OPAD: An Optimized Policy-based Active Learning Framework for Document Content Analysis

Sumit Shekhar
Adobe Research
sushekha@adobe.com

Bhanu Prakash Reddy Guda
Adobe Research
guda@adobe.com

Ashutosh Chaubey
IIT Roorkee
achaubey@cs.iitr.ac.in

Ishan Jindal
IIT Roorkee
ijindal@ec.iitr.ac.in

Avneet Jain
IIT Roorkee
ajain1@ee.iitr.ac.in

Abstract

Documents are central to many business systems, and include forms, reports, contracts, invoices or purchase orders. The information in documents is typically in natural language, but can be organized in various layouts and formats. There have been recent spurt of interest in understanding document content with novel deep learning architectures. However, document understanding tasks need dense information annotations, which are costly to scale and generalize. Several active learning techniques have been proposed to reduce the overall budget of annotation while maintaining the performance of the underlying deep learning model. In this paper, we propose OPAD, a novel framework using reinforcement policy for active learning in content detection tasks for documents. The proposed framework learns the acquisition function to decide the samples to be selected while optimizing performance metrics that the tasks typically have. Furthermore, we extend to weak labelling scenarios to further reduce the cost of annotation significantly. We propose novel rewards to account for class imbalance and user feedback in the annotation interface, to improve the active learning method. We show superior performance of the proposed OPAD framework for active learning for various tasks related to document understanding like layout parsing, object detection and named entity recognition. Ablation studies for human feedback and class imbalance rewards are presented, along with a comparison of annotation times for different approaches.

1. Introduction

Documents are a key part of several business processes, which can include reports, business contracts, forms, agreements, etc. Extracting data from documents through deep networks have recently started gaining attention. These tasks include document page segmentation, entity extraction or classification. Fueled by the availability of both labeled and unlabeled data, and advances in the computation infrastructure, recently, a number of deep learning models have been proposed for modeling complex tasks [12,23,39]. The promising results from this research direction motivated development of several deep learning models which show significant performance improvements on these tasks when trained on a large amount of labelled data [35,53,55]. However, deployment of these models requires considerable *effort* and *cost* to annotate unlabelled data especially for document tasks because of requirements for dense annotations, e.g. annotating page structures with components like *title*, *table*, *figures* or *references*. Thus, there is a need to explore methods to optimize annotation budgets to accelerate the development of document analysis models.

Several approaches have been proposed in the domain of semi-supervised learning [56], unsupervised learning [52], few-shot learning [51], active learning [42] etc... to overcome the limitation of availability of labeled data. Each of these approaches have their own objectives incorporated in either modeling or data annotation or both for achieving superior performances in a limited annotated data setup. Among these, our motivation for using active learning is two-folds: (1) active learning bridges the gap in the model by querying samples in the data space, for which the model does not have enough information [42], (2) the active learning approaches seek to learn higher accuracy models within a given annotation cost, through optimizing data acquisition, which align well with our objective of optimizing annotation costs. Recent methods for *pool-based active learning* scenario, the query for annotations selects a subset batch of data samples for the oracle (*i.e.* the annotator). Pool or batch-based active learning methods are more scalable than querying single data sample per learning cycle [20]. Most of

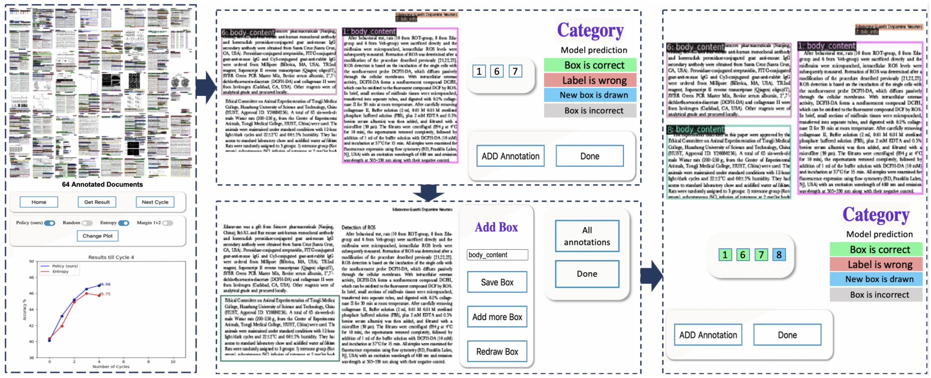


Figure 1. The proposed active learning-based interface, OPAD, enables intelligent annotation functionalities like optimized selection of documents for layout detection task, annotating instances and text with boxes (strong labelling) or verification and correction of annotations (for weak labelling scenarios). *Document is cropped for better visualization*

the active learning work [2, 42] formulate acquisition functions as information theoretic uncertainty estimates. While uncertainty-based methods work well for tasks like classification [17, 49], where a single annotation is required per data sample, generalizations to document tasks such as page segmentation and named entity recognition, which require multiple annotations per selected data sample, have been scarcely explored. This is because methods to aggregate uncertainties over various entities present in a data sample are not well developed [7, 41]. Recent techniques have been proposed to obtain a better acquisition function for active learning in these tasks [29, 31]. However, these methods assume highly task-specific heuristics, and hence can not be generalized across different content detection scenarios.

