Improve the Sample Efficiency of Machine for Interactive Data Annotation

Anonymous ARR submission

Abstract

To reduce human labor on manual annotations, interactive annotation leverages a model to provide annotation suggestions for the human to approve or correct. When the model is under-trained due to limited data, it tends to make wrong suggestions, requiring extra human labor to correct. To this end, we resort to analogical reasoning and propose a general sample-efficient plug-in module. This module builds analogies to historical annotated data and refines the suggestions through a dynamic weighting mechanism, thus reducing human labor. Empirical studies show the flexibility of our method in being compatible with various annotation tasks. With our method, the model, on average, saves a relative 145.08% of annotated data to reach the required accuracy. It translates to an estimated 20% less human labor compared to the original interactive annotation.

1 Introduction

Generating a high-quality and fully annotated data set by experts is criticized to be expensive. Such the data set is impossible to realise if we are operating on a fixed budget. Although the crowd-work platforms[1] provide us a cheap alternative to obtain annotations from the non-expert, the quality is hampered (Wang et al., 2022). As a compromise, only a small subset of data could be afford to be annotated by the expert (Ringger et al., 2007; Chaudhary et al., 2021). It motivates the annotation demand on limited data. In this paper, we refer it as **limited data annotation**, such as the under-documented languages (Mager et al., 2018) annotation.

To ensure the quality in the limited data annotation setting, the expert is still required to carefully encode linguistic knowledge into the data, which leads to large annotation labors (Yeung et al., 2019; Qian et al., 2022; Forbes et al., 2022). Such the large labors is believed to lower the annotation quality (Wallen et al., 2005; Lee et al., 2022). To reduce the labors, the human-machine interactive annotation methods are getting more attention (Vondrick and Ramanan, 2011; Klie et al., 2018, 2020; Le et al., 2021; Deng et al., 2021). As illustrated in Figure 1(Left), a model is introduced to provide predicted annotation suggestions to the expert on the fly. Consequently, the labors are reduced if the expert accepts the correct suggestion (i.e., *by clicking an 'OK' button*). Otherwise, extra human labors are required to correct the wrong one. At the end of each iteration, the model is updated based on the historical accepted or corrected data (we call them *annotated data*), expecting to improve prediction accuracy and lower human labors. To this end, current studies integrate the active learning (Laws et al., 2011; Klie et al., 2018; Li et al., 2021b), preferentially selecting and annotating data with potential values to improve model accuracy.

However, in the setting of limited data annotation, current methods lose the strength due to the model of low *sample-efficiency*[2] fails to learn efficiently from limited annotated data (*See examples in Figure 1*). In this case, the model annotator would be largely under-trained, even with carefully selected data by active learning methods (Müller et al., 2022), as the potential values of the selected subset data may not be fully leveraged (Tang and Huang, 2019; Mindermann et al., 2022). Consequently, the model of low sample-efficiency constantly makes mistakes and the human correction labors increase. Such the problem is important and noticed by recent work (Rietz and Maedche, 2021), but it remains unaddressed.

In this paper, we call attention to the topic of sample efficiency in the interactive annotation scenario. To improve the sample efficiency, we trace back cognitive studies (Lake et al., 2017; Castro et al., 2008; Lake et al., 2015; Mitchell, 2021),

[1] For example, the Amazon Mechanical Turk.

[2] The sample efficiency of a model refers to the number of training data required to reach a certain performance level (Dorner, 2021).

