Learning How to Actively Learn: A Deep Imitation Learning Approach

Ming Liu **Wray Buntine** **Gholamreza Haffari**
Faculty of Information Technology, Monash University
{ming.m.liu, wray.buntine, gholamreza.haffari} @ monash.edu

Abstract

Heuristic-based active learning (AL) methods are limited when the data distribution of the underlying learning problems vary. We introduce a method that learns an AL *policy* using *imitation learning* (IL). Our IL-based approach makes use of an efficient and effective *algorithmic expert*, which provides the policy learner with good actions in the encountered AL situations. The AL strategy is then learned with a feedforward network, mapping situations to most informative query datapoints. We evaluate our method on two different tasks: text classification and named entity recognition. Experimental results show that our IL-based AL strategy is more effective than strong previous methods using heuristics and reinforcement learning.

1 Introduction

For many real-world NLP tasks, labeled data is rare while unlabelled data is abundant. Active learning (AL) seeks to learn an accurate model with minimum amount of annotation cost. It is inspired by the observation that a model can get better performance if it is allowed to choose the data points on which it is trained. For example, the learner can identify the areas of the space where it does not have enough knowledge, and query those data points which bridge its knowledge gap.

Traditionally, AL is performed using engineered heuristics in order to estimate the usefulness of unlabeled data points as queries to an annotator. Recent work (Fang et al., 2017; Bachman et al., 2017; Woodward and Finn, 2017) have focused on learning the AL querying strategy, as engineered heuristics are not flexible to exploit characteristics inherent to a given problem. The basic idea is to cast AL as a decision process, where the most informative unlabeled data point needs to be selected based on the history of previous queries. However, previous works train for the AL policy by a reinforcement learning (RL) formulation, where the rewards are provided at the end of sequences of queries. This makes learning the AL policy difficult, as the policy learner needs to deal with the credit assignment problem. Intuitively, the learner needs to observe many pairs of query sequences and the resulting end-rewards to be able to associate single queries with their utility scores.

In this work, we formulate learning AL strategies as an imitation learning problem. In particular, we consider the popular pool-based AL scenario, where an AL agent is presented with a pool of unlabelled data. Inspired by the Dataset Aggregation (DAGGER) algorithm (Ross et al., 2011), we develop an effective AL policy learning method by designing an efficient and effective algorithmic expert, which provides the AL agent with good decisions in the encountered states. We then use a deep feedforward network to learn the AL policy to associate states to actions. Unlike the RL approach, our method can get observations and actions directly from the expert's trajectory. Therefore, our trained policy can make better rankings of unlabelled datapoints in the pool, leading to more effective AL strategies.

We evaluate our method on text classification and named entity recognition. The results show our method performs better than strong AL methods using heuristics and reinforcement learning, in that it boosts the performance of the underlying model with fewer labelling queries. An open source implementation of our model is available at: https://github.com/Grayming/ALIL.

2 Pool-based AL as a Decision Process

We consider the popular pool-based AL setting where we are given a small set of initial labeled data and a large pool of unlabelled data, and a budget for getting the annotation of some unlabelled data by querying an oracle, e.g. a human annotator. The goal is to intelligently pick those unlabelled data for which if the annotations were available, the performance of the underlying re-trained model would be improved the most.

The main challenge in AL is how to identify and select the most beneficial unlabelled data points. Various *heuristics* have been proposed to guide the unlabelled data selection (Settles, 2010). However, there is no one AL heuristic which performs best for all problems. The goal of this paper is to provide an approach to *learn* an AL strategy which is best suited for the problem at hand, instead of resorting to ad-hoc heuristics.

The AL strategy can be learned by attempting to actively learn on tasks sampled from a distribution over the tasks (Bachman et al., 2017). The idea is to *simulate* the AL scenario on *instances* of the problem created using available labeled data, where the label of some part of the data is kept hidden. This allows to have an *automatic* oracle to reveal the labels of the queried data, resulting in an efficient way to quickly evaluate a hypothesised AL strategy. Once the AL strategy is learned on simulations, it is then applied to real AL scenarios. The more related are the tasks in the real scenario to those used to train the AL strategy, the more effective the AL strategy would be.