In addition to active learning, in particular for dense annotation tasks in documents, weak learning can be an effective approach to reduce annotator's efforts [36, 37, 50]. When there are multiple entities to be annotated in a data sample, weak learning reduces the annotation effort, either by providing faster variations of annotation techniques [37] or simply asking the annotator to verify the model predictions [36]. However, there are very few works [8, 11] that combine weak learning with active learning. Furthermore, to the best of our knowledge, none of the works takes advantage of the annotator feedback (e.g. from annotator's corrections of detected instance boundaries) during an active learning cycle.

In this work, we propose a policy-based active learning approach, taking into account the complexities of aggregating model uncertainties in the selection of samples to be labelled. We model the task of active learning as a Markov decision process (MDP) and learn an optimal acquisition function using deep Q-learning [34]. While several works rely on reinforcement learning for learning an optimal acquisition function [9, 22, 29, 31], they assume task-specific representations of states and actions and hence are not generalizable across tasks. We further show that the proposed method can be combined with weak labelling, reducing the cost of annotation compared to strong labelling. Moreover, we incorporate class imbalance and human feedback signals into the design of MDP using suitable reward functions to further improve the performance of our approach.

To summarize, the major contributions of our work are as follows:

- We propose a policy-based task-agnostic active learning approach for complex content detection tasks, layout detection and named entity recognition in documents.
- We report that the proposed approach is generalizable, through demonstrating the performance of our active learning setup on varied detection tasks.
- We investigate the effectiveness of incorporating class balance and human feedback rewards in improving the active learning policy.
- We demonstrate the advantage of the proposed approach in reducing the costs of annotation in aforementioned complex detection tasks.

Throughout the remainder of the paper, we explain the proposed concepts, models, configurations, and discussions from the perspective of the layout and object detection, and named entity recognition tasks.

2. Related Work

Document content analysis has been studied extensively along several dimensions such as document classification (image [53, 54] or text [1, 38] or both [4, 25]), named entity recognition in documents [32, 55], content segmentation [19, 35], document retrieval [10, 45, 48], layout analysis [5] among many others. The availability of large scale labeled datasets of documents [21, 26, 27, 47, 58] led to the advent of several state-of-the-art deep learning models which have significantly improved these tasks in a large scale data setup. However, to the best of our knowledge, there is very limited amount of literature which uses active learning to optimize data annotation cost in a low resource setting, specifically for document analysis tasks [6, 18]. Therefore, in this section, we discuss about works that deal with general active learning policies, and active learning in a couple of related well studied domains, image classification, object detection and named entity recognition.

Active learning selects data samples with high uncertainty in the model prediction, which can provide more information to the underlying model. Different works have proposed different ways to compute model uncertainty [42]. While some methods depend on information theory for designing acquisition functions [17, 24, 49], others rely on alternative ways to approximate model uncertainty [13, 16]. Yoo *et al* [57] add a light-weight loss prediction module to the prediction model to predict the loss for the unlabelled samples, and use that as an uncertainty measure. Mayer *et al* [33] use uncertainty measure to find the optimal sample and query the data sample closest to the optimal sample.

For complex tasks such as object detection and named entity recognition, recent works [7, 41, 44] have been proposed to use uncertainty scores for the acquisition of samples. Most of these methods rely on aggregating the uncertainties of various entities within a data sample using max, sum or average functions [7,41]. Aghdam *et al* [3] proposed a novel approach combining pixel-level scores to obtain an image-level score for doing active learning for the task of pedestrian detection task.

Several works have been proposed to incorporate reinforcement learning to learn an optimal acquisition function for active learning. The objective of these approaches is to model the active learning process into a Markov decision process through defining and designing suitable representations for states, actions, and rewards [15, 22, 30]. Liu *et al* [29] proposed an imitation learning approach for active learning in tasks related to natural language processing, relying on an algorithmic expert to find an optimal acquisition function. We differ from the work of Casanova *et al* [9] on using reinforced active learning approach for image segmentation, in terms of the generalize-ability of our approach on various tasks. We also report the effectiveness of using weak learning on top of policy-based active learning in consuming the budget with maximum efficiency.

3. Proposed OPAD Framework

In this section, we describe the proposed **O**ptimized **P**olicy-based **A**ctive Learning Framework for **D**ocument Content Analysis, *OPAD*. Figure 1 shows the interface for *OPAD*, which enables various scenarios of detection tasks for human annotators. The underlying algorithm for *OPAD* is a Deep Query Network (DQN)-based reinforcement learning policy, optimized for data sample selection based on the performance metrics for the task. *OPAD* has two stages - policy training stage and deployment stage. In the policy training stage, *OPAD* is trained using simulated active learning cycles to maximize performance on a validation set. While deploying, the trained policy is used to make online batch selection for annotation. The overall formulation for OPAD is described below.