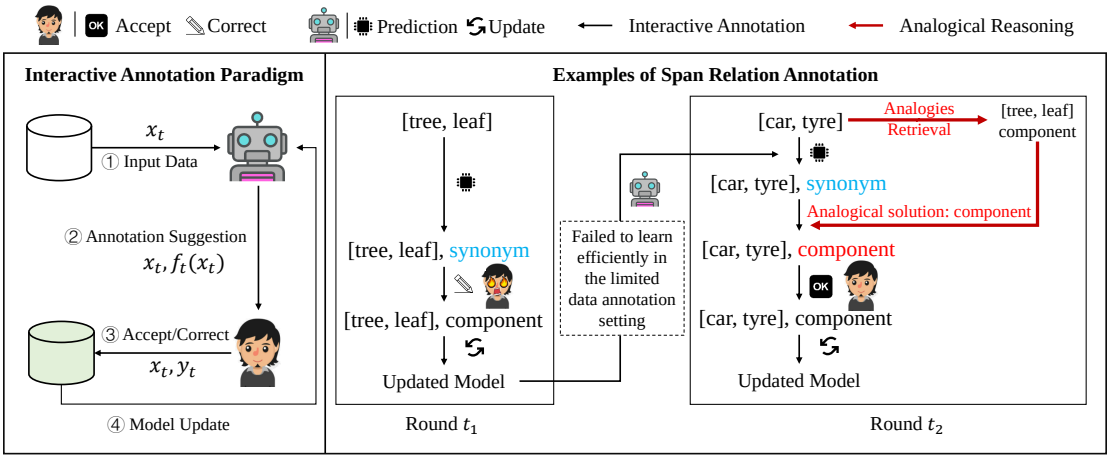


Figure 1: Framework and examples of interactive annotation. Low sample-efficiency model can not learn efficiency, leading to more model errors and human correction labors. Equipping with the analogical reasoning, model predictions are refined and corrections are saved.

finding that the human brain can learn from a few examples because our brain is continuously building analogies during the learning process of concepts to facilitate comprehension. Inspired by this, we propose a novel Analogical Reasoning for Interactive Annotation paradigm (called ARIA). In ARIA, an analogical reasoning module and a dynamic weighting mechanism are proposed. The former, mimicking the human cognition process, builds analogies from input data to historical annotated data and derives analogical solutions, while the latter automatically integrates the solutions to refine model suggestions (See Section 4.2). As such, the refined annotations are more accurate and human labors are reduced in the limited data annotation setting. In this paper, we highlight that the analogical reasoning module is a general plug-in module. This brings more flexibility, allowing for collaboration with any preferred model annotator.

Considering the cost and reproducibility of human-machine interaction, we conduct simulation experiments to evaluate the proposed paradigm, where human labors are estimated by the number of human corrections (Hwa, 2000; Kristjansson et al., 2004). We experiment on both the token-level and sentence-level annotation tasks and four commonly-used data sets. For all experiments, we simulate the scenario of limited data annotation by limiting the amount of data to annotate. The results show the flexibility of our analogical module on different model annotators and annotation tasks. To reach the required performance level, ARIR saves relative 220.36% annotated data for the sentence-level task and 69.79% for the token-level task (145.08% annotated data are saved on average). By estimation, it also saves 9.14% and 32.32% human corrections for the sentence-level and token-level tasks, respectively (20.73% are saved on average). In summary, our contributions are as follows:

- We highlight the sample efficiency problem, which is crucial but neglected in the context of interactive annotation, and we take the first step to tackle it.
- We introduce the analogical reasoning module as a model-agnostic plug-in module to improve sample efficiency. This is achieved by the dynamic weighting mechanism.
- We conduct experiments on both token-level and sentence-level tasks. The results show the flexibility of our plug-in module in combining with different model annotators and the effectiveness in improving the sample efficiency.

2 Related Work

We aim to improve sample efficiency in interactive annotation through analogical reasoning. We offer a literature review on interactive annotation and analogical reasoning. In addition, the idea of sample efficiency shares similar ideas with the general concept of few-shot learning. Thus, we also give a brief review on it and discuss the differences.