We are interested to train a model $m_{\boldsymbol{\phi}}$ which maps an input $\boldsymbol{x} \in \mathcal{X}$ to its label $\boldsymbol{y} \in \mathcal{Y}_{\boldsymbol{x}}$, where $\mathcal{Y}_{\boldsymbol{x}}$ is the set of labels for the input $\boldsymbol{x}$ and $\boldsymbol{\phi}$ is the parameter vector of the underling model. For example, in the named entity recognition (NER) task, the input is a sentence and the output is its label sequence, e.g. in the IBO format. Let $D = \{(\boldsymbol{x}, \boldsymbol{y})\}$ be a support set of labeled data, which is randomly partitioned into labeled D^{lab}, unlabelled D^{unl}, and evaluation D^{evl} datasets. Repeated random partitioning creates multiple instances of the AL problem. At each time step t of an AL problem, the algorithm interacts with the oracle and queries the label of a datapoint $\boldsymbol{x}_t \in D_t^{unl}$. As the result of this *action*, the followings happen:

- The automatic oracle reveals the label $\boldsymbol{y}_t$;
- The labeled and unlabelled datasets are updated to include and exclude the recently queried data point, respectively;
- The underlying model is re-trained based on the enlarged labeled data to update $\boldsymbol{\phi}$; and
- The AL algorithm receives a reward $-loss(m_{\boldsymbol{\phi}}, D^{evl})$, which is the negative loss of the current trained model on the evaluation set, defined as

$$loss(m_{\boldsymbol{\phi}}, D^{evl}) := \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D^{evl}} loss(m_{\boldsymbol{\phi}}(\boldsymbol{x}), \boldsymbol{y})$$

  where $loss(\boldsymbol{y}', \boldsymbol{y})$ is the loss incurred due to predicting $\boldsymbol{y}'$ instead of the ground truth $\boldsymbol{y}$.

More formally, a pool-based AL problem is a Markov decision process (MDP), denoted by $(S, A, Pr(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t), R)$ where S is the state space, A is the set of actions, $Pr(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t)$ is the transition function, and R is the reward function. The state $\boldsymbol{s}_t \in S$ at time t consists of the labeled D_t^{lab} and unlabelled D_t^{unl} datasets paired with the parameters of the currently trained model ϕ_t. An action $a_t \in A$ corresponds to the selection of a query datapoint, and the reward function $R(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}) := -loss(m_{\boldsymbol{\phi}_t}, D^{evl})$.

We aim to find the optimal AL *policy* prescribing which datapoint needs to be queried in a given state to get the most benefit. The optimal policy is found by maximising the following objective over the parameterised policies:

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

where $\pi_{\boldsymbol{\theta}}$ is the *policy network* parameterised by $\boldsymbol{\theta}$, $\mathcal{D}$ is a distribution over possible AL problem instances, and $\mathcal{B}$ is the maximum number of queries made in an AL run, a.k.a. an *episode*. Following (Bachman et al., 2017), we maximise the sum of the rewards after each time step to encourage the *anytime* behaviour, i.e. the model should perform well after each label query.

3 Deep Imitation Learning to Train the AL Policy

The question remains as how can we train the policy network to maximise the training objective in eqn 1. Typical learning approaches resort to deep reinforcement learning (RL) and provide training signal at the end of each episode to learn the optimal policy (Fang et al., 2017; Bachman

et al., 2017) e.g., using policy gradient methods. These approaches, however, need a large number of training episodes to learn a reasonable policy as they need to deal with the credit assignment problem, i.e. discovery of the utility of individual actions in the sequence based on the achieved reward at the end of the episode. This exacerbates the difficulty of finding a good AL policy.

We formulate learning for the AL policy as an imitation learning problem. At each state, we provide the AL agent with *a correct* action which is computed by an *algorithmic expert*. The AL agent uses the sequence of states observed in an episode paired with the expert's sequence of actions to update its policy. This directly addresses the credit assignment problem, and reduces the complexity of the problem compared to the RL approaches. In what follows, we describe the ingredients of our deep imitation learning (IL) approach, which is summarised in Algorithm 1.

Algorithmic Expert. At a given AL state $\boldsymbol{s}_t$, our algorithmic expert computes an action by evaluating the current pool of unlabeled data. More concretely, for each $\boldsymbol{x}' \in D^{pool}_{rnd}$ and its correct label $\boldsymbol{y}'$, the underlying model $m_{\boldsymbol{\phi}_t}$ is re-trained to get $m^{\boldsymbol{x}',\boldsymbol{y}'}_{\boldsymbol{\phi}_t}$, where $D^{pool}_{rnd} \subset D^{unl}_t$ is a small subset of the current large pool of unlabeled data. The expert action is then computed as:

$$\arg \min_{\boldsymbol{x}' \in D^{pool}_{rnd}} loss(m^{\boldsymbol{x}',\boldsymbol{y}'}_{\boldsymbol{\phi}_t}(\boldsymbol{x}), D^{evl}). \quad (2)$$

In other words, our algorithmic expert tries a subset of actions to *roll-out* one step from the current state, in order to *efficiently* compute a reasonable action. Searching for the *optimal* action would be $\mathcal{O}(|D^{unl}|^{\mathcal{B}})$, which is computationally challenging due to (i) the large action set, and (ii) the exponential dependence on the length of the roll out. We will see in the experiments that our method efficiently learns effective AL policies.

