Learning How to Active Learn by Dreaming

Thuy-Trang Vu
Faculty of Information Technology
Monash University, Australia
trang.vuthithuy@monash.edu

Ming Liu
School of Info Technology
Deakin University, Australia
m.liu@deakin.edu

Dinh Phung
Faculty of Information Technology
Monash University, Australia
dinh.phung@monash.edu

Gholamreza Haffari
Faculty of Information Technology
Monash University, Australia
gholamreza.haffari@monash.edu

Abstract

Heuristic-based active learning (AL) methods are limited when the data distribution of the underlying learning problems vary. Recent data-driven AL policy learning methods are also restricted to learn from closely related domains. We introduce a new sample-efficient method that learns the AL policy directly on the target domain of interest by using *wake* and *dream* cycles. Our approach interleaves between querying the annotation of the selected datapoints to update the underlying student learner and improving AL policy using simulation where the current student learner acts as an *imperfect annotator*. We evaluate our method on cross-domain and cross-lingual text classification and named entity recognition tasks. Experimental results show that our dream-based AL policy training strategy is more effective than applying the pretrained policy without further fine-tuning, and better than the existing strong baseline methods that use heuristics or reinforcement learning.

1 Introduction

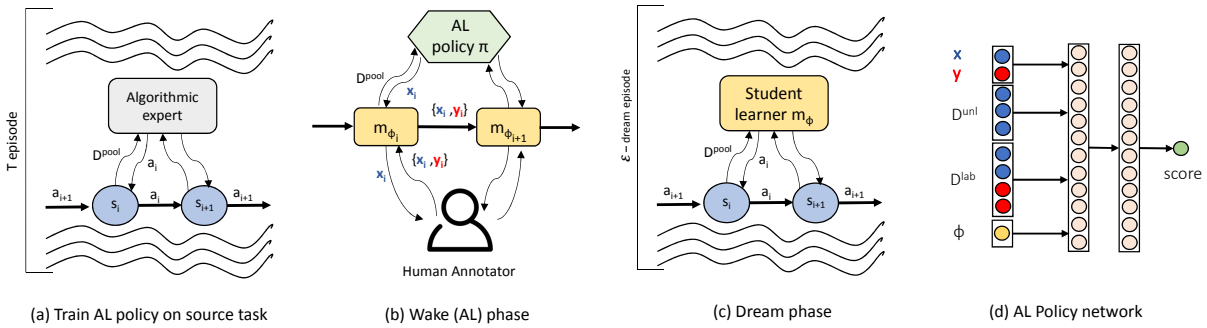
Obtaining adequate annotated data is often expensive and time consuming for many real-world NLP tasks. Active learning (AL) aims to *economically* learn an accurate model by reducing the annotation cost. It is based on the premise that a model can get better performance if it is allowed to prepare its own training data, by choosing the most beneficial data points and querying their annotations from annotators. For example, the learner can identify its knowledge gaps in order to select the most informative query data points.

The core AL problem is how to identify the most beneficial query data points. Traditionally, they are identified using various hand crafted heuristics (Settles, 2012). Recent work has investigated *learning* the AL query strategy from the data (Fang et al., 2017; Bachman et al., 2017; Woodward and Finn, 2017; Contardo et al., 2017; Liu et al., 2018a; Pang et al., 2018), as engineered heuristics are not flexible to exploit characteristics inherent to a given problem. These works are all based on the idea that aims to learn an AL query strategy on a *related* problem for which enough annotated data exist via AL *simulations*, and then *transfers* it to the target AL scenario of interest. The success of this approach, however, highly depends on the relatedness of the source and target AL problems, as the transferred AL strategy is *not* adapted to the characteristics of the target AL problem.

To address this mismatch challenge, we introduce a new approach that learns an AL query strategy *directly* for the target problem of interest. Starting from an initial (pre-trained) AL strategy, our approach interleaves between querying the annotation of the selected data points to update the underlying student model, and improving the AL strategy using simulations. Crucially, in order to improve the query strategy, our AL simulations are based on the target problem, where we make use of the current student learner as an *imperfect annotator*. The AL query strategy is used to train the underlying student learner in the *wake* cycles through interactions with the human annotator, and the student learner is used to train the query strategy in the *dream* cycles via simulations, as illustrated in Figure 1.

Our contribution are as follows: (i) we propose a sample-efficient AL policy learning method to make the best use of the annotation budget to improve both the student learner and the AL policy directly on the target task of interest; (ii) we provide comprehensive experimental results comparing our method to strong heuristic-based and data-driven AL query strategy learning-based methods on cross-lingual and cross-domain text classifica-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4091–4101
Florence, Italy, July 28 - August 2, 2019.



