Active Imitation Learning with Noisy Guidance

Kianté Brantley,¹ Amr Sharaf,¹ Hal Daumé III ¹,²

¹ University of Maryland, ² Microsoft Research
Structured Prediction Problems

for example, Named Entity Recognition:

<table>
<thead>
<tr>
<th>Word</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>After</td>
<td>O</td>
</tr>
<tr>
<td>completing</td>
<td>O</td>
</tr>
<tr>
<td>his</td>
<td>O</td>
</tr>
<tr>
<td>Ph.D.</td>
<td>O</td>
</tr>
<tr>
<td>,</td>
<td>O</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>
Problem:

☐ Can we design an algorithm to reduce expert annotation cost for structure prediction problems?
Imitation Learning

Expert Demonstrator: (Annotator)

Named Entity Recognition

Input: After completing his Ph.D., Ellis worked at Bell Labs from 1969 to 1972 on probability theory.

Prediction: 0

- states combine input with previous prediction
- actions o, per, org, misc, loc

training set: $D = \{(state, actions)\}$ from expert $\pi^*$
goal: learn agent $\pi_\theta(s) \rightarrow a$
Imitation Learning using **DAgger**

Initialize Dataset \( D \)

Initialize \( \hat{\pi}_1 \)

for \( i = 1 \) to \( N \) do

\[ \pi_i = \beta_i \pi^* + (1 - \beta_i) \hat{\pi}_i \]

Sample T-step trajectory from \( \pi_i \)

Get dataset \( D_i = \{ (s, \pi^*(s)) \} \)

Aggregate dataset \( D \leftarrow D \cup D_i \)

Train classifier \( \hat{\pi}_{i+1} \) on \( D \)

---

**Pro:**

- The policy is able to learn from its own state distribution.

---


---

**Named Entity Recognition**

...
Initialize Dataset $D$

Initialize $\hat{\pi}_1$

for $i = 1$ to $N$

$\pi_i = \beta_i \pi^* + (1 - \beta_i) \hat{\pi}_i$

Sample $T$-step trajectory from $\pi_i$

Get dataset $D_i = \{(s, \pi^*(s))\}$

Aggregate dataset $D \leftarrow D \cup D_i$

Train classifier $\hat{\pi}_{i+1}$ on $D$

Con:

□ For every state that we visited we queried an expert for the optimal action.
Active Learning

Key Idea: The learner queries the expert for labels — only when it is uncertain

Formally

for each trial $t = 1, 2, ...$
observe instance $x_t \in \mathbb{R}$
set $\hat{p}_t = \pi_\theta(y^1_t | x_t) - \pi_\theta(y^2_t | x_t)$ (Margin between the most likely and the second most likely labels)
predict with $\hat{y}_t = \text{argmax}(\pi_\theta)$
draw a Bernoulli variable $Z_t$ of parameter $\frac{b}{b + |\hat{p}_t|}$ (Confidence parameter $b$)
if $Z_t = 1$
query label $y_t$ and perform update


Leveraging Active Learning

Key Idea: The learner queries the expert for labels — only when it is uncertain

Formally

for each trial \( t = 1,2,... \)

observe instance \( x_t \in \mathbb{R} \)

set \( \hat{p}_t = \pi_\theta(y^1_t \mid x_t) - \pi_\theta(y^2_t \mid x_t) \) (Margin)

predict with \( \hat{y}_t = \text{argmax}(\pi_\theta) \)

draw a Bernoulli variable \( Z_t \) of parameter \( b + |\hat{p}_t| \) (Confidence parameter \( b \))

if \( Z_t = 1 \)

query label \( y_t \) and perform update

Confidence parameter: \( b \)

- big - increases the probability of requesting a label
- small - decreases the probability of requesting a label


Active Learning with DAgger

Initialize Dataset $D$
Initialize $\hat{\pi}_i$

for $i = 1$ to $N$ do

$\pi_i = \beta_i \pi^* + (1 - \beta_i) \hat{\pi}_i$

Sample T-step trajectory

for $t = 1$ to $T$

set $\hat{p}_t = \pi_\theta(y_t^1 | s_t)$

draw Bernoulli variable $Z_t$ or parameter $b + |\hat{p}_t|$

if $Z_t = 1$

Get dataset $D_t = \{(s_t, \pi^*(s_t))\}$

Aggregate dataset $D \leftarrow D \cup D_t$

Train classifier $\hat{\pi}_{i+1}$ on $D$

Question:

☐ Can reduce expert queries even further?
Our Approach: LeaQI
(Learning to Query for Imitation)

Key Ideas:

- We assume access to a noisy heuristic function

- Use a disagreement classifier to decide if we should query the expert or the heuristic function

- Train the disagreement classifier using the Apple Tasting framework
Apple Tasting Framework
One-Side Feedback Problem

Learner encounters apples one by one

Goal is to avoid tasting to many bad apples and avoid throwing away to many good apples (reduce false negative rates)

Problem is the learner can only identify the good and bad apples by tasting them

Learner only gets feedback for apples that it tastes

Learner does not feedback for apples that it throws away
One-Sided Feedback Learning

Named Entity Recognition

Input: After completing his Ph.D., Ellis worked at Bell Labs from 1969 to 1972 on probability theory.

Heuristic Function

\[ \pi \]

Heuristic Function

\[ \pi^* \]

LeaQI One-Side Feedback Problem

- Learn difference classifier to predict when a Heuristic and Expert disagree
- Difference classifier only gets feedback when it predicts disagree and we query the expert
- Difference classifier does not get feedback when it predicts agree and we query the heuristic function

Heuristic Function

- Noisy, bias and cheap
Train classifier $\hat{\pi}_{i+1}$ on $D$

Train difference classifier $h_{i+1}$ on $S$

draw Bernoulli variable $Z_i$ of parameter $\frac{b}{b + |\hat{p}_i|}$

if $Z_i = 1$

$\hat{d}_i = h_i(s)$  Set difference classifier

if AppleTaste($s$, $\pi^h(s)$, $\hat{d}_i$)

Aggregate dataset $D \leftarrow D \cup \{(s, \pi^h(s))\}$

else

Aggregate dataset $D \leftarrow D \cup \{(s, \pi^*(s))\}$

Aggregate dataset $S \leftarrow S \cup \{(s, \pi^h(s), \hat{d}, d)\}$

Name Entity Recognition

Input: After completing his Ph.D., Ellis worked

Gazetteer: $\pi$ $\pi^*$

Difference Classifier: Y N Y N Y Y Y O
## Experiment Details

<table>
<thead>
<tr>
<th>Language</th>
<th>NER</th>
<th>Keyphrase</th>
<th>POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modern Greek</td>
<td>English</td>
<td>English</td>
<td>Modern Greek</td>
</tr>
<tr>
<td>English</td>
<td>CoNLL’03</td>
<td>SemEval 2017</td>
<td>Universal Dependencies</td>
</tr>
<tr>
<td>Modern Greek</td>
<td>67% acc</td>
<td>Gazeteer</td>
<td>Unsupervised model</td>
</tr>
<tr>
<td>English</td>
<td>P88%, R27%</td>
<td>P20%, R44%</td>
<td>67% acc</td>
</tr>
<tr>
<td>Modern Greek</td>
<td>67% acc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>Gazeteer</td>
<td>Unsupervised model</td>
<td>Dictionary Wiktionary</td>
</tr>
<tr>
<td>English</td>
<td>P88%, R27%</td>
<td>P20%, R44%</td>
<td>67% acc</td>
</tr>
<tr>
<td>Modern Greek</td>
<td>67% acc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Q1  
Active vs Passive

Q2  
Heuristic as features vs Policy

Q3  
Difference Classifier Efficacy

Q4  
Apple Tasting Efficacy

Q5  
Robustness to Poor a Heuristic

Experiment Results

- Named Entity Recognition
- Keyphrase Extraction
- Part of Speech Tagging

Graphs showing performance metrics over different datasets and model configurations.
We showed that the Apple Tasting framework has practical benefits.

We showed a relationship between using a heuristic function and One-side feedback learning.

We introduced a new algorithm and evaluated it on 3 tasks.
Thank you!