Policy Network. Our policy network is a feed-forward network with two fully-connected hidden layers. It receives the current AL state, and provides a *preference* score for a given unlabeled data point, allowing to select the most beneficial one corresponding to the highest score. The input to our policy network consists of three parts: (i) a fixed dimensional representation of the content and the predicted label of the unlabeled data point under consideration, (ii) a fixed-dimensional representation of the content and the labels of the labeled dataset, and (iii) a fixed-dimensional representation of the content of the unlabeled dataset.

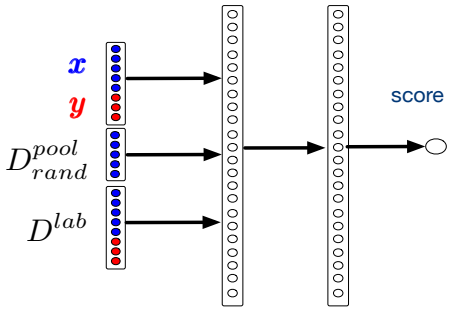

Figure 1: The policy network and its inputs.

Imitation Learning Algorithm. A typical approach to imitation learning (IL) is to train the policy network so that it mimics the expert's behaviour given training data of the encountered states (input) and actions (output) performed by the expert. The policy network's prediction affects future inputs during the execution of the policy. This violates the crucial independent and identically distributed (iid) assumption, inherent to most statistical supervised learning approaches for learning a mapping from states to actions.

We make use of Dataset Aggregation (DAGGER) (Ross et al., 2011), an iterative algorithm for IL which addresses the *non*-iid nature of the encountered states during the AL process (see Algorithm 1). In round τ of DAGGER, the learned policy network $\hat{\pi}_\tau$ is applied to the AL problem to collect a sequence of states which are paired with the expert actions. The collected pair of states and actions are aggregated to the dataset of such pairs M, collected from the previous iterations of the algorithm. The policy network is then re-trained on the aggregated set, resulting in $\hat{\pi}_{\tau+1}$ for the next iteration of the algorithm. The intuition is to build up the set of states that the algorithm is likely to encounter during its execution, in order to increase the generalization of the policy network. To better leverage the training signal from the algorithmic expert, we allow the algorithm to collect state-action pairs according to a modified policy which is a mixture of $\hat{\pi}_\tau$ and the expert policy $\tilde{\pi}^*_\tau$, i.e.

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

where $\beta_\tau \in [0, 1]$ is a mixing coefficient. This amounts to tossing a coin with parameter β_τ in

each iteration of the algorithm to decide one of these two policies for data collection.

Re-training the Policy Network. To train our policy network, we turn the preference scores to probabilities, and optimise the parameters such that the probability of the action prescribed by the expert is maximized. More specifically, let $\mathcal{M} := \{(\boldsymbol{s}_i, \boldsymbol{a}_i)\}_{i=1}^{I}$ be the collected states paired with their expert's prescribed actions. Let D_i^{pool} be the set of unlabelled datapoints in the pool within the state, and $\boldsymbol{a}_i$ denote the datapoint selected by the expert in the set. Our training objective is $\sum_{i=1}^{I} \log Pr(\boldsymbol{a}_i | D_i^{pool})$ where

$$Pr(\boldsymbol{a}_i | D_i^{pool}) := \frac{\exp \hat{\pi}(\boldsymbol{a}_i; \boldsymbol{s}_i)}{\sum_{\boldsymbol{x} \in D_i^{pool}} \exp \hat{\pi}(\boldsymbol{x}; \boldsymbol{s}_i)}.$$

The above can be interpreted as the probability of $\boldsymbol{a}_i$ being the best action among all possible actions in the state. Following (Mnih et al., 2015), we randomly sample multiple[1] mini-batches from the *replay memory* $\mathcal{M}$, in addition to the current round's stat-action pair, in order to retrain the policy network. For each mini-batch, we make one SGD step to update the policy, where the gradients of the network parameters are calculated using the backpropagation algorithm.

Transferring the Policy. We now apply the policy learned on the source task to AL in the target task. We expect the learned policy to be effective for target tasks which are related to the source task in terms of the data distribution and characteristics. Algorithm 2 illustrates the policy transfer. The pool-based AL scenario in Algorithm 2 is cold-start; however, extending to incorporate initially available labeled data is straightforward.