(a) Train AL policy on source task (b) Wake (AL) phase (c) Dream phase (d) AL Policy network

Figure 1: Illustration of our dream-based AL approach. Unlabelled data selection policy is learned in AL simulation on source task (a). At transferring time, we interleave wake phase (b) where the learned policy is applied to train the student learner, and dream phase (c) where the student learner in turn acts as an imperfect annotator to fine-tune the policy.

tion, and cross annotation scheme named entity recognition tasks[1]. The experiment results demonstrate the ability of our method to quickly learn a good policy directly on the task of interest. Compared to the previous work (Fang et al., 2017; Liu et al., 2018a) which transfers a policy learned on a source task to target task, our dream-based AL query policies are consistently more effective even when the data domain and annotation scheme of target task are different from the source task.

2 AL Query Strategy as a Sequential Decision Process

We consider the popular pool-based AL setting where we are given a small set of initial labelled data D^{lab}, and a large pool of unlabelled data D^{unl}, and a budget $\mathcal{B}$ 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 annotations were available, the performance of the underlying re-trained model $m_{\boldsymbol{\phi}}$ would be improved the most.

More specifically, a pool-based AL problem is a Markov decision process (MDP) (Bachman et al., 2017; Liu et al., 2018a), 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 labelled 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 is the improvement in the *generalisation* of the student learner. Assuming the availability of an evaluation set D^{evl}, the reward function can be formalised as:

$$\mathrm{R}(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}) = \mathrm{loss}(m_{\boldsymbol{\phi}_{t-1}}, D^{evl}) - \mathrm{loss}(m_{\boldsymbol{\phi}_t}, D^{evl}) \quad (1)$$

through a suitable *loss* function.

The goal is 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:

$$\mathbb{E}_{(D^{lab}, D^{unl}, D^{evl})\sim\mathcal{D}}\left[\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\sum_{t=1}^{\mathcal{B}} R(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})\right]\right] \quad (2)$$

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 annotation budget, i.e. the maximum number of queries made in an AL episode.

In the previous work, the distribution of AL problems is constructed via simulations on a *related* task for which enough labelled data exist. That is, the labelled data is randomly partitioned into the training, evaluation, and pool of unlabelled (by pretending the labels are unobserved) datasets. Answering the AL queries is easy in the simulations, as it does not involve actual interaction with the human annotator. As such, a large number of AL episodes can be simulated efficiently, allowing to learn a query *policy* using reinforcement (Fang et al., 2017; Bachman et al., 2017; Pang et al., 2018) and imitation (Liu et al., 2018a,b) learning algorithms. However, the effectiveness of the resulting query policy depends on the relatedness of the source and target tasks; a notion which is hard to formalise and evaluate in practice. Our goal in this paper is to learn the

[1]Source code is available at https://github.com/trangvu/alil-dream

Algorithm 1 Learning to AL by Dreaming

Input: labelled data D^{lab}, unlabelled pool D^{unl}, initial student model $\hat{\boldsymbol{\phi}}$, initial policy $\hat{\pi}$, dream episodes $\mathcal{E}$, dream length T_d , annotation budget $\mathcal{B}$, wake-dream cycles $\mathcal{W}$
Output: labelled dataset, trained model, policy

1: $\boldsymbol{\phi}_0 \leftarrow \hat{\boldsymbol{\phi}}$
2: $\pi_0 \leftarrow \hat{\pi}$
3: $T_w \leftarrow \frac{\mathcal{B}}{\mathcal{W}}$ ▷ length of the wake phase
4: **for** $t \in 1, \ldots, \mathcal{W}$ **do**
5: $D^{lab}, \boldsymbol{\phi}_t \leftarrow \text{wakeLearn}(D^{lab}, D^{unl}, \boldsymbol{\phi}_{t-1}, \pi_{t-1}, T_w)$
6: $\pi_t \leftarrow \text{dreamLearn}(D^{lab}, D^{unl}, \boldsymbol{\phi}_{t-1}, \pi_{t-1}, \mathcal{E}, T_d)$
7: **end for**
8: **return** $\boldsymbol{\phi}_{\mathcal{W}}$

query policy *directly* on the target AL task of interest, allowing for more effective query policies.

3 Dream-based Learning of AL Policy

In this section, we propose our sample-efficient AL policy learning method. While interacting with the human annotator, one may decide to split the total annotation budget $\mathcal{B}$ between two types of queries: (i) those which improve the underlying student learner based on the suggestions of the policy, and (ii) those which improve the policy. However, this approach may not make the best use of the annotation budget, as it is not clear whether the budget used to improve the policy (via the second type of queries) would pay back the improvement which could have been achieved on the student learner (via the queries of the first type).

Our approach aims to spend the annotation budget only for improving the student learner. To improve the policy, we use the trained student learner as an *imperfect annotator* in order to improve the policy via simulations using data of the AL task of interest. More specifically, our approach interleaves between querying the annotation of the selected data points to update the underlying student model, and improving the AL strategy using simulations; see Algorithm 1. As such, the AL policy is used to train the underlying student learner in the *wake* cycles through interactions with the human annotator (line 5 of Algorithm 1), and the student learner is used to train the query policy in the *dream* cycles via simulations (line 6 of Algorithm 1), which we elaborate in the following.