3.1. Formulation

The underlying objective for policy training in *OPAD* is to perform an iterative selection of the samples from an unlabelled pool, X_u, which would maximally increase the performance of the model being trained, Θ until the annotation budget, $\mathbb{B}$ is consumed. In each active learning cycle, the policy DQN Π [34] selects a batch of n_{cycle} samples, which are labelled, and added to the set of labelled samples X_l. The detection model Θ is then trained for a fixed number of epochs using the expanded set, X_l. The reward for the policy network for selecting the samples is the performance of the underlying model Θ computed using a metric apropos to the task (e.g. *Average Precision* for layout detection, and *F-score* for named entity recognition) on a separate held-out set, X_{met}. The training of the policy Π is performed through episodes of active learning.

Notations	Description
X_{train}, X_{val}, X_{test}	Train, Validation and Test sets of a given dataset
X_u, X_l, X_{init}	Unlabelled, labelled, and initial labelled sets
X_{cand}	Candidate unlabelled examples for an active learning cycle
X_{met}, X_{state}	Metric calculation set, State representation set
$\mathcal{A}_t$, $\mathcal{S}_t$, $\mathcal{R}_t$	Action, State and Reward at time t
Π, Θ	Policy deep Q network and Prediction model to be trained
$\mathbb{M}$, $\mathbb{B}$	Memory buffer for Q learning, Total budget for active learning
n_{cycle}, n_{pool}, n_{init}	Number of samples to be acquired in one active learning cycle, Number of samples in a pool, Number of samples labelled for initial training

Table 1. Notations used to represent various data splits and model components.

We now describe various components of the proposed policy-based active learning approach in details.

3.2. Data Splits

Given a dataset $\mathbb{D}$, we split the samples (or use the existing splits of the dataset) into X_{train}, X_{val}, and X_{test} sets. For the two stages of *OPAD*, the further splits are as follows.

During policy training stage We separate a set of samples X_{met} along with their labels from X_{train}, which is used for validating the performance of underlying model Θ and computing rewards for training the policy DQN Π. For the RL setup of the policy DQN, we use a held-out set X_{state} which is used together with X_{cand} later to compute overall state representation. Note that, unlike [9], we do not require labels for X_{state}, which further reduces the annotation budget. During this stage, we train the detection model Θ on X_l, which is initialized with X_{init} and populated with samples from X_u as the active learning progresses. Here, X_{init} is a set with n_{init} randomly selected samples with the corresponding labels for initial training of the model Θ. Therefore, before the active learning process starts, X_u equals $X_{train} - \{X_{init} + X_{state} + X_{met}\}$, and X_l equals X_{init}.

During deployment stage We utilize the X_{val} set for training the detection model Θ. We make this differentiation from the policy training stage to ensure that sample selection by the policy happens on an unseen set. During this stage, we use the same terminology X_{init}, X_l, and X_u from the previous stage. However, the n_{init} samples in X_{init} set are selected from the X_{val} set and therefore, at the start of the active learning process X_u equals $X_{val} - \{X_{init}\}$, and X_l equals X_{init}. We use the same set of examples for the state computation set X_{state}. In this stage we do not require the X_{met} set.

Though we have ground truth annotations available for all the samples in all the three sets, to simulate the annotation setup, we mask this data from both Θ and Π models and utilize the labels as and when required.

3.3. Active Learning

[!h] [1] **Input:** X_{train}, budget $\mathbb{B}$ **Output:** Policy DQN, Π, trained for querying the samples for annotation Randomly sample examples from X_{train} to form X_{state} and X_{met} sets. Initialize policy and target DQN Initialize memory replay buffer $\mathbb{M}$ convergence of DQN loss Initialize Θ Randomly sample n_{init} from $X_{train} - \{X_{state} + X_{met}\}$ to form X_{init} Initialize X_u to $X_{train} - \{X_{state} + X_{met} + X_{init}\}$ Initialize X_l to X_{init} Train the model Θ on X_l Compute the performance metric on X_{met} Consumption of budget $\mathbb{B}$ Sample $n_{pool} \times n_{cycle}$ number of samples from X_u as candidates for labelling X_{cand} Compute state representation $\mathcal{S}_t$ using predictions of model Θ on X_{state} and X_{cand}

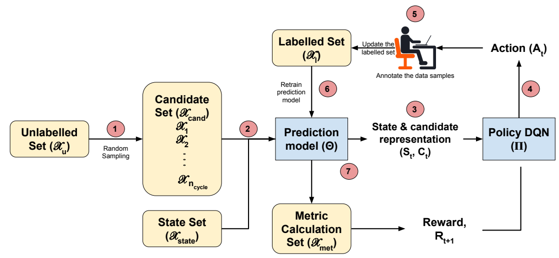


Figure 2. **Overview of the policy training in *OPAD*** - (1) Candidate samples are chosen randomly from the unlabelled pool X_u. (2) State representation is calculated using X_{cand} and X_{state}, which is then passed to the policy DQN Π to select the samples to be annotated (3, 4 and 5). (6) The labelled set X_l is then updated and the model Θ is retrained. (7) Finally, reward is computed using the set X_{met}.

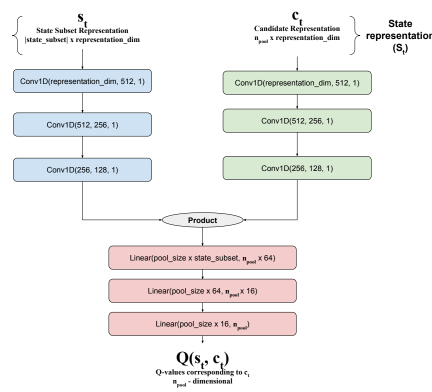


Figure 3. Architecture of the proposed Deep Query Network, Π for the policy.