Algorithm 1 Learn active learning policy via imitation learning

Input: large labeled data D, max episodes T, budget $\mathcal{B}$, sample size K, the coin parameter β
Output: The learned policy

$M \leftarrow \emptyset$ ▷ the aggregated dataset
initialise $\hat{\pi}_1$ with a random policy
for τ=1, ..., T **do**
  $D^{lab}, D^{unl}, D^{evl} \leftarrow$ dataPartition(D)
  $\boldsymbol{\phi}_1 \leftarrow$ trainModel(D^{lab})
  $c \leftarrow$ coinToss(β)
  for $t \in 1, \ldots, \mathcal{B}$ **do**
    $D_{rnd}^{pool} \leftarrow$ sampleUniform(D^{unl}, K)
    $\boldsymbol{s}_t \leftarrow (D^{lab}, D_{rnd}^{pool}, \boldsymbol{\phi}_t)$
    $\boldsymbol{a}_t \leftarrow \arg\min_{\boldsymbol{x}' \in D_{rnd}^{pool}} loss(m_{\boldsymbol{\phi}_t}^{\boldsymbol{x}'}, D^{evl})$
    if c is head **then** ▷ the expert
      $\boldsymbol{x}_t \leftarrow \boldsymbol{a}_t$
    else ▷ the policy
      $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}' \in D_{rnd}^{pool}} \hat{\pi}_\tau(\boldsymbol{x}'; \boldsymbol{s}_t)$
    end if
    $D^{lab} \leftarrow D^{lab} + \{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$
    $D^{unl} \leftarrow D^{unl} - \{\boldsymbol{x}_t\}$
    $M \leftarrow M + \{(\boldsymbol{s}_t, \boldsymbol{a}_t)\}$
    $\boldsymbol{\phi}_{t+1} \leftarrow$ retrainModel($\boldsymbol{\phi}_t, D^{lab}$)
  end for
  $\hat{\pi}_{\tau+1} \leftarrow$ retrainPolicy($\hat{\pi}_\tau, M$)
end for
return $\hat{\pi}_{T+1}$

Algorithm 2 Active learning by policy transfer

Input: unlabeled pool D^{unl}, budget $\mathcal{B}$, policy $\hat{\pi}$
Output: labeled dataset and trained model

$D^{lab} \leftarrow \emptyset$
initialise $\boldsymbol{\phi}$ randomly
for $t \in 1, \ldots, \mathcal{B}$ **do**
  $\boldsymbol{s}_t \leftarrow (D^{lab}, D^{unl}, \boldsymbol{\phi})$
  $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}' \in D^{unl}} \hat{\pi}(\boldsymbol{x}'; \boldsymbol{s}_t)$
  $\boldsymbol{y}_t \leftarrow$ askAnnotation($\boldsymbol{x}_t$)
  $D^{lab} \leftarrow D^{lab} + \{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$
  $D^{unl} \leftarrow D^{unl} - \{\boldsymbol{x}_t\}$
  $\phi \leftarrow$ retrainModel($\boldsymbol{\phi}, D^{lab}$)
end for
return D^{lab} and $\boldsymbol{\phi}$

4 Experiments

We conduct experiments on text classification and named entity recognition (NER). The AL scenarios include cross-domain sentiment classification, cross-lingual authorship profiling, and cross-lingual named entity recognition (NER), whereby an AL policy trained on a source domain/language is transferred to the target domain/language.

We compare our proposed AL method using imitation learning (ALIL) with the followings:

- *Random sampling*: The query datapoint is chosen randomly.
- *Diversity sampling*: The query datapoint is $\arg\min_{\boldsymbol{x}} \sum_{\boldsymbol{x}' \in D^{lab}} \text{Jaccard}(\boldsymbol{x}, \boldsymbol{x}')$, where the Jaccard coefficient between the unigram features of the two given texts is used as the similarity measure.
- *Uncertainty-based sampling*: For text classification, we use the datapoint with the highest predictive entropy, $\arg\max_{\boldsymbol{x}} -\sum_y p(y|\boldsymbol{x}, D^{lab}) \log p(y|\boldsymbol{x}, D^{lab})$ where $p(\boldsymbol{y}|\boldsymbol{x}, D^{lab})$ comes from the underlying model. We further use a state-of-the-art extension of this method, called *uncertainty with rationals* (Sharma et al., 2015), which not only considers uncertainty but also looks whether

[1] In our experiments, we use 10 mini-bathes, each of which of size 100.

src	tgt	doc. (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

Table 1: The data sets used in sentiment classification (top part) and gender profiling (bottom part).

the unlabelled document contains sentiment words or phrases that were returned as rationales for any of the existing labeled documents. For NER, we use the Total Token Entropy (TTE) as the uncertainty sampling method, $\arg\max_{\boldsymbol{x}} -\sum_{i=1}^{|\boldsymbol{x}|}\sum_{y_i} p(y_i|\boldsymbol{x}, D^{lab}) \log p(y_i|\boldsymbol{x}, D^{lab})$ which has been shown to be the best heuristic for this task among 17 different heuristics (Settles and Craven, 2008).

- *PAL*: A reinforcement learning based approach (Fang et al., 2017), which makes use a deep Q-network to make the selection decision for stream-based active learning.