3.1 Wake Phase: Improving Student Learner

Assuming that the full annotation budget is $\mathcal{B}$ and the number of wake-dream cycles is $\mathcal{W}$, there are $T_w = \frac{\mathcal{B}}{\mathcal{W}}$ AL queries asked from a human annotator in each wake cycle; see Algorithm 2.

Algorithm 2 Wake Learn

Input: labelled data D^{lab}, unlabelled pool D^{unl}, student model $\boldsymbol{\phi}$, query policy π, wake length T_w,
Output: labelled dataset and trained model

1: **for** $t \in 1, \ldots, T_w$ **do**
2: $\boldsymbol{s}_t \leftarrow (D^{lab}, D^{unl}, \boldsymbol{\phi})$
3: $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x}' \in D^{unl}} \pi(\boldsymbol{x}'; \boldsymbol{s}_t)$
4: $\boldsymbol{y}_t \leftarrow \text{askHumanAnnotation}(\boldsymbol{x}_t)$
5: $D^{lab} \leftarrow D^{lab} + \{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$
6: $D^{unl} \leftarrow D^{unl} - \{\boldsymbol{x}_t\}$
7: $\phi \leftarrow \text{retrainModel}(\boldsymbol{\phi}, D^{lab})$
8: **end for**
9: **return** D^{lab} and $\boldsymbol{\phi}$

Algorithm 3 Dream Learn

Input: labelled data D^{lab}, unlabelled data D^{unl}, student model $\boldsymbol{\phi}$, policy $\hat{\pi}$, dream episodes $\mathcal{E}$, dream length T_d
Output: The learned policy

1: $M \leftarrow \emptyset$ ▷ the aggregated dreamt AL trajectories
2: $\hat{\pi}_0 \leftarrow \pi$
3: $D^{pool} \leftarrow \text{labelGen}(m_{\boldsymbol{\phi}}, D^{unl})$
4: **for** $\tau \in 1, \ldots, \mathcal{E}$ **do**
5: $D^{trn}, D^{evl} \leftarrow \text{dataPartition}(D^{lab})$
6: $M \mathrel{+}= \text{trajectoryGen}(D^{trn}, D^{evl}, D^{pool}, \pi_{\tau-1}, T_d)$
7: $\pi_\tau \leftarrow \text{retrainPolicy}(\pi_{\tau-1}, M)$
8: **end for**
9: **return** $\hat{\pi}_{\mathcal{E}}$

At each time step t of this real AL trajectory, the algorithm picks the query point suggested by the policy network (line 3 of Algorithm 2). As the policy network $\pi(.)$, we consider a feed forward neural network; see Figure 1(d), which assigns an importance score to each potential query from the unlabelled dataset $\boldsymbol{x}' \in D^{unl}$ in the current AL state $\boldsymbol{s}_t$. We summarise the AL state $\boldsymbol{s}_t$ by a fixed dimensional vector, consisting of the labelled and unlabelled datasets as well as the student learner; this is problem-specific and will be detailed in Section 4 for our classification and sequence labelling tasks. Together with the representation of the candidate $\boldsymbol{x}'$, they are fed to the policy network as the input. The label of the selected query is then asked from the human annotator (line 4 of Algorithm 2), and added to the labelled dataset to re-train the underlying student learner (lines 5-7 of Algorithm 2).

3.2 Dream Phase: Policy Improvement

In each dream cycle, the student learner teaches the AL querying policy and updates it; see Algorithm 3. We first generate the labels of the unlabelled data using the current student learner to get a pseudo-labelled data containing imperfect labels (line 3 of Algorithm 3). Afterwards, we synthesise AL tasks by randomly partitioning the collected

labelled data into training and evaluation sets, for each of which we simulate an AL trajectory efficiently using the pseudo-labelled pool to retrain the policy (lines 4-7 in Algorithm 3).

The policy can be re-trained using policy gradient algorithms, e.g. REINFORCE in deep reinforcement learning (RL) (Williams, 1988), or behavioural cloning in deep imitation learning (IL). We make use of DAGGER (Ross et al., 2011), a behavioural cloning algorithm for IL, which previous work has shown to be more effective than deep RL for learning AL policies (Liu et al., 2018a).