Select n_{cycle} samples from X_{cand} using ϵ-greedy policy and add it to X_l - Action $\mathcal{A}_t$ Retrain the model Θ on X_l Compute the metric on the X_{met} Compute the reward $\mathcal{R}_{t+1}$ as the difference in metric Re-do steps 14 and 15 - Next State $\mathcal{S}_{t+1}$ Add tuple $(\mathcal{S}_t, \mathcal{A}_t, \mathcal{R}_{t+1}, \mathcal{S}_{t+1})$ to the memory replay buffer $\mathbb{M}$ Optimize policy DQN, Π Figure 2 shows an overview of active learning (*inner while loop* at step 11 in Algorithm 3.3) in a single episode of policy training. In an active learning cycle, we select $n_{pool} \times n_{cycle}$ number of samples from the set X_u, which represent the candidates selected for the current active learning cycle X_{cand}. The policy DQN Π computes Q-value for samples within each pool containing n_{pool} samples, based on candidate set X_{cand} and state representation set X_{state}. The policy selection network is optimized to maximize the reward, $\mathcal{R}_t$:

$$Q^*(\mathcal{S}_t, \mathcal{A}_t) = \max_{\Pi} \mathbb{E}[\mathcal{R}_{t+1}|\mathcal{S}_t, \mathcal{A}_t, \Pi] \tag{1}$$

The annotator then annotates the selected samples, and the labelled set X_l is updated by adding these new samples.

We then retrain the model Θ using the updated labelled set and finally calculate the reward for the current cycle $\mathcal{R}_t$ by measuring the performance of the model Θ on X_{met}.

$$\mathcal{R}_{t+1} = Performance_{t,X_{met}} - Performance_{t-1,X_{met}} \tag{2}$$

where $Performance$ is measured in terms of *AP metric* for layout and object detection tasks, and *F-score* for named entity recognition task. Algorithm 3.3 summarizes the training phase of the proposed approach.

3.4. Policy Training Stage

Policy Network Our policy network Π is a deep query network, as shown in Figure 3. The underlying prediction model Θ computes the representations $\mathbf{c_t}$ and $\mathbf{s_t}$ from the sets X_{cand} and X_{state} respectively (details in Section 4.3). The policy network then receives the two inputs s_t, and c_t, which we denote as the state representation $\mathcal{S}_t$ in Figure 3. We pass the two representations through convolution layers, followed by vector product of state and candidate representations. The final Q-value is obtained by passing the combined representation through fully connected layers.

Policy Optimization The computed Q-value is used for selecting n_{cycle} samples at each step. For this, a memory or experience replay buffer, $\mathbb{M}$ is created using MDP state representation tuples, $(\mathcal{S}_t, \mathcal{A}_t, \mathcal{R}_{t+1}, \mathcal{S}_{t+1})$. Further, as a batch of n_{cycle} needs to be selected, the candidate set, $\mathcal{X}_{cand}$, is randomly partitioned into n_{cycle} mini-batches, and action set $\mathcal{A}_t$ is set to $\mathcal{A}_{ti=1}^{i n_{cycle}}$. The loss is then optimized as follows to train the policy network:

$$Loss(\Pi) = \mathbb{E}_{t \in \mathbb{M}}[(\mathcal{Y}_t^i - Q(\mathcal{S}_t, \mathcal{A}_t^i)); \Pi))^2] \tag{3}$$

The values for $\mathcal{Y}_t^i$ are computed using a double DQN formulation [22] incorporating a target network, Π' for stable training:

$$\mathcal{Y}_t^i = \mathcal{R}_{t+1} + \max_{\mathcal{A}_{t+1}^i} \gamma Q(\mathcal{S}_{t+1}, \mathcal{A}_{t+1}^i; \Pi'); \Pi) \tag{4}$$

where, γ is the discount factor for future reward, set to 0.9 in our experiments.

ϵ-greedy selection To encourage exploration of diverse samples by the policy during training, an ϵ-greedy strategy is followed while training the policy, which selects a random sample for the action $\mathcal{A}_t^i$ with probability $epsilon$, instead of the sample maximizing Q-value. The ϵ value starts with 0.9 for the initial cycle, and decreases by a factor of 0.1 for subsequent cycles. For policy deployment, ϵ is set to 0. The gradient optimization is done using the temporal difference method [46].

3.5. Deployment Stage

[1] **Input:** X_{val}, X_{test}, X_{state}, budget $\mathbb{B}$ Randomly sample n_{init} from X_{val} to form X_{init} Initialize X_u to $X_{val} - \{X_{init}\}$ Initialize X_l to X_{init} Initialize Θ Train the model Θ on X_l Compute the performance metric on X_{test} Consumption of budget $\mathbb{B}$ Sample $n_{pool} \times n_{cycle}$ number of samples from X_u as candidates for labelling X_{cand} Compute state representation $\mathcal{S}_t$ using predictions of model Θ on X_{state} and X_{cand} Select n_{cycle} samples from X_{cand} using ϵ-greedy policy and add it to X_l - Action $\mathcal{A}_t$ Retrain the model Θ on X_l Compute the metric on the X_{test} and report

Algorithm 3.5 summarizes the deployment stage (or policy testing stage). We freeze the parameters of the model Π in this stage. We use the X_{val} set to iteratively select the samples and train the model Θ. At the end of each active learning cycle we compute the performance of the model Θ on the held-out set X_{test} and report the values in Section 4.