4.1 Text Classification

Datasets and Setup. The first task is sentiment classification, in which product reviews express either positive or negative sentiment. The data comes from the Amazon product reviews (McAuley and Yang, 2016); see Table 1 for data statistics.

The second task is Authorship Profiling, in which we aim to predict the gender of the text author. The data comes from the gender profiling task in PAN 2017 (Rangel et al., 2017), which consists of a large Twitter corpus in multiple languages: English (en), Spanish (es) and Portuguese (pt). For each language, all tweets collected from a user constitute one document; Table 1 shows data statistics. The multilingual embeddings for this task come from off-the-shelf CCA-trained embeddings (Ammar et al., 2016) for twelve languages, including English, Spanish and Portuguese. We fix these word embeddings during training of both the policy and the underlying classification model.

For training, 10% of the source data is used as the evaluation set for computing the best action in imitation learning. We run $T = 100$ episodes with the budget $\mathcal{B} = 100$ documents in each episode, set the sample size $K = 5$, and fix the mixing coefficient $\beta_\tau = 0.5$. For testing, we take 90% of the target data as the unlabeled pool, and the remaining 10% as the test set. We show the test accuracy w.r.t. the number of labelled documents selected in the AL process.

As the underlying model $m_{\boldsymbol{\phi}}$, we use a fast and efficient text classifier based on convolutional neural networks. More specifically, we apply 50 convolutional filters with ReLU activation on the embedding of all words in a document $\boldsymbol{x}$, where the width of the filters is 3. The filter outputs are averaged to produce a 50-dimensional document representation $\boldsymbol{h}(\boldsymbol{x})$, which is then fed into a softmax to predict the class.

Representing state-action. The input to the policy network, i.e. the feature vector representing a state-action pair, includes: the candidate document represented by the convolutional net $\boldsymbol{h}(\boldsymbol{x})$, the distribution over the document's class labels $m_{\boldsymbol{\phi}}(\boldsymbol{x})$, the sum of all document vector representations in the labeled set $\sum_{\boldsymbol{x}' \in D^{lab}} \boldsymbol{h}(\boldsymbol{x}')$, the sum of all document vectors in the random pool of unlabelled data $\sum_{\boldsymbol{x}' \in D^{pool}_{rnd}} \boldsymbol{h}(\boldsymbol{x}')$, and the empirical distribution of class labels in the labeled dataset.

Results. Fig 2 shows the results on product sentiment prediction and authorship profiling, in cross-domain and cross-lingual AL scenarios[2]. Our ALIL method consistently outperforms both heuristic-based and RL-based (PAL) (Fang et al., 2017) approaches across all tasks. ALIL tends to convergence faster than other methods, which indicates its policy can quickly select the most informative datapoints. Interestingly, the uncertainty and diversity sampling heuristics perform worse than random sampling on sentiment classification. We speculate this may be due to these two heuristics not being able to capture the polarity information during the data selection process. PAL performs on-par with uncertainty with rationals on musical device, both of which outperform the traditional diversity and uncertainty sampling heuristics. Interestingly, PAL is outperformed by random sampling on movie reviews, and by the traditional uncertainty sampling heuristic on authorship profiling tasks. We attribute this to ineffectiveness of the RL-based approach for learning a reasonable AL query strategy.

We further investigate combining the transfer of the policy network with the transfer of the underlying classifier. That is, we first train a classi-

[2]Uncertainty with rationale cannot be done for authorship profiling as the rationales come from a sentiment dictionary.

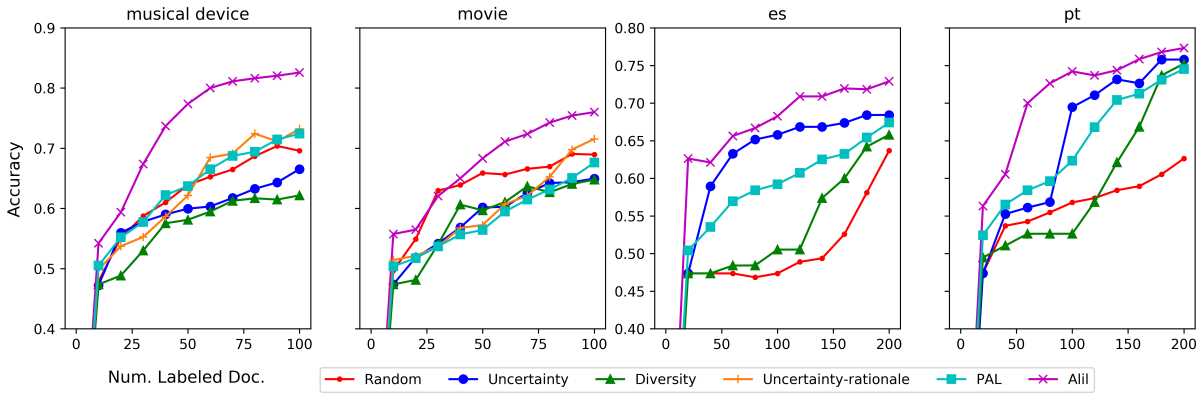

Figure 2: The performance of different active learning methods for cross domain sentiment classification (left two plots) and cross lingual authorship profiling (right two plots).