DAGGER with Imperfect Teacher To generate a simulated AL trajectory (line 6 in Algorithm 3), we run the querying policy for T_d time steps. For each time step t, we either select the next query based on the recommendation of the policy $\boldsymbol{b}_t = \arg\max_{\boldsymbol{x} \in D_t^{pool}} \pi(\boldsymbol{s}_t; \boldsymbol{x}', \hat{y}')$, or select the best query by one-step *roll-out*; that is

$$\boldsymbol{a}_t = \arg\max_{(\boldsymbol{x}', \hat{y}') \in D_t^{pool}} -\text{loss}(m_{\boldsymbol{\phi}'_t}, D^{evl}) \quad (3)$$

where $m_{\boldsymbol{\phi}'_t} = \text{retrainModel}(\boldsymbol{\phi}_{t-1}, (\boldsymbol{x}', y'))$. This choice is generated by the parameter β, which we refer to as the mixing coefficient. Importantly, the roll-out in Equation 3 uses the imperfect label $\hat{y}'$ for a candidate data point $\boldsymbol{x}'$. For computational efficiency, we take the maximisation in Equation 3 over a random subset of size k from the full data pool, as it involves retraining the underlying model and calculating the loss of the resulting model on the evaluation set. We refer to the above procedure to select actions $\boldsymbol{a}_t$ as the *imperfect algorithmic expert*.

To update the policy network (line 7 in Algorithm 3), we train it on a set of collected states paired with the imperfect expert's actions $M = \{(\boldsymbol{s}_i, \boldsymbol{a}_i)\}$ to maximize the objective $\sum_{i=1}^{|M|} \log Pr(\boldsymbol{a}_i | D_i^{pool})$, where $Pr(\boldsymbol{a}_i | D_i^{pool})$ is the probability of $\boldsymbol{a}_i$ being the best action among all possible actions in the data pool D_i^{pool} at state $\boldsymbol{s}_i$. The probability $Pr(\boldsymbol{a}_i | D_i^{pool})$ can be estimated using the preference score $\pi(\boldsymbol{a}_i; \boldsymbol{s}_i)$ computed by the AL policy π

$$Pr(\boldsymbol{a}_i | D_i^{pool}) = \frac{\exp \pi(\boldsymbol{a}_i; \boldsymbol{s}_i)}{\sum_{\boldsymbol{x} \in D_i^{pool}} \exp \pi(\boldsymbol{x}; \boldsymbol{s}_i)} \quad (4)$$

In addition to current trajectory, we make use of an experience replay memory $\mathcal{M}$ (Mnih et al., 2015) to store historic state-action transitions and random sample multiple mini-batches from it to retrain the policy network.

Unlabelled Candidate Selection. An important design consideration for our proposed algorithm is the selection of the unlabelled pool in each wake/dream cycle. To guide the policy toward selecting worthwhile datapoints, the candidate pool can be sampled randomly from a larger set of top uncertain and diverse datapoints in the wake cycles. During the dream cycles where the policy is strengthened based on the prediction of the imperfect expert, we can exploit the expert by sampling from its top confidence shortlist. We will see in the analysis that the candidate pool selection strategy further improves the quality of the student learner.

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 and cross-domain named entity recognition (NER), whereby an AL policy trained on a source domain/language is transferred to the target domain/language[2].

We compare our proposed dream-based AL policy learning method with the following baselines:

- *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, $\arg\max_{\boldsymbol{x}} -\sum_y p(y|\boldsymbol{x}, D^{lab}) \log p(y|\boldsymbol{x}, D^{lab})$
- *PAL*: A reinforcement learning based approach (Fang et al., 2017), which makes use of a deep Q-network to make the selection decision for stream-based active learning. It learns the policy on a source task and then transfers it to the target task.
- ALIL: An imitation learning based approach (Liu et al., 2018a), which transfer the learned policy from a source task to the target task without further fine-tuning.

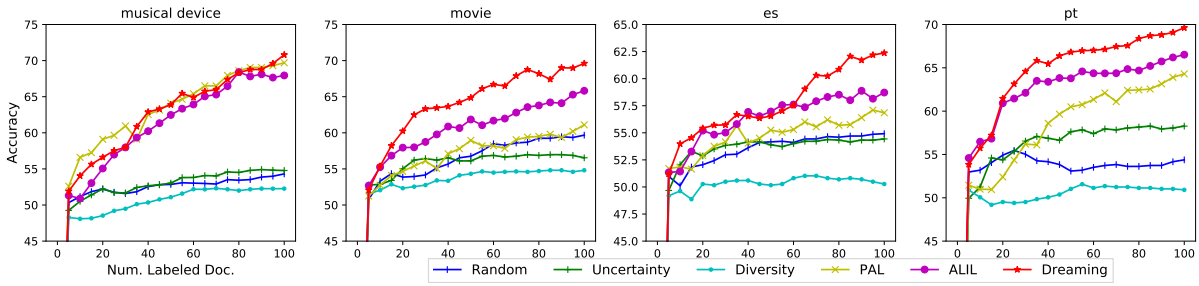


Figure 2: Accuracy of different active learning methods for cross domain sentiment classification (left two plots) and cross lingual authorship profiling (right two plots).

		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

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

4.1 Text Classification

Datasets and Setup. We run experiments on sentiment classification and authorship profiling tasks. Sentiment classification dataset were extracted from the Amazon product reviews (McAuley and Yang, 2016). The goal is to classify these reviews as positive or negative sentiments. The authorship profiling task aims to predict the gender of the text author. The data came 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). Table 1 shows data statistics for these two tasks.