2.1 Interactive annotation

Interactive annotation incorporates a machine learning model with the human in the loop. Essentially, it leverages a model annotator to iteratively offer the expert annotation suggestions (Tratz and Phan,

2018; Klie et al., 2018; Lohr et al., 2019; Jo et al., 2020; Klie et al., 2020; Cucurnia et al., 2021; Ashktorab et al., 2021; Le et al., 2021). Towards offering accurate suggestions to reduce human labor, current studies have been carried out on the active learning (Klie et al., 2018; Laws et al., 2011; Beck et al., 2013; Li et al., 2021b) and active reinforcement learning (Casanova et al., 2020; Fang et al., 2017). Although active learning could enhance the machine performance to a certain extent, the values reflecting complex hypotheses or semantics usually require more training data for machines to learn (Kearns et al., 1994; Dasgupta, 2005; Rietz and Maedche, 2021), which is infeasible for the limited data annotation setting. In this paper, we shift the focus to sample efficiency, we are interested in the machine that learns and generalizes efficiently from very limited data, regardless of the annotation orders and how they help the learning process.

2.2 Sample efficiency

Previous evidences indicate the sample efficiency could be improved by training model with human-designed curriculum (Chevalier-Boisvert et al., 2019), safely integrating larger gradient during parameters optimization (Schulman et al., 2015), utilizing sparse attention mechanism (Spilsbury and Ilin, 2022), and injecting more supervisions, such as reward shaping (Carta et al., 2022; Mirchandani et al., 2021), data augmentations (Röder et al., 2022; Żołna et al., 2021), rich interactive advices (Watkins et al., 2021) and language descriptions (Nguyen et al., 2021). In this paper, we use analogical reasoning to make full use of historical instances and improve the sample efficiency of the model annotator, which is largely neglected in the data annotation setting (Rietz and Maedche, 2021).

2.3 Few-shot Learning

Few-shot learning, a fundamental topic in machine learning, aims at learning from limited training examples (Wang et al., 2020). Various methods can fall into its scope. For example, the fine-tuning mechanism takes advantage of the pre-trained knowledge (Chen et al., 2019; Nakamura and Harada, 2019), kernel alignment learns cross-domain representation for insufficient data (Li et al., 2021a), meta-learning realizes an optimal initialization for model parameters (Ren et al., 2018; Jamal and Qi, 2019), and metric learning explicitly builds similarities to seen training data (Wang et al., 2019; Snell et al., 2017). Technically, our analogical reasoning shares a similar idea with the metric-based methods. However, instead of utilizing a few-shot learner as the model annotator, we involve analogical reasoning as a model-agnostic plug-in module and combine it with model prediction through the dynamic weighting mechanism. As such, ARIA brings more flexibility to the interactive annotation.

3 Analogical reasoning.

Analogical reasoning targets to retrieve relevant experience to enhance the learning of the current task (Gentner and Holyoak, 1997; Carbonell, 1983). It has enhanced many tasks, such as unseen data recognition (Peyre et al., 2019) and analogical visual reasoning (Hu et al., 2021). The core procedure falls into two parts, including analogies retrieval and analogical inference.

Analogies retrieval. Building upon the assumption that the more similarities shared between two items or tasks, the stronger the analogy (Bartha, 2013), this procedure is developed to retrieve knowledge from experience that bears a strong similarity to the current task. In our settings, given a sample x_t to annotate, an analogy set A_t of analogies a_t from historical annotated data, a distance metric $d(\cdot, \cdot)$ between x_t and a_t, the retrieval module first yield a ranking of the historical analogies according to the distance to the x_t. A sorting operator π_{x_t} is further defined to sort the database analogies and it increases in distance to x_t. It outputs the orders for the analogies. Namely, $\pi_{x_t}(i)$ means the order of i-th analogies. Following those notations, we denote the retrieved analogies of x_t as $\rho_t = \{a_i | \pi_{x_t}(i) \leq k\}$, which is given by the set of the first k items w.r.t. the sorting operator. The i-th retrieved analogy $a_i = \{a_i^x, a_i^y, f_t(a_i^x)\}$ contains the instance feature a_i^x, human annotation a_i^y and machine annotation $f_t(a_i^x)$.

Analogical inference. This procedure focuses on generating exemplary solutions. According to (Bartha, 2013), no formulated acceptable rule for valid analogical inference is proposed yet, but the analogical argument can be summarized. An analogical argument is an explicit representation of a form of analogical reasoning that cites accepted similarities between two systems to support the conclusion that some further similarity exists.