fier on all of the annotated data from the source domain/language. Then, this classifier is ported to the target domain/language; for cross-language transfer, we make use of multilingual word embeddings. We start the AL process starting from the transferred classifier, referred to as the warm-start AL. We compare the performance of the *directly transferred* classifier with those obtained after the AL process in the warm-start and cold-start scenarios. The results are shown in Table 2. We have run the cold-start and warm-start AL for 25 times, and reported the average accuracy in Table 2. As seen from the results, both the cold and warm start AL settings outperform the direct transfer significantly, and the warm start consistently gets higher accuracy than the cold start. The difference between the results are statistically significant, with a p-value of .001, according to McNemar test[3] (Dietterich, 1998).

	musical	movie	es	pt
direct transfer	0.715	0.640	0.675	0.740
cold-start AL	0.800	0.760	0.728	0.773
warm-start AL	**0.825**	**0.765**	**0.730**	**0.780**

Table 2: Classifiers performance under three different transfer settings.

4.2 Named Entity Recognition

Data and setup We use NER corpora from the CONLL2002/2003 shared tasks, which include annotated text in English (en), German (de), Spanish (es), and Dutch (nl). The original annotation is based on IOB1, which we convert to the IO labelling scheme. Following Fang et al. (2017), we consider two experimental conditions: (i) the bilingual scenario where English is the source (used for policy training) and other languages are the target, and (ii) the multilingual scenario where one of the languages (except English) is the target and the remaining ones are the source used in joint training of the AL policy. The underlying model $m_{\boldsymbol{\phi}}$ is a conditional random field (CRF) treating NER as a sequence labelling task. The prediction is made using the Viterbi algorithm.

In the existing corpus partitions from CoNLL, each language has three subsets: **train**, **testa** and **testb**. During policy training with the source language(s), we combine these three subsets, shuffle, and re-split them into simulated training, unlabelled pool, and evaluation sets in every episode. We run $N = 100$ episodes with the budget $\mathcal{B} = 200$, and set the sample size $k = 5$. When we transfer the policy to the target language, we do one episode and select $\mathcal{B}$ datapoints from **train** (treated as the pool of unlabeled data) and report F1 scores on **testa**.

Representing state-action. The input to the policy network includes the representation of the candidate sentence using the sum of its words' embeddings $\boldsymbol{h}(\boldsymbol{x})$, the representation of the labelling marginals using the label-level convolutional network $\text{cnn}_{\text{lab}}(\mathbb{E}_{m_{\boldsymbol{\phi}}(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}])$ (Fang et al., 2017), the representation of sentences in the labeled data $\sum_{(\boldsymbol{x}',\boldsymbol{y}')\in D^{lab}} \boldsymbol{h}(\boldsymbol{x}')$, the representation of sentences in the random pool of unlabelled data $\sum_{\boldsymbol{x}'\in D^{pool}_{rnd}} \boldsymbol{h}(\boldsymbol{x}')$, the representation of ground-truth labels in the labeled data $\sum_{(\boldsymbol{x}',\boldsymbol{y}')\in D^{lab}} \text{cnn}_{\text{lab}}(\boldsymbol{y}')$ using the empirical distributions, and the confidence of the sequential pre-

[3] As the contingency table needed for the McNemar test, we have used the average counts across the 25 runs.

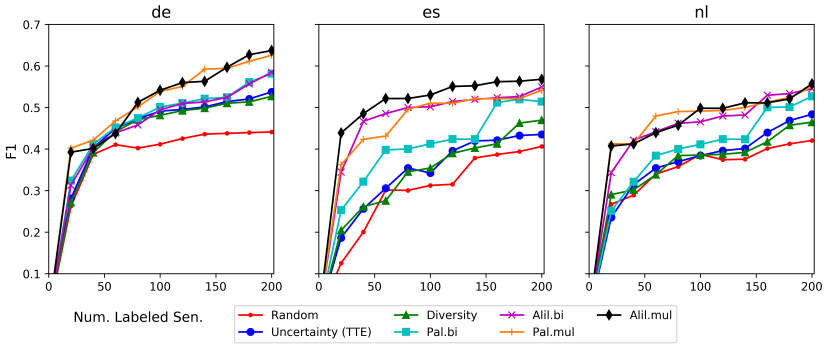

Figure 3: The performance of active learning methods on the bilingual and multilingual settings for three target languages: German (de), Spanish (es) and Dutch (nl).