For PAL and ALIL, the AL policy is first trained by AL simulation on the source task and then directly transferred to the target task. In our dream-based approach, the pretrained AL policy on the source task is used to warm-start the AL policy learned on the target task.

For training, 10% of the source data is used as the evaluation set to learn the best action in imitation learning. Following the experiment setting in Liu et al. (2018a), we run $T = 100$ episodes with the total annotation budget $\mathcal{B} = 100$ documents in each episode, set the sample unlabelled pool size $k = 5.$, and set the mixing coefficient in DAGGER $\beta = 0.5$. At transferring time, we take 90% of the target data as the unlabelled pool, and the remaining 10% as the test set. We set the number of wake-dream cycles $\mathcal{W} = 20$ which corresponding to the wake phase length $T_w = 5$. We set the number of dream episode $\mathcal{E} = 5$ and dream length $T_d = 10$. We run each AL method 20 times and report the average test accuracy w.r.t. the number of labelled documents selected in the AL process.

For the underlying model $m_{\boldsymbol{\phi}}$, we use a fast and efficient text classifier based on convolutional neural networks (CNN). More specifically, we apply 50 convolutional filters with ReLU activation and width of 3 on the embedding of all words in a document $\boldsymbol{x}$. 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. We use pretrained multilingual embeddings (Ammar et al., 2016) and fix these word embeddings during training for both the policy and the underlying classification model.

State representation. The AL state is a fixed dimensional vector, includes: (i) the candidate document represented by a CNN $\boldsymbol{h}(\boldsymbol{x})$, (ii) the distribution over the document's class labels $m_{\boldsymbol{\phi}}(\boldsymbol{x})$, (iii) the sum of all document vector representations in the labelled set $\sum_{\boldsymbol{x}' \in D^{lab}} \boldsymbol{h}(\boldsymbol{x}')$, (iv) the sum of all document vectors in the sample unlabelled pool $\sum_{\boldsymbol{x}' \in D^{pool}_{rnd}} \boldsymbol{h}(\boldsymbol{x}')$, and (v) the empirical distribution of class labels in the labelled dataset.

Results. Figure 2 shows the result on the product sentiment and authorship profiling tasks in cross-domain and cross-lingual AL scenarios. Our dream-based method consistently outperforms both heuristic-based, RL-based (PAL) (Fang et al., 2017) and direct-transfer IL (ALIL) (Liu et al., 2018a) approaches across all tasks. Our approach performs similar to the ALIL approach in the beginning of AL process and starts to outperform in

[2]Source code: https://github.com/trangvu/alil-dream

	musical	movie	es	pt
CL-warm-transfer (ALIL)	67.95	65.82	58.72	66.52
CL-cold-dream	62.32	64.86	57.43	58.30
CL-warm-dream	**70.80**	**69.60**	**62.37**	**69.62**
WL-warm-transfer (ALIL)	76.79	80.81	64.35	69.05
WL-cold-dream	76.00	80.07	63.78	68.56
WL-warm-dream	**77.92**	**81.62**	**67.57**	**70.70**

Table 2: Classifiers performance under different initialization settings of underlying classifier and AL policy. CL, WL denotes cold-start and warm-start classifier.

later cycles. We speculate that it is due to the noisy learning signal in the first few dream phases. In later cycles, the AL policy starts to adapt to the target task and learn to select good datapoints to train the underlying classifier.

We further investigate the combination of transferring the policy network with transferring the underlying learner. That is, we first train a classifier on all of the annotated data from the source domain/language; this classifier is then transferred to the target task and further fine-tuned using the collected labelled data. We compare the performance of the warm-start student learner (WL) and the random initialized cold-start student learner (CL) in different policy transfer scenarios: (i) warm-transfer (ALIL): the IL policy is transferred directly to target task, similar to ALIL approach (Liu et al., 2018a); (ii) cold-dream: our dream-based approach where the policy is initialized randomly; and (iii) warm-dream: our proposed approach where the pretrained policy is fine-tuned to the target task. The results are shown in Table 2. As anticipated, the cold-dream settings always perform worst in all tasks. In both warm-start and cold-start student learner scenarios, the warm-dream setting always outperforms the warm-transfer.

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 with four named entity classes. We convert the annotation to IO labelling scheme and train the policy on source language. We consider the bilingual and cross-annotation transferring scenario. More specifically, the English dataset with IO annotation is the source and other languages with either IO or IBO annotation are the target.