4 Method

We elaborate the proposed paradigm. Section 4.1 shows the formalization of sample efficiency and

section 4.2 offers the details of ARIA.

4.1 Sample efficiency

Sample efficiency of a model refers to the number of training data needed to reach a certain performance level (Chevalier-Boisvert et al., 2019; Dorner, 2021). Following this notation, we define a metric to measure the sample efficiency for human-machine annotation tasks. Considering each model suggestion is checked by the expert, we utilize the annotation accuracy of the model as the performance level in the definition of the sample efficiency. Basically, given an unlabeled data set X of size T to annotate, we define the performance level $PL(f, t_i)$ of machine f at interaction round t_i to be its cumulative accuracy.

$$PL(f, t_i, X) = \frac{1}{t_i} \sum_{t=t_0}^{t_i} \mathbb{1}[f_t(x_t) = y_t] \quad (1)$$

, where t_o is the first interaction round, $x_t \in X$ is the instance at round t sampled from X without replacement, $\mathbb{1}[\cdot]$ is an indicator, and y_t is its ground truth annotation. Note that for the sake of robustness, we use the cumulative accuracy to be the performance metric, rather than the batch accuracy. Given a specified performance level pl on the interactive annotation task, the sample efficiency $S(f, pl, X)$ of machine f is defined as follows.

$$\begin{aligned} &S(f, pl, X) = T/t_k \\ &t_k = \inf\{t_i | PL(f, t_i, X) = pl, t_i \leq T\} \end{aligned} \quad (2)$$

When the model can not reach the given performance (e.g, the set of t_i's is empty), we set $S(f, pl, X)$ to be ∞. Essentially, $S(f, pl, X)$ captures the minimal human labors that the machine annotator requires to train to the given performance level. Specifically, a high sample efficiency is achieved if the machine annotator requires fewer human corrections and learns efficiently. We further define relative sample efficiency, conditioned on f_2, as follows.

$$RS(f_1, pl, X | f_2) = \frac{S(f_1, pl, X)}{S(f_2, pl, X)} \quad (3)$$

Basically, $RS(f_1, pl, X | f_2)$ measures the quantity of data saved by machine f_1 compared to machine f_2 when they reach the same performance level.

4.2 ARIA: the proposed method

To improve sample efficiency, two modules are proposed in ARIA, including analogical reasoning module and dynamic weighting mechanism. Given input data x_t at round t, the annotation suggestion, denoted as F_t, is obtained by $F_t(x_t) = \lambda(x_t) f_t(x_t) + (1 - \lambda(x_t)) g_t(x_t)$. Here $f_t(x_t)$, $g_t(x_t)$ and $\lambda(x_t) \in [0, 1]$ are the output of model annotator, analogical reasoning and dynamic weighting mechanism, respectively.

4.2.1 Analogical reasoning module

To perform analogical inference (See section 3), we formalize the analogical argument into a trainable aggregator function g_t. Ideally, an aggregator function should maintain a high capacity for making use of both the positive (i.e., common properties between two items or tasks) and negative effects (i.e., different properties) of different analogies (Bartha, 2013; Leclercq-Vandelannoitte and Bertin, 2018), strengthening inferred solution if the input contains more positive analogies and less negative analogies. An straightforward example of aggregator function is weighted mean aggregator as follows $g_t(x_t) = \arg\max_{y \in Y} \sum_{a \in \rho_t} w_a a^y$, which leads to a weighted KNN classifier that resorts to the finite linear combination of analogical arguments in an ensemble style, where $w_a = d(x_t, a)$ is the weight of a that down-weights those analogies with relatively more negative effects.