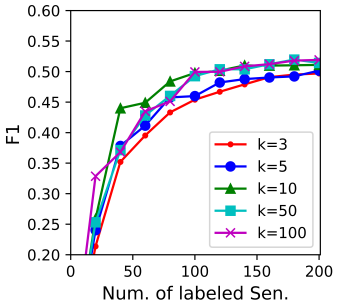

Figure 4: The learning curves of agents with different K on Spanish (es) NER.

diction $\sqrt[|\boldsymbol{x}|]{\max_{\boldsymbol{y}} m_{\boldsymbol{\phi}}(\boldsymbol{y}|\boldsymbol{x})}$, where $|\boldsymbol{x}|$ denotes the length of the sentence $\boldsymbol{x}$. For the word embeddings, we use off-the-shelf CCA trained multilingual embeddings (Ammar et al., 2016) with 40 dimensions; we fix these during policy training.

Results. Fig. 3 shows the results for three target languages. In addition to the strong heuristic-based methods, we compare our imitation learning approach (ALIL) with the reinforcement learning approach (PAL) (Fang et al., 2017), in both bilingual (bi) and multilingual (mul) transfer settings. Across all three languages, ALIL.bi and ALIL.mul outperform the heuristic methods, including Uncertainty Sampling based on TTE. This is expected as the uncertainty sampling largely relies on a high quality underlying model, and diversity sampling ignores the labelling information. In the bilingual case, ALIL.bi outperforms PAL.bi on Spanish (es) and Dutch (nl), and performs similarly on German (de). In the multilingual case, ALIL.mul achieves the best performance on Spanish, and performs competitively with PAL.mul on German and Dutch.

4.3 Analysis

Insight on the selected data. We compare the data selected by ALIL to other methods. This will confirm that ALIL learns policies which are suitable for the problem at hand, without resorting to a fixed engineered heuristics. For this analysis, we report the mean reciprocal rank (MRR) of the data points selected by the ALIL policy under rankings of the unlabelled pool generated by the uncertainty and diversity sampling. Furthermore, we measure the fraction of times the decisions made by the ALIL policy agrees with those which would have been made by the heuristic methods, which is measured by the accuracy (acc). Table 3 report these measures. As we can see, for sentiment classification since uncertainty and diversity sampling perform badly, ALIL has a big disagreement with them on the selected data points. While for gender classification on Portuguese and NER on Spanish, ALIL shows much more agreement with other three heuristics.

	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

Table 3: The first four rows show MRR and accuracy of instances returned by ALIL under the rankings of uncertainty and diversity sampling, the last row give average accuracy of instances under PAL.

Lastly, we compare chosen queries by ALIL to those by PAL, to investigate the extent of the agreement between these two methods. This is simply measure by the fraction of identical query data points among the total number of queries (i.e. accuracy). Since PAL is stream-based and sensitive to the order in which it receives the data points, we report the *average* accuracy taken over multiple runs with random input streams. The expected accuracy numbers are reported in Table 3. As seen, ALIL has higher overlap with PAL than the heuristic-based methods, in terms of the selected queries.

Sensitivity to K. As seen in Algorithm 1, we resort to an approximate algorithmic expert, which selects the best action in a random subset of the

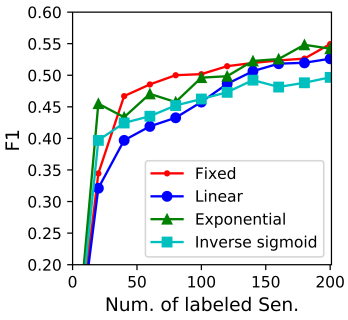

Figure 5: The learning curves of agents with different schedules for β before the trajectory on Spanish (es) NER.

pool of unlabelled data with size K, in order to make the policy training efficient. Note that, in policy training, setting K to one and the size of the unlabelled data pool correspond to stream-based and pool-based AL scenarios, respectively. By changing K to values between these two extremes, we can analyse the effect of the quality of the algorithmic expert on the trained policy; Figure 4 shows the results. A larger candidate set may correspond to a better learned policy, needed to be traded off with the training time growing linearly with K. Interestingly, even small candidate sets lead to strong AL policies as increasing K beyond 10 does not change the performance significantly.

Dynamically changing β. In our algorithm, β plays an important role as it trades off exploration versus exploitation. In the above experiments, we fix it to 0.5; however, we can change its value throughout trajectory collection as a function of τ (see Algorithm 1). We investigate schedules which tend to put more emphasis on exploration and exploitation towards the beginning and end of data collection, respectively. We investigate the following schedules: (i) linear $\beta_\tau = \max(0.5, 1 - 0.01\tau)$, (ii) exponential $\beta_\tau = 0.9^\tau$, and (iii) and inverse sigmoid $\beta_\tau = \frac{5}{5+exp(\tau/5)}$, as a function of iterations. Fig. 5 shows the comparisons of these schedules. The learned policy seems to perform competitively with either a fixed or an exponential schedule. We have also investigated tossing the coin in each step *within* the trajectory roll out, but found that it is more effective to have it *before* the full trajectory roll out (as currently done in Algorithm 1).