The CoNLL NER corpus of each language has three subsets: **train**, **testa** and **testb**. During policy training with the source language, we combine these three subsets, shuffle, and re-split them into simulated training, unlabelled pool, and evaluation sets in every episode. Following the experiment setting in Liu et al. (2018a), we also train the policy in $T = 100$ episodes with the budget $\mathcal{B} = 200$, and set the sample size $k = 5$ for the AL simulation on the source task,. At transferring time, we select $\mathcal{B}$ datapoints from **train** of the target language (treated as the pool of unlabelled data) and report F1 scores on **testa**. We set the number of wake-dream cycles $\mathcal{W} = 20$, dream episode $\mathcal{E} = 5$ and dream length $T_d = 10$. During wake cycle, we sample a subset of 10 unlabelled datapoints from the top 100 datapoints with the highest labelling uncertainty as the input to the policy network. In the dream phase, the sample pool is constructed randomly as usual.

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. For the word embeddings, we also use the pretrained multilingual embeddings (Ammar et al., 2016) with 40 dimensions and fix these during policy training.

State representation. The input to the policy network is the concatenation of:

(i) the representation of the candidate sentence using the sentence convolution network cnn_{sent} (Kim, 2014)

(ii) the representation of the labelling marginals using the label-level convolution network $\text{cnn}_{\text{lab}}(\mathbb{E}_{m_{\boldsymbol{\phi}}(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}])$ (Fang et al., 2017)

(iii) the bag-of-word representation of sentences in the sample pool of unlabelled data $\sum_{\boldsymbol{x}' \in D^{pool}_{rnd}} \sum_{w \in \boldsymbol{x}'} \boldsymbol{e}(w)$ where $\boldsymbol{e}(w)$ is embedding of word w

(iv) the representation of ground-truth labels in the labelled data $\sum_{(\boldsymbol{x}', \boldsymbol{y}') \in D^{lab}} \text{cnn}_{\text{lab}}(\boldsymbol{y}')$ using the empirical distributions

(v) the confidence of the sequential prediction $\sqrt[|\boldsymbol{x}|]{\max_{\boldsymbol{y}} m_{\boldsymbol{\phi}}(\boldsymbol{y}|\boldsymbol{x})}$

(vi) the representation of the entropy sequences for each word label in the sentence using another convolution network cnn_{ent}

(vii) entropy statistics includes max entropy, average entropy and sum entropy

In cross-annotation scheme scenarios, the AL policy is trained on source task with IO annotation

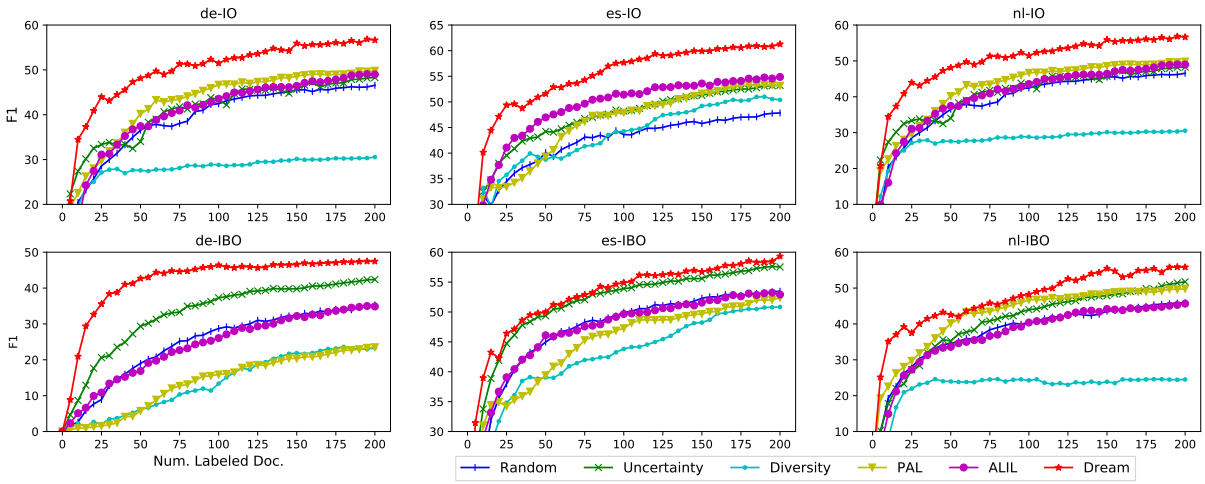


Figure 3: The performance of dreaming methods on bilingual settings under IO and IBO annotation scheme for three target languages: German (de), Spanish (es) and Dutch (nl).

and then later transferred to target task which is under IBO annotation scheme. With the same named entity set, the number of prediction classes under IO and IBO annotation scheme is 5 and 9 respectively. We only transfer the policy network and CNNs to the target task.

Results. Figure 3 shows the results for three target languages in cross-language and cross-annotation scheme scenarios. In bilingual and same annotation scenarios, our dream-based transfer method consistently outperforms other data-driven AL query strategy learning and heuristic methods. Specifically, diversity-based query strategy performs badly in almost every case because it ignores the labelling information. ALIL and PAL performance are either on par or slightly better than uncertainty sampling. However, these methods only perform similar to a random strategy when testing on new labelling scheme. Uncertainty sampling is still the best heuristic among other strategies. In cross-annotation scheme experiments, our proposed method surpasses the uncertainty-based strategy in German and Dutch, and achieves slightly higher score in Spanish. This suggests that uncertainty is a good informative measure, which is outperformed by our flexible and adaptive data-driven AL policy learning technique.