3.6. Weak labelling

In a usual annotation scenario (as shown in Figure 4 - top), the annotator has to mark all the entities present in a sample by drawing the bounding boxes and selecting labels for them. To reduce the annotation cost, we propose a weak labelling annotation framework (Figure 4 - bottom). Inspired from [36], the annotator is shown the document as well as the predictions with high confidence from the model Θ for that document. The annotator can then (1) add a missing box, (2) mark a box either correct or incorrect, and (3) mark a label either correct or incorrect for the associated box. The annotation interface for the weak labelling approach is shown in Figure 1.

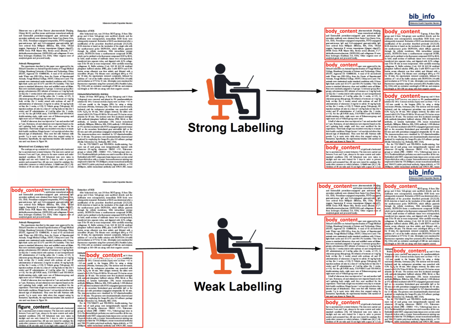


Figure 4. **Weak labelling in the case of layout detection.** In the top image, the annotator has to draw and mark all the layout boxes, while in the bottom image, the annotator can verify the predictions of the model in the input image, and add new boxes. Image is best viewed in color.

The advantage of weak labelling is that it significantly reduces the annotation time. Annotation of a new entity by drawing a bounding box or selecting words takes ~ 15 seconds on an average in the case of detection tasks and ~ 4 seconds in case of named entity recognition. Verifying an entity takes ~ 5 seconds for layout detection task and ~ 2 seconds for named entity recognition[1].

3.7. Additional Rewards

We propose the following additional rewards to improve the performance of the active learning approach.

- **Class balance reward:** To reduce class imbalance in the newly acquired samples that are to be labelled, X_{new}, we propose an additional class distribution entropy reward which reinforces a class-balanced selection of samples.

$$\mathcal{R}_{cls\_ent} = \mathcal{H}(P(X_{new})) \tag{5}$$

  where $\mathcal{H}$ is the Shannon entropy function [43], and $P(X_{new})$ is the probability distribution over various classes for the newly acquired samples X_{new}.

- **Human feedback reward:** In a weak labelling scenario, where the annotator can modify the output from the prediction model, Θ, a human feedback signal could be added at each active learning cycle while training the policy. The objective is to promote the selection of those samples for which the annotator modifies the high confidence predictions of Θ heavily because such samples would be more informative for the model Θ. Accordingly, the additional human feedback reward for detection during training time is given as,

$$\mathcal{R}_{feedback} = AP_{after\_feedback} - AP_{before\_feedback} \tag{6}$$

  where $AP_{after\_feedback}$ is the AP metric on the newly acquired samples, after the annotator has verified the predictions, and $AP_{before\_feedback}$ is the AP of the samples before feedback.

4. Experiments and Results

In this section, we provide a comprehensive experimental evaluation of the proposed policy-based active learning approach on the document understanding tasks, document layout detection and named entity recognition. Furthermore, we also evaluate our models on Pascal VOC object detection task to demonstrate the generalizability of the proposed solution across different domains.

4.1. Datasets

We use the following datasets for the corresponding tasks:

- **GROTOAP2** [47] dataset is used for the complex document layout detection task. The dataset consists of 22 layout classes for scientific journals. We sampled two sets of 5000 images as training and validation sets. Among these, we hold-out 10% for reward computation set X_{met} and 256 random samples for X_{state} and use the remaining samples for the active learning setup. We use the validation set for simulating the active learning during the deployment phase and finally report the performance on a held-out subset of 2500 images. Further, we merged those classes having very few instances (e.g. *glossary*, *equation*, etc.) with the *body content* class, resulting into a modified dataset with 13 classes.

- **Pascal VOC-2007** [14] dataset with 20 object classes is used for the object detection task. We use the *train* set of VOC-2007 containing 2501 images during the policy training phase. Similar to layout detection task, we hold-out 10% for reward computation set X_{met} and 256 random samples for X_{state} and use remaining samples for the active learning setup. During the deployment phase, we utilize the *val* set of VOC-2007 containing 2510 images for simulating the active learning setup i.e selecting samples using trained Π model and training the model Θ. We use the *test* set of VOC-2007 consisting of — samples for reporting the performance of model Θ after each active learning cycle during the deployment stage.

We also use the following datasets for pre-training the underlying model Θ:

- **PubLayNet** [58] We use this dataset for pre-training Θ for document layout detection. This dataset contains over 360K page samples and has typical document layout elements such as *text*, *title*, *list*, *figure*, and *table* as the annotations. While the *list*, *figure*, *table* and *title* classes contains the corresponding information from document, the *text* category consists of the rest of the content such as author, author affiliation; paper information; copyright information; abstract; paragraph in main text, footnote, and appendix; figure & table caption; table footnote.