To strengthen the capacity of the aggregator function, we resort to Mahalanobis distance to learn w_a. Note that the Mahalanobis distance is also used to retrieve analogies. Moreover, since we regard historical annotations as analogies buffer A_t, the size of the buffer grows linearly with the times of human-machine interactions. we budget the buffer size by a class-aware strategy, where we require the human to pre-define the maximum buffer size. In this strategy, once the model receives feedback from the expert, the analogies buffer is updated by adding the newly arrived data. If the size of the analogies buffer overflows its budget, the analogy from the majority class that is most similar to its label prototype would be simply discarded first. Although it may change the hypothesis and decrease model accuracy, it turns out to be an efficient way in our experiment. To avoid overconfidence in the analogies aggregator, we also apply label smoothing (Szegedy et al., 2016) on the labels of retrieved analogies before deriving the analogical solution. We set $\alpha = 1 - \frac{1}{K}$ and K is the number of classes.

$$a^y_{LS} = a^y(1 - \alpha) + \frac{\alpha}{K} \quad (4)$$

4.2.2 Dynamic weighting mechanism

An intuitive method to integrate the model prediction and plug-in solution is the fixed weighted linear combination. However, finding out the optimal weight requires trial and error, meaning involving more human annotation hours in the parameter tuning, during the human-machine interactions. Instead, we propose a dynamic weighting mechanism λ to learn the weight for each instance automatically and encourage the model f and the analogical solution g to cooperate. At round t, the idea is to combine f_t and g_t so that F_t more resorts to f_t if its prediction is known to be safe, otherwise it safely resorts to analogies. To this end, we propose to use the analogies set A_t to examine the learning state of f_t. Formally, a instance-wise weight, denoted as $\lambda_t = \lambda_t(x_t)$, is defined to capture the reliability of $f_t(x_t)$, modeled by a neural network. Such a network takes two kinds of information as input, including the local error estimation and local density. The former is denoted as $E_t = [e_{1t}, e_{2t}, ..., e_{kt}]$, where $e_{it} = \mathbb{1}[a_i^y = f_t(a_i^x)], a_i \in \rho_t$. The latter is $D_t = [d(a_1^x, x_t), d(a_2^x, x_t), ..., d(a_k^x, x_t)]$, where $a_i \in \rho_t$. More formally, the input of the dynamic weighting mechanism is calculated as $D_t \odot E_t - D_t \odot (1 - E_t)$, where $\odot$ is an element-wise multiplication operator. By this means, such input simultaneously captures how similar x_t and retrieved analogies are and how likely the current model suggestion $f_t(x_t)$ tends to be wrong.

Technically, the most relevant method to us is the memory-based model (Khandelwal et al., 2019, 2020; Kassner and Schütze, 2020). These methods derive an additional model to enhance the raw model and combine their outputs with a fixed hyper-parameter. Those methods are infeasible for the interactive annotation setting as they require human labor to tune the hyper-parameter. We refer to Section 5.4 for more experiment analysis.

4.2.3 Loss function & Optimization

The negative log-likelihood loss $\ell(y_t, F_t(x_t))$ is used to optimize the model and the plug-in module. Also, we introduce a MSE loss $\ell_d(\mathbb{1}[y_t \neq f_t], \lambda_t)$ to train the dynamic weight mechanism λ_t, enforcing that λ_t recognizes the misclassifications of f_t. Building upon the above, we conclude the loss function for each data batch to annotate as follows.

$$\mathcal{L}(f, g, \lambda) = \sum_{i=1}^{B_t} \ell(y_i, F_t(x_i)) + \ell_d(\mathbb{1}[y_i \neq f_t(x_i)], \lambda_t) \tag{5}$$

, where B_t is the cumulative data size until round t. Notice that λ and f are trained based on the same training data. In this case, λ may overfit the performance of f on the training data, hindering λ from converging to the optimal value. One solution is to involve a valid set, but it requires additional human labor to annotate the validation set, which is impractical in our setting. Since this limitation wouldn't bother the main contributions of this paper, we'll leave it to our future work. To optimize the objective, there are two points to consider. First, operator ρ (i.e., retrieve analogies) is not differentiable, we exploit the Gumbel-softmax-based re-parameterization trick (Jang et al., 2016) to ease the optimization in our experiments. Second, the loss $\mathcal{L}$ is an bi-level optimization problems (Bard, 2013), where optimizing λ_t is nested within the optimization problems of f_t and g_t. Thus, we update f_t, g_t and λ_t separately and iteratively in a coordinate-descent style. Formally, let θ_f^k, θ_g^k and θ_λ^k be the network parameters for f^k, g^k and λ^k at the optimization iteration k, the update procedures are as follows.