4.3 Biomedical Named Entity Recognition

In Section 4.2, we evaluated our approach on transferring the AL policy to a target task which shares the same labelling scheme as the source task. We further evaluate our methods in the scenario where the source and target tasks have different characteristic. Specifically, we conduct experiment on cross-domain cross-annotation NER.

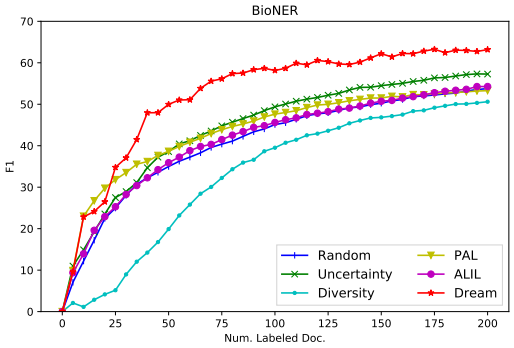


Figure 4: The performance of transferring trained policy on English NER to BioNER task.

Data and setup. We transfer the AL policy trained on the CoNLL2003 English NER task in the previous experiment, which is in the news domain, to the biomedical NER (BioNER) task. We use Genia4ER named entity corpus of MEDLINE abstracts from JNLPBA 2004 shared task.[3] The Genia4ER corpus is annotated in IBO2 scheme and contains five classes *protein*, *DNA*, *RNA*, *cell-line* and *cell-type*. The dataset has two subsets: training set of 18,758 sentences and test set of 3,918 sentences. We take out 1,758 sentences from the training set as validation set.

The experiment setup for policy transfer and underlying model is kept the same as in the NER experiments in Section 4.2. For the word embed-

[3] http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html

ding, we use the pre-trained English BioNLP embedding[4] (Chiu et al., 2016) with 200 dimension. Vocabulary size is set to 20,000.

Results. Figure 4 shows the F1 score on BioNER task. Similar to the bilingual cross-annotation NER experiment results, we observe that our dream-based approach outperforms all other strategies. AL policy learning methods from previous works perform on par with random query and slight worse than uncertainty.

We further compare the data selected by our dream-based method to other heuristic methods in terms of average length in every ten queried sentences. While the average sentence length in random strategy is consistently around 25-27 words, uncertainty strategy is bias toward very long sentences, up to more than 100 words in the first 20 queries, and gradually drops to 55 in the last 10 queries. Our dream-based method is also inclined to long sentence of 55-78 words, compared to random selection; but generally shorter than the uncertainty-based method.

5 Analysis

Sensitivity analysis. We evaluate the sensitivity with respect to the parameters in our proposed algorithm: the length of the wake phase T_w and number of dream episode $\mathcal{E}$. Given a fixed budget annotation, the wake phase length T_w determines the number of wake/dream cycles, the expert quality in the dream phase, and how often to retrain AL policy. The number of dream episode $\mathcal{E}$ decides how much adaptation to be performed to the AL policy. Results are shown in Figure 5. We observe some significant difference between each configuration only at the beginning of AL process where only a few labelled data are available.

Candidate selection strategy. We explore the effect of the candidate selection strategy on our dream-based AL policy learning. We consider two selection strategies in the wake phase: (i) *random* and (ii) *uncertainty* where a subset of 10 candidates are sampled from the top 100 datapoints with the highest labelling entropy. In the dream phase, five candidates are selected by the following strategies: (i) *random*, (ii) *certainty* where candidates are sampled from the top 100 low entropy labelling distribution, and (iii) *mixed* strategy whereby either *random* or *certainty* strategy are applied with probability of 0.5.

[4] https://github.com/cambridgeltl/BioNLP-2016

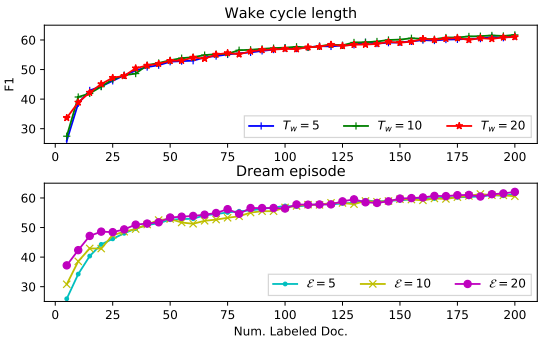


Figure 5: The performance of Spanish NER taggers respect to different wake phase length T_w and number of dream episode $\mathcal{E}$.