- **MS-COCO** [28] This dataset consists of 91 object classes. We use this dataset to pre-train the underlying classification model Θ (i.e. Faster-RCNN model) in the case of object detection on the VOC dataset. We pre-train the model Θ on this dataset and remove the last layers from both the class prediction and bounding box regression branches which are class-specific.

4.2. Models and configurations

We use the Faster-RCNN model [40] with RESNET-101 backbone[2] [23] as the underlying prediction model for the

[1] All the mentioned values are average annotation times of 3 individuals measured on the developed annotation tool

[2] https://github.com/facebookresearch/detectron2

layout detection and object detection tasks. The Faster-RCNN model is pre-trained on a subset of 15000 images from PubLayNet [58] dataset for the layout detection task, and on MS-COCO [28] dataset for the object detection task to bootstrap the active learning experiments.

For active learning, we use a seed set of 512 labelled samples in case of detection tasks initially. The Faster-RCNN model is trained for 1000 iterations on the labelled set in an active learning cycle. In each of the 10 active learning cycles we select 64 samples for the detection tasks, from unlabelled dataset for labelling giving a total of 1152 and 350 labelled samples in a single episode for detection tasks and NER respectively. We run 10 episodes of these active learning cycles to train the policy network. The learning rate for training the policy DQN is set to 0.001 with a gamma value of 0.998. The learning rates of Faster-RCNN is set to 0.00025. We also apply a momentum of 0.95 to optimize the training of policy network. We set the size of memory replay buffer $\mathbb{M}$ to 1000 samples with first-in-first-out mechanism.

4.3. MDP state representation

For the layout detection and object detection tasks, we use a randomly sampled set of 256 images from the *train* set as the subset for representing the overall distribution of the dataset (X_{state}). We pass each instance from the candidate (X_{cand}) and state (X_{state}) subsets through the Faster-RCNN model, to get the top 50 confident bounding box predictions. We concatenate the class scores for these top 50 predictions to the feature map of RESNET-101 backbone to get a final representation (1256-dimension for VOC-2007, and 906-dimension for GROTOAP2) for each sample in the candidate and state subset sets. The representations thus obtained from the samples in X_{cand} are stacked to form $\mathbf{c_t}$, and similarly $\mathbf{s_t}$ from the set X_{state}. Together $\mathbf{c_t}$ and $\mathbf{s_t}$ form the state representation $\mathcal{S}_t$ in Figure 3.

4.4. Human Annotation Simulation

To simulate the role of a human annotator for weak labelling, we use the ground truths of the datasets on which we perform our experiments. In detection tasks (i.e. layout detection and object detection), we consider the predictions which have an IoU greater than 0.5 with the ground truth box as the boxes being marked as correct by the annotator. For those boxes in the ground truth which do not have any prediction with IoU greater than 0.5, we include that box into the labelled set marking as a full annotation (a strong label).

4.5. Results

We compare the performance of our proposed method with three baselines -

	Method↓	Avg time (seconds)⟶ GROTOAP2	VOC2007
Strong	Random	10m14s	12m21s
	Entropy Max	18m14s	17m16s
	Entropy Sum	18m01s	15m53s
	Margin	18m00s	15m39s
	OPAD	11m22s	14m00s
Weak	Random	10m23s	12m24s
	Entropy Max	18m20s	17m31s
	Entropy Sum	18m10s	15m26s
	Margin	18m03s	15m48s
	OPAD	11m36s	14m12s

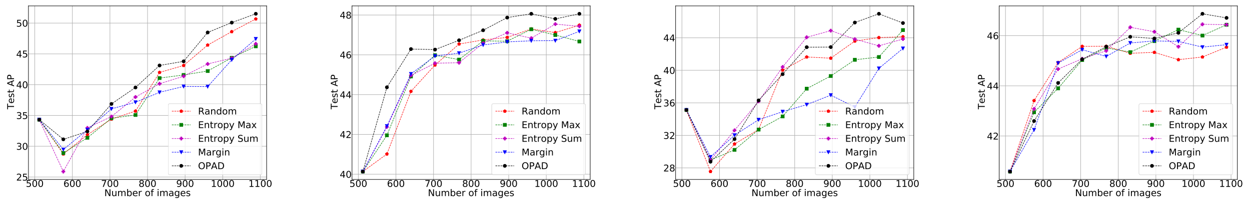
Table 2. Time required for one active learning cycle i.e selection of samples for various algorithms along with the model training time. Note that the model training time is constant.

	Method↓	Annotation time required(seconds)⟶ GROTOAP2	VOC2007
Strong	Random	72500	9000
	Entropy Max	81600	10500
	Entropy Sum	81200	10000
	Margin	92700	11000
	Ours	**66000**	**7000**
Weak	Random	38000	4250
	Entropy Max	39000	**2000**
	Entropy Sum	41000	2500
	Margin	48000	7500
	Ours	**33000**	2250

Table 3. Annotation time required to reach an AP of 42.5 on GROTOAP2 and an AP of 45.5 on VOC-2007. These values indicate the minimum achievable best performances by all the models on the datasets.