$$\begin{aligned}
\theta_f^{k+1} &= \theta_f^k - \bigtriangledown_f \mathcal{L}(f, g^k, \lambda^k) \\
\theta_g^{k+1} &= \theta_g^k - \bigtriangledown_g \mathcal{L}(f^k, g, \lambda^k) \\
\theta_\lambda^{k+1} &= \theta_\lambda^k - \bigtriangledown_\lambda \mathcal{L}(f^{k+1}, g^{k+1}, \lambda)
\end{aligned} \tag{6}$$

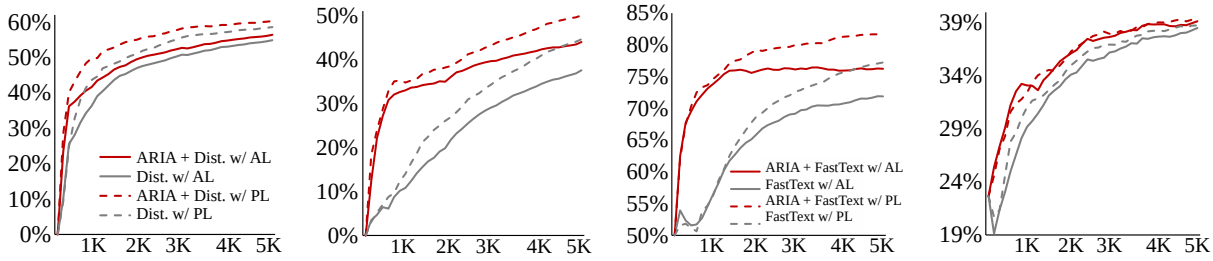
5 Experiments

To improve sample efficiency, ARIA automatically refines the annotation suggestions from any preferred model annotator. As such, higher sample efficiency can improves model accuracy on the limited data and lowers human labors. In this section, extensive experiments are conducted to evaluate the effectiveness of ARIA in terms of the human correction labor and sample efficiency. Specifically, we are interested in the following questions.

- **Q1**: How much human correction labors can ARIA reduce compared to the conventional interactive annotation?
- **Q2**: How many gains does ARIA bring with regard to the sample efficiency?
- **Q3**: Without human labors on tuning parameters, does the dynamic weighting mechanism achieve promising results?

5.1 Experiment setup

Considering the cost and reproducibility of human-computer interaction, we report the results of simulation experiments and human evaluations. We



(a) KGC - WN18RR (b) KGC - FreeBase (c) SA - IMDB (d) SA - SST

Figure 2: Illustration of annotation performance on various data sets. The X-axis and Y-axis represent the number of data to annotate and the Machine Cumulative Accuracy, respectively.

focus on the annotation task in the token-level and sentence-level settings, including knowledge graph completion and sentiment analysis.

Dataset. For simplicity, we use the non-contextual pre-trained GloVe (Pennington et al., 2014) for word embedding and sentence embedding without fine-tuning. The contextual one (e.g., BERT embedding) is not used, because the token-level task in our setting has no context information. Following previous work (Dou et al., 2019), we simulate the interactive annotation process for the *limited data annotation* setting. We place restrictions on the size of the data set, ranging from $1K$ to $5K$ with the step size being $1K$ (i.e., 1000).

- Knowledge graph completion (KGC). We focus on annotating the semantic class of the input word pair. Here, two benchmark knowledge graph data sets are used in our experiments, including WN18RR[3] and Freebase[4].

- Sentiment analysis (SA). Two benchmark data sets are used in our experiments, including SST[5], and IMDB[6]. In this case, the human-machine team is required to annotate the sentiment label for the input data.