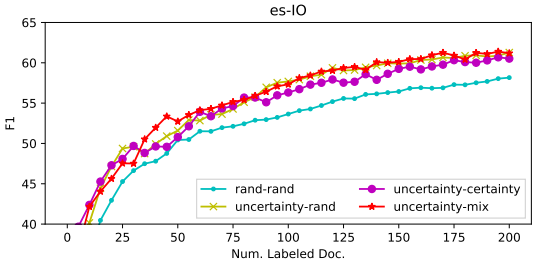


Figure 6: The performance of Spanish NER taggers under different candidate selection strategies.

Figure 6 shows the result of our dream-based approach on Spanish NER task. We observe that uncertainty strategy provides a better candidate pool for the AL policy to improve the student learner. Interestingly, random and mixed selection strategy seem to perform better than certainty strategy, especially in the later stages of the AL process where we have a better student learner. This suggests that exploration plays a more important role in strengthening the query policy.

6 Related Works

Heuristic-based AL. Traditional active learning algorithms rely on various heuristics (Settles, 2010) to guide the selection of most informative datapoints, 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). Combined with transfer learning, pre-existing labelled 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). However, these methods are not flexible to exploit characteristics inherent to a particular problem.

Policy-based AL. Recent research has formalized the AL process as a sequential decision process, and applied reinforcement/imitation learning to learn the AL query strategy (Woodward and Finn, 2017; Bachman et al., 2017; Fang et al., 2017; Liu et al., 2018a,b; Contardo et al., 2017). The AL policy learned via simulations on a source task for which enough labeled data exists. It is then transferred to related target tasks, e.g. in other languages or domains. However, the success of this approach heavily depends on the relatedness of the source and target tasks. Pang et al. (2018) has tried to address this problem by meta-learning a dataset-agnostic AL policy parmaterised by the dataset embedding. Konyushkova et al. (2018) has introduced a transferable AL strategy across unrelated datasets. In contrast, we learn a policy *directly* on the target task without requiring additional annotation budget.

Unsupervised Imitation Learning. From the theoretical perspective, unsupervised imitation learning has recently gained attention in machine learning (Torabi et al., 2018; Curi et al., 2018) and robotics (Piergiovanni et al., 2018). They consider a problem setup assuming the existence of an expert, where the expert's actions are unobservable but the world state transitions are observable, e.g. videos from a car driven by a human without observing the actual driving actions. This unsupervised imitation learning scenario is different from our more challenging problem setup, where we do not have an already-existing expert AL strategy to observe its world state transitions. We address the absence of the expert by exploiting the student learner as the imperfect annotator.

7 Conclusion

We have introduced a dream-based approach to directly learn pool-based AL query strategies on the target task of interest. Our approach is the first study to interleave (i) the wake phase, where the AL policy is exploited to improve the student learner and (ii) the dream phase, where the student learner in turn acts as an imperfect annotator to enhance the AL policy. This allows the learning of a policy from scratch, or adapt a pretrained AL policy on the target task, without requiring additional annotation budget. We provide comprehensive experimental results, comparing our method to strong heuristic-based and AL policy learning-based methods on several classification and sequence learning tasks, showing the effectiveness of our proposed method.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful comments and corrections. This work was supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE)[5].

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.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 166–174. Association for Computational Linguistics.

Gabriella Contardo, Ludovic Denoyer, and Thierry Artières. 2017. A Meta-Learning Approach to One-Step Active-Learning. In *International Workshop on Automatic Selection, Configuration and Composition of Machine Learning Algorithms*, volume 1998 of *CEUR Workshop Proceedings*, pages 28–40, Skopje, Macedonia. CEUR.

Sebastian Curi, Kfir Y. Levy, and Andreas Krause. 2018. Unsupervised imitation learning. *arXiv preprint arXiv:1806.07200*.

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

[5] www. massive.org.au

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.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

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.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. 2018. Discovering general-purpose active learning strategies. *CoRR*, abs/1810.04114.

Ming Liu, Wray Buntine, and Gholamreza Haffari. 2018a. Learning how to actively learn: A deep imitation learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1874–1883. Association for Computational Linguistics.

Ming Liu, Wray L. Buntine, and Gholamreza Haffari. 2018b. Learning to actively learn neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018, Brussels, Belgium, October 31 - November 1, 2018*, pages 334–344.

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.

Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy Hospedales. 2018. Meta-learning transferable active learning policies by deep reinforcement learning. *International Workshop on Automatic Machine Learning (ICML AutoML 2018)*.

AJ Piergiovanni, Alan Wu, and Michael S. Ryoo. 2018. Learning real-world robot policies by dreaming. *arXiv preprint arXiv:1805.07813*.

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*.

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. 2012. *Active Learning*. Morgan & Claypool Publishers.

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.

Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 4950–4957.

R. J. Williams. 1988. Toward a theory of reinforcement-learning connectionist systems. Technical Report NU-CCS-88-3, College of Comp. Sci., Northeastern University, Boston, MA.

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.