- **Random** Data samples from the unlabelled pool are randomly chosen for annotation.
- **Entropy** [41] For the entropy-based selection, first the entropy of class prediction probability by Θ is computed over all the entities of a data sample. We present results for aggregating entropy of a single sample in two ways: 1. maximum entropy, 2. sum of entropy of all detected entities within the sample, and then the samples with the highest aggregate entropy are selected for labelling.
- **Margin** [7] Similar to entropy, a v_{1vs2} margin score is computed using the difference of prediction probability of highest and second highest class for all the instances of a sample. Then, the maximum margin score over all the instances is taken to be the aggregate margin measure for the sample. Samples with the highest aggregate margin are selected for labelling. The baseline metrics are as described in the existing prior art.

Figure 5 shows the accuracy of all the methods on the test sets of different datasets, for both strong and weak labelling settings. We can observe that the proposed policy-based



(a) GROTOAP2 strong labelling (b) VOC-2007 strong labelling (c) GROTOAP2 weak labelling (d) VOC-2007 weak labelling

Figure 5. Plots showing the performance of the methods, *viz.* random, entropy, margin and proposed, for GROTOAP2 and VOC-2007 for both strong and weak labelling settings.

AL method significantly outperforms the baseline methods. This is because of the optimized selection policy, learned to reward the better performance of the prediction model. While the curves for VOC-2007 approach saturation, we stop the GROTOAP2 training before reaching saturation as our objective is to show the performance of the underlying model with a limited budget. Note that the proposed method uses vanilla reward in all the plots in Figure 5. Further, as shown in Table 2 and Table 3, the proposed method takes significantly less time for annotation than the baselines to reach the minimum best performance achievable by all the models, while performing only next to random algorithm for sample selection timings. The annotation times in Table 3 are based on the number of samples selected for annotation multiplied by the average human annotation times mentioned in Section 3.6.

5. Ablation Study

In this section we discuss the importance of the proposed additional rewards in improving the performance of the proposed AL approach.

5.1. Class balance reward

We conduct ablations by adding the class distribution entropy reward (Equation 5) to the vanilla reward function. The overall reward function is:

$$\mathcal{R}_{overall} = \mathcal{R}_t + \lambda * \mathcal{R}_{cls\_ent} \tag{7}$$

where λ is a hyper-parameter, and $\mathcal{R}_t$ is the vanilla reward. As seen in Table 4, we observe a significant increase in performance as compared to the vanilla reward policy.

5.2. Human feedback reward

In this experiment we report the effect of adding human feedback to the vanilla reward, i.e.

$$\mathcal{R}_{overall} = \mathcal{R}_t + \lambda * \mathcal{R}_{feedback} \tag{8}$$

where λ is a hyper-parameter. We report the results of using this overall reward in our policy in Table 5, along with the baselines and vanilla policy in a weak labelling setup. We observe that having a small weight on the feedback reward results in a jump in the performance.

Method ↓	AP GROTOAP2	F-score VOC-2007
Random	50.668	47.490
Entropy Max	46.229	46.671
Entropy Sum	46.634	47.431
Margin	47.428	47.179
OPAD	51.508	48.061
OPAD (ClsEnt λ - 0.25)	**53.241**	47.727
OPAD (ClsEnt λ - 0.50)	51.185	47.701
OPAD (ClsEnt λ - 0.75)	52.143	**48.566**
OPAD (ClsEnt λ - 1.0)	51.530	48.060

Table 4. Performance of our method on test data with class distribution entropy reward on various datasets. The total budget is 1152 samples for GROTOAP2 and VOC-2007.

	AP ⟶	
Method ↓	GROTOAP2	VOC2007
Random	44.127	45.541
Entropy Max	44.951	46.433
Entropy Sum	43.842	46.437
Margin	42.690	45.639
OPAD	45.813	46.708
OPAD (Feedback λ - 0.1)	**48.524**	**47.238**
OPAD (Feedback λ - 0.25)	46.266	46.835
OPAD (Feedback λ - 0.40)	44.899	46.646
OPAD (Feedback λ - 0.70)	44.839	46.071
OPAD (Feedback λ - 1.0)	44.110	46.304

Table 5. Performance of our method with human feedback reward for weak labelling on GROTOAP2 and VOC2007. AP after consuming a total budget of 1152 samples.

6. Conclusion and Future Works

We present a robust policy-based method for active learning task in complex content detection problems. The problem of active learning in detection is formulated using a DQN-based sampling network, optimized for task performance metrics. We extend the active learning setting to weak labelling, and propose rewards for class balance and human feedback. To the best of our knowledge, this is first-of-its-kind work optimizing active learning for detection tasks in documents. We show the efficacy of the proposed methods on a large document detection set as well as object detection. As a future direction, we would like to improve on the DQN, and further explore more recent active learning acquisition functions.