Comparative Methods. Considering the requirements of the annotation system for the time efficiency of annotation, we use a lightweight model for experiments. The classic distributional model (Roller et al., 2014; Kober et al., 2021) and FastText (Joulin et al., 2017) are utilized in token-level and sentence-level annotation tasks, respectively. Considering previous works on interactive annotation focus on active learning, to comprehensively evaluate the sample efficiency enhancement, all methods are equipped with uncertainty-based active learning (Ren et al., 2020), denoted as Dist. w/ AL and FastText w/ AL, and passive learning (i.e., random strategy), denoted as Dist. w/ PL and FastText w/ PL. Methods without analogical reasoning enhancement play the role of baselines. Furthermore, we analyze the effectiveness of the dynamic weighting mechanism and compare it with the fixed weighting strategy, used in the memory-based model (Khandelwal et al., 2019; Kassner and Schütze, 2020; Khandelwal et al., 2020).

Evaluation Metric. Interactive annotation assumes human annotation is the ground truth. We only evaluate the annotation performance of the machine, denoted as the *Machine Cumulative Accuracy* (MCA), and the relative sample efficiency conditioned on the baselines (See Section 4). Note the values of relative sample efficiency of baselines are 100%. Also, considering that machine errors involve extra human labors to correct, the overall human labors could be measured by the aforementioned Machine Cumulative Accuracy. Here, we use the *Exact Match* between model and human annotations to calculate the accuracy, as data should be annotated in deterministic labels, not the probabilistic distributions.

Implementation details. All experiments are carried out on a machine with Intel(R) Core(TM) i5-12400F, 16GB memory, and GeForce RTX 3060. For simplicity, the model annotator and the dynamic weighting mechanism are both implemented by a three-layer FC with ReLu activation and dropout. As for the analogical reasoning module, we set $k = 20$ for ρ_t, and the size of analogies buffer A_t is set as 1000 for all data sets. Also, we apply the label smoothing trick according to Eq.(4), so that ARIA could avoid over-confidence in analogical solutions. To simulate the interactions with the human expert, we mask out the ground truth

[3]https://paperswithcode.com/dataset/wn18rr

[4]https://www.microsoft.com/en-us/download/details.aspx?id=52312

[5]https://nlp.stanford.edu/sentiment/code.html

[6]https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Table 1: Machine Cumulative Accuracy gains of ARIA over conventional interactive annotation paradigm.

Token-level Annotation (%)	**WN18RR**	**FreeBase**
ARIA + Dist. w/ PL	5.93	48.90
ARIA + Dist. w/ AL	5.24	69.21
Sentence-level Annotation (%)	**IMDB**	**SST**
ARIA + FastText w/ PL	14.19	2.93
ARIA + FastText w/ AL	13.83	5.6

Table 2: Relative sample efficiency gains of ARIA over conventional interactive annotation paradigm.

Token-level Annotation (%)	**WN18RR**	**FreeBase**
ARIA + Dist. w/ PL	45.94	79.85
ARIA + Dist. w/ AL	31.73	121.64
Sentence-level Annotation (%)	**IMDB**	**SST**
ARIA + FastText w/ PL	217.26	29.50
ARIA + FastText w/ AL	591.47	43.22

labels until the model made predictions.

5.2 Evaluation on labor reduction (Q1)

Machine annotation suggestions should be accurate so that the human labor to correct annotations is saved. Therefore, the labor of human correcting annotations goes linearly with the number of machine misclassifications during the interactions. To evaluate the labor reduction in the limited data annotation setting, Figure 2 illustrates the curves of the Machine Cumulative Accuracy (MCA) of different annotation methods. Table 1 further provides the relative MCA gains of ARIA, averaged over the different amounts of data to annotate. There are two main conclusions.

