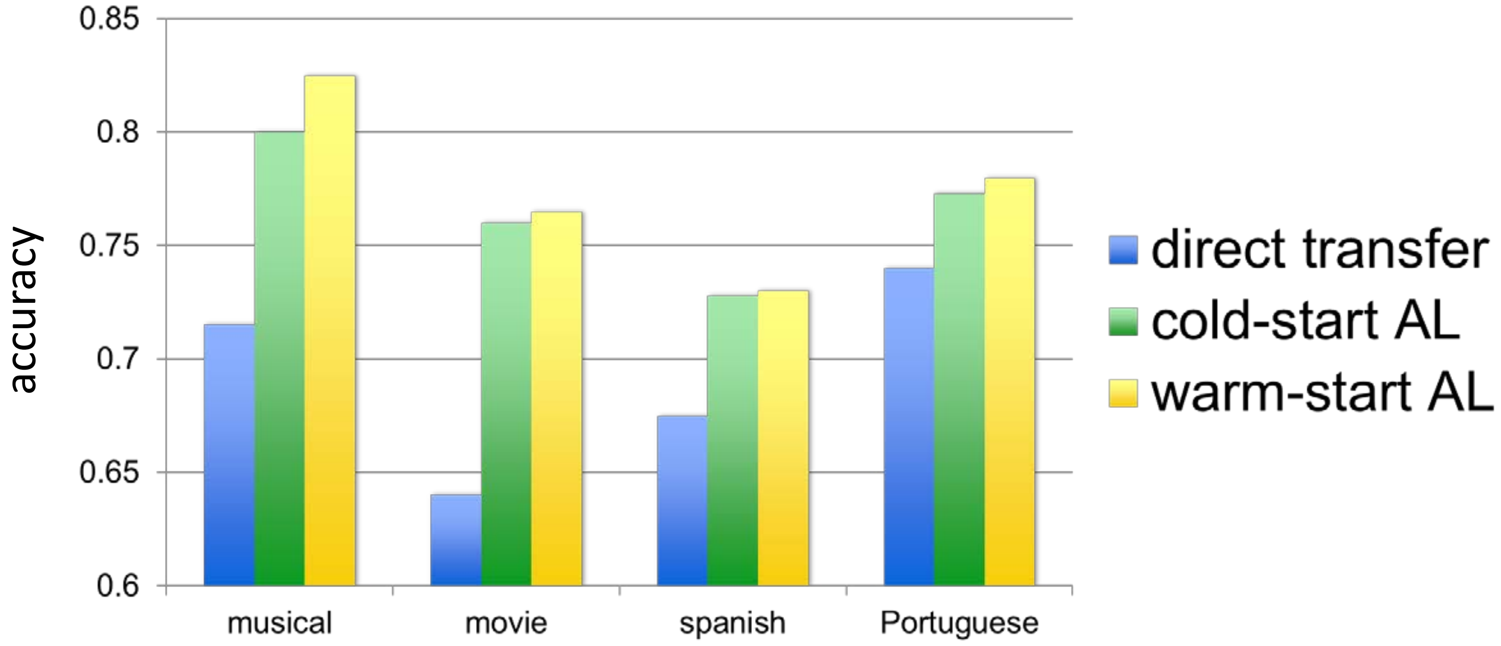
Experiments (Baseline methods)

- Random sampling
- Uncertainty-based sampling
- Diversity-based sampling
- PAL (Fang et al., 2017) : A deep reinforcement learning based approach, they designed a Q-network for stream-based AL

Experiments (Task 1: text classification)



Experiments (Task 1: text classification)



- Direct transfer: Initialize the classifier on the source data, without AL
- Cold-start: Start training the classifier from random initialization, continue training with AL agent
- Warm-start: Start training the classifier from the pre-trained model on the source data, continue training with AL agent