References

[1] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*, 2019. 2

[2] Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. *Active learning: A survey*, pages 571–605. CRC Press, Jan. 2014. 2

[3] Hamed H. Aghdam, Abel Gonzalez-Garcia, Antonio Lopez, and Joost Weijer. Active learning for deep detection neural networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. 3

[4] Nicolas Audebert, Catherine Herold, Kuider Slimani, and Cédric Vidal. Multimodal deep networks for text and image-based document classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 427–443. Springer, 2019. 2

[5] Galal M Binmakhashen and Sabri A Mahmoud. Document layout analysis: a comprehensive survey. *ACM Computing Surveys (CSUR)*, 52(6):1–36, 2019. 2

[6] Mohamed-Rafik Bouguelia, Yolande Belaïd, and Abdel Belaïd. A stream-based semi-supervised active learning approach for document classification. In *2013 12th International Conference on Document Analysis and Recognition*, pages 611–615. IEEE, 2013. 3

[7] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. Active learning for deep object detection. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019. 2, 3, 7

[8] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. Active and incremental learning with weak supervision. *KI - Künstliche Intelligenz*, 34(2):165–180, Jan 2020. 2

[9] Arantxa Casanova, Pedro O. Pinheiro, Negar Rostamzadeh, and Christopher J. Pal. Reinforced active learning for image segmentation. In *International Conference on Learning Representations*, 2020. 2, 3, 4

[10] Sneha Choudhary, Haritha Guttikonda, Dibyendu Roy Chowdhury, and Gerard P Learmonth. Document retrieval using deep learning. In *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6. IEEE, 2020. 2

[11] Sai Vikas Desai, Akshay Chandra Lagandula, Wei Guo, Seishi Ninomiya, and Vineeth N. Balasubramanian. An adaptive supervision framework for active learning in object detection. In *BMVC*, 2019. 2

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 1

[13] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach, 2018. 3

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 6

[15] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. 3

[16] Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 562–577, Cham, 2014. Springer International Publishing. 3

[17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 2, 3

[18] Shantanu Godbole, Abhay Harpale, Sunita Sarawagi, and Soumen Chakrabarti. Document classification through interactive supervision of document and term labels. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 185–196. Springer, 2004. 3

[19] Tobias Grüning, Gundram Leifert, Tobias Strauß, Johannes Michael, and Roger Labahn. A two-stage method for text line detection in historical documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 22(3):285–302, 2019. 2

[20] Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *NIPS*, pages 593–600. Citeseer, 2007. 1

[21] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995. IEEE, 2015. 3

[22] Manuel Haussmann, Fred Hamprecht, and Melih Kandemir. Deep active learning with adaptive acquisition. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Aug 2019. 2, 3, 5

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1, 6

[24] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Jose M. Hernández-lobato. Collaborative gaussian processes for preference learning. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2096–2104. Curran Associates, Inc., 2012. 3

[25] Rajiv Jain and Curtis Wigington. Multimodal document image classification. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 71–77. IEEE, 2019. 2

[26] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. Building a test col-

lection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666, 2006. 3

[27] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*, 2020. 3

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6, 7

[29] Ming Liu, Wray Buntine, and Gholamreza Haffari. 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, Melbourne, Australia, July 2018. Association for Computational Linguistics. 2, 3

[30] Ming Liu, Wray Buntine, and Gholamreza Haffari. Learning to actively learn neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 334–344, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. 3

[31] Z. Liu, J. Wang, S. Gong, D. Tao, and H. Lu. Deep reinforcement active learning for human-in-the-loop person re-identification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6121–6130, 2019. 2

[32] Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8):1381–1388, 2018. 2

[33] Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3060–3068, 2020. 3

[34] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013. 2, 3

[35] Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. dhsegment: A generic deep-learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 7–12. IEEE, 2018. 1, 2

[36] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari. We don't need no bounding-boxes: Training object class detectors using only human verification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 854–863, 2016. 2, 5

[37] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. Training object class detectors with click supervision. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 2

[38] Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844. IEEE, 2019. 2

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 1

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 6

[41] Soumya Roy, Asim Unmesh, and Vinay P Namboodiri. Deep active learning for object detection. In *BMVC*, page 91, 2018. 2, 3, 7

[42] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009. 1, 2, 3

[43] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001. 6

[44] Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017. 3

[45] Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. Legal document retrieval using document vector embeddings and deep learning. In *Science and information conference*, pages 160–175. Springer, 2018. 2

[46] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988. 5

[47] Dominika Tkaczyk, Pawel Szostek, and Lukasz Bolikowski. Grotoap2 - the methodology of creating a large ground truth dataset of scientific articles. *D-Lib Mag.*, 20, 2014. 3, 6

[48] Mohamed Trabelsi, Zhiyu Chen, Brian D Davison, and Jeff Heflin. Neural ranking models for document retrieval. *arXiv preprint arXiv:2102.11903*, 2021. 2

[49] D. Wang and Y. Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 112–119, 2014. 2, 3

[50] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, Dec 2017. 2

[51] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020. 1

[52] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020. 1

[53] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020. 1, 2

[54] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020. 2

[55] Bishan Yang and Tom Mitchell. Joint extraction of events and entities within a document context. *arXiv preprint arXiv:1609.03632*, 2016. 1, 2

[56] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550*, 2021. 1

[57] Donggeun Yoo and In So Kweon. Learning loss for active learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. 3

[58] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019. 3, 6, 7