5 Related Work

Traditional active learning algorithms rely on various heuristics (Settles, 2010), such as uncertainty sampling (Settles and Craven, 2008; Houlsby et al., 2011), query-by-committee (Gilad-Bachrach et al., 2006), and diversity sampling (Brinker, 2003; Joshi et al., 2009; Yang et al., 2015). Apart from these, different heuristics can be combined, thus creating integrated strategy which consider one or more heuristics at the same time. Combined with transfer learning, pre-existing labeled data from related tasks can help improve the performance of an active learner (Xiao and Guo, 2013; Kale and Liu, 2013; Huang and Chen, 2016; Konyushkova et al., 2017). More recently, deep reinforcement learning is used as the framework for *learning* active learning algorithms, where the active learning cycle is considered as a decision process. (Woodward and Finn, 2017) extended one shot learning to active learning and combined reinforcement learning with a deep recurrent model to make labeling decisions. (Bachman et al., 2017) introduced a policy gradient based method which jointly learns data representation, selection heuristic as well as the model prediction function. (Fang et al., 2017) designed an active learning algorithm based on a deep Q-network, in which the action corresponds to binary annotation decisions applied to a stream of data. The learned policy can then be transferred between languages or domains.

Imitation learning (IL) refers to an agent's acquisition of skills or behaviours by observing an expert's trajectory in a given task. It helps reduce sequential prediction tasks into supervised learning by employing a (near) optimal oracle at training time. Several IL algorithms has been proposed in sequential prediction tasks, including SEARA (Daumé et al., 2009), AggreVaTe (Ross and Bagnell, 2014), DaD (Venkatraman et al., 2015), LOLS(Chang et al., 2015), DeeplyAggreVaTe (Sun et al., 2017). Our work is closely related to Dagger (Ross et al., 2011), which can guarantee to find a good policy by addressing the dependency nature of encountered states in a trajectory.

6 Conclusion

In this paper, we have proposed a new method for learning active learning algorithms using deep imitation learning. We formalize pool-based active

learning as a Markov decision process, in which active learning corresponds to the selection decision of the most informative data points from the pool. Our efficient algorithmic expert provides state-action pairs from which effective active learning policies can be learned. We show that the algorithmic expert allows direct policy learning, while at the same time, the learned policies transfer successfully between domains and languages, demonstrating improvement over previous heuristic and reinforcement learning approaches.

Acknowledgments

We would like to thank the feedback from anonymous reviewers. GH is grateful to Mohammad Ghavamzadeh and Trevor Cohn for interesting discussions. This work was supported by computational resources from the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) at Monash University, and partly by an NVIDIA GPU grant.

References

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

Philip Bachman, Alessandro Sordoni, and Adam Trischler. 2017. Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 301–310, International Convention Centre, Sydney, Australia. PMLR.

Klaus Brinker. 2003. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 59–66.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume, and John Langford. 2015. Learning to search better than your teacher. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2058–2066.

Hal Daumé, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine learning*, 75(3):297–325.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, 10(7):1895–1923.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. *arXiv preprint arXiv:1708.02383*.

Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. 2006. Query by committee made real. In *Advances in neural information processing systems*, pages 443–450.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Sheng-Jun Huang and Songcan Chen. 2016. Transfer learning with active queries from source domain. In *IJCAI*, pages 1592–1598.

Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. 2009. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2372–2379. IEEE.

David Kale and Yan Liu. 2013. Accelerating active learning with transfer learning. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1085–1090. IEEE.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. 2017. Learning active learning from data. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4225–4235. Curran Associates, Inc.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. 2017. Overview of the 5th author profiling task at PAN 2017: Gender and language variety identification in twitter. *Working Notes Papers of the CLEF*.

Stephane Ross and J Andrew Bagnell. 2014. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*.

Stéphane Ross, Geoffrey J Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635.

Burr Settles. 2010. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics.

Manali Sharma, Di Zhuang, and Mustafa Bilgic. 2015. Active learning with rationales for text classification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 441–451.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. 2017. Deeply aggrevated: Differentiable imitation learning for sequential prediction. *arXiv preprint arXiv:1703.01030*.

Arun Venkatraman, Martial Hebert, and J Andrew Bagnell. 2015. Improving multi-step prediction of learned time series models. In *AAAI*, pages 3024–3030.

Mark Woodward and Chelsea Finn. 2017. Active one-shot learning. *arXiv preprint arXiv:1702.06559*.

Min Xiao and Yuhong Guo. 2013. Online active learning for cost sensitive domain adaptation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 1–9.

Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. 2015. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127.