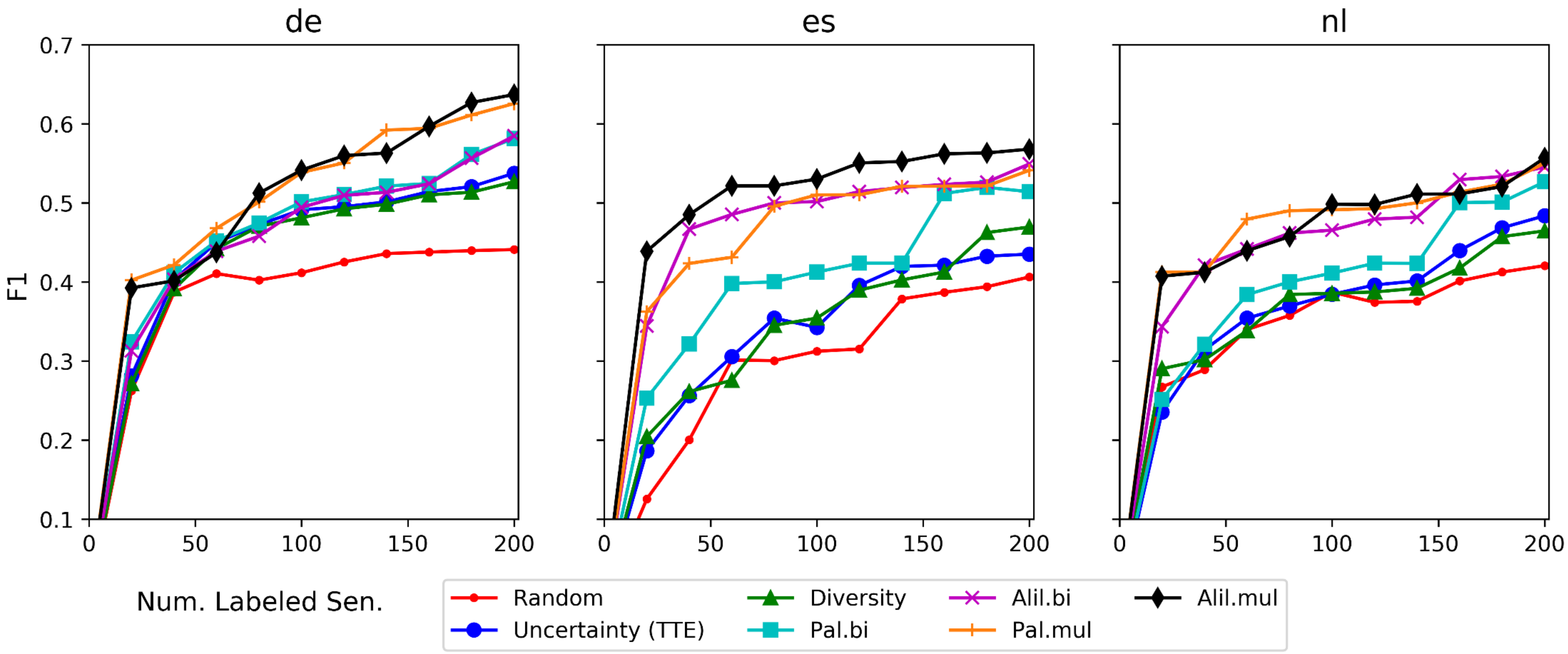
Experiments (Task 2: Named Entity Recognition)

- Data sets: CoNLL 2002/2003

Bilingual		Multilingual	
tgt	src	tgt	src
de	en	de	en,nl,es
nl	en	nl	en,de,es
es	en	es	en,de,nl

Table 2: Experimental settings for cross lingual NER, in which source language (src) is used for policy training.

Experiments (Task 2: Named Entity Recognition)



Analysis: Insight on the selected data

$$acc = \frac{total\ \#\ of\ overlapped\ examples}{budget}$$

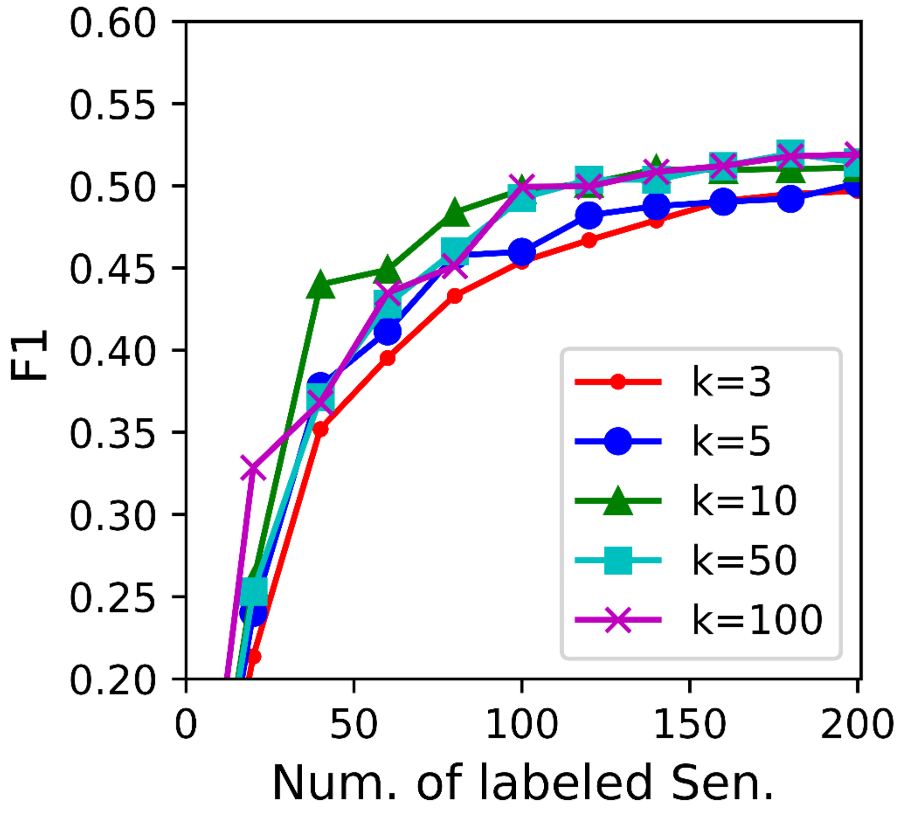
$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

	movie sentiment	gender pt	NER es
acc Unc.	0.06	0.58	0.51
MRR Unc.	0.083	0.674	0.551
acc Div.	0.05	0.52	0.45
MRR Div.	0.057	0.593	0.530
acc PAL	0.15	0.56	0.52

We use MRR(Mean reciprocal rank) and acc to show the agreement of queried data points returned by our AL agent and other strategies.

Analysis: Sensitivity to K (size of unlabeled subset)

K: size of subset from the original unlabelled set

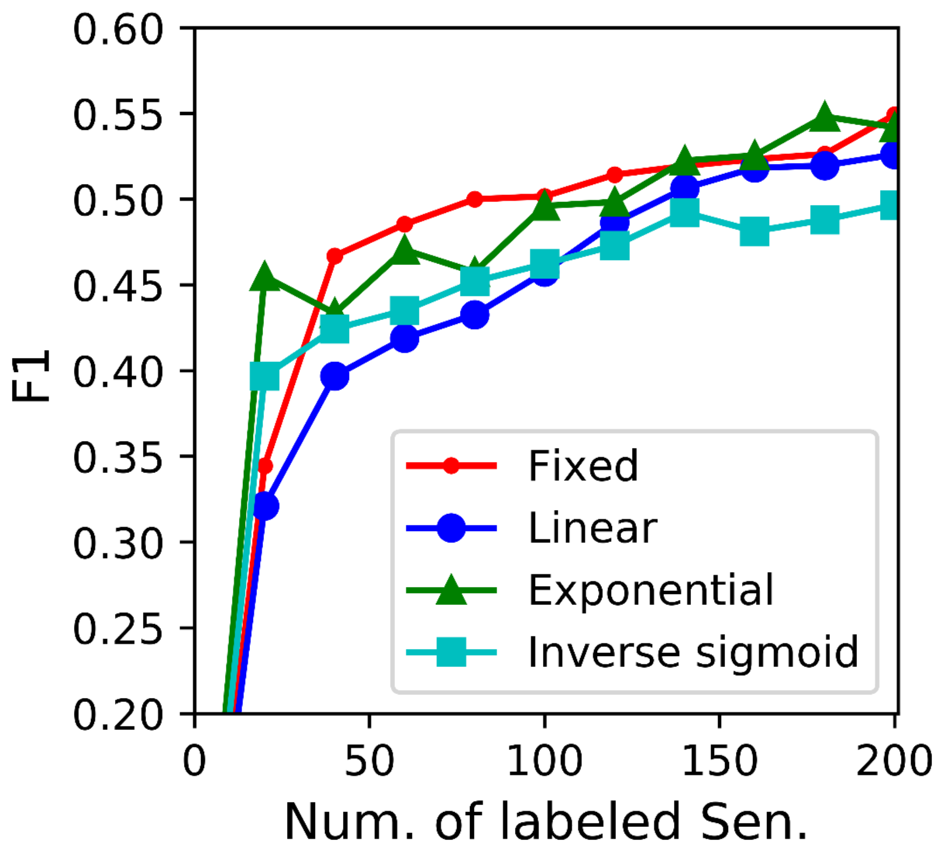


Analysis :β (schedule parameter for the policy)

$$\pi_\tau = \beta_\tau \tilde{\pi}^* + (1 - \beta_\tau)\hat{\pi}_\tau$$

Options for β

- Fixed: β=0.5
- Linear: $\beta_\tau = \max(0.5, 1 - 0.01\tau)$
- Exponential: $\beta_\tau = 0.9^\tau$
- Inverse sigmoid: $\beta_\tau = \frac{5}{5+\exp(\tau/5)}$



Related work

- Meta learning eg learning to learn without gradient descent by gradient descent (Chen et al 2016)

- Stream-based AL as MDP; learning the policy with reinforcement learning (Fang et al, 2017) suffers from the credit assignment problem (Bechman et al 2017)

- Imitation Learning: Lerning from expert demonstrations eg (Schaal 2009, Abbeel & Ng 2004, Silver et al 2008)

Conclusion

- Use heuristics or learn an agent for the AL query strategy.
- Agent-based AL as a Markov Decision Process.
- Formulate learning AL strategies/policies as an imitation learning problem.
- Our imitation learning approach performs better than previous heuristic-based and RL-based methods.

Thanks