First, ARIA, equipped with analogical reasoning, outperforms all baselines on both token-level and sentence-level tasks and improves the qualities of machine annotations. By estimation, ARIA, on average, saves 9.14% and 32.32% human correction labors for sentiment analysis and knowledge graph completion tasks, respectively. We further estimate that the proposed method saves 20.73% human labor to correct model suggestions over all data sets. Moreover, such advantages hold along with the increasing data size (See solid and dotted lines marked in red in Figure 2), which indicates that our analogical plug-in module generally works well on different annotation tasks and data with different sizes. For a better understanding, the representative results on $1K$ data are provided in Table 3. In this simulation, users are extremely budgeted and only afford to annotate very limited data. In this case, ARIA saves 20.31% (31.71% on IMDB and 8.90% on SST) and 93.67% (11.42% on WN18RR and 175.91% on FreeBase) human correction labor on SA and KGC tasks, respectively. Compared to Table 1, we argue ARIA brings larger performance gains if the data are very limited.

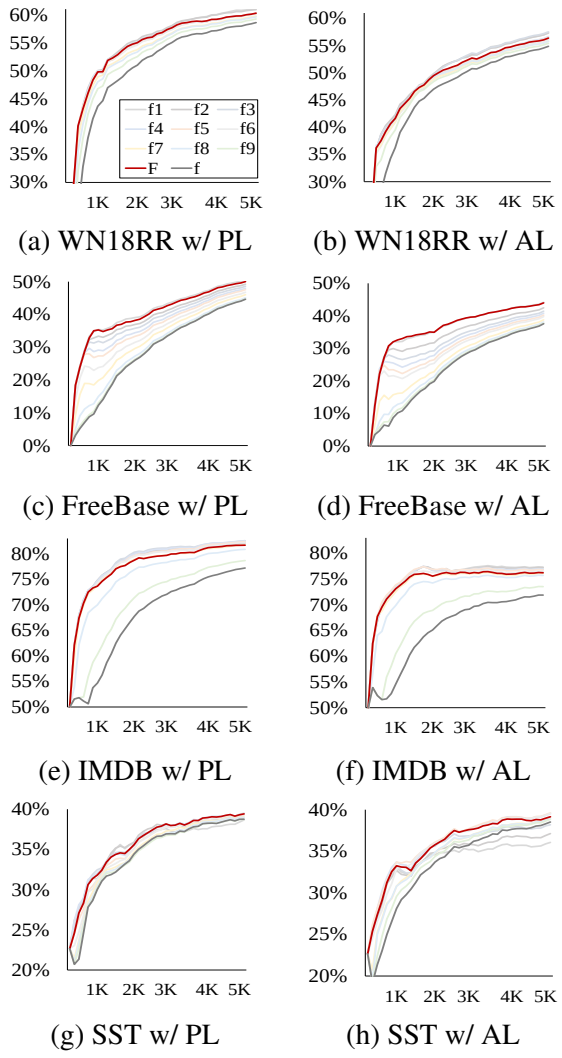
Second, we notice that active learning may play a negative role during the interactive annotation, leading to the worse performance on WN18RR, FreeBase, and SST data sets compared to the passive learning. One possible reason lies in the fact that representative or informative data, selected by active learning, may be hard for machine (Tang and Huang, 2019) and require more training data to learn (Kearns et al., 1994; Dasgupta, 2005; Rietz and Maedche, 2021). In the case of limited data annotation setting, those valuable data from active learning strategy thus lose advantages compared to ones from passive learning (Pezeshkpour et al., 2020). This further supports the necessity of building analogies during the human-machine interaction in the limited data annotation setting.

Table 3: Representative comparison for Model Cumulative Accuracy (denoted as MCA) and Relative Sample Efficiency (denoted as RSE) under $1K$ data.

Methods/Tasks	**MCA ↑ (%)**		**RSE ↑ (%)**	
Token-level Annotation	WN18RR	FreeBase	WN18RR	FreeBase
Dist. w/ PL	44.63	14.06	100	100
Dist. w/ AL	39.06	10.84	100	100
ARIA + Dist. w/ PL	**49.80**	**34.77**	**162.5**	**287.5**
ARIA + Dist. w/ AL	**43.46**	**33.01**	**137.5**	**375.0**
Sentence-level Annotation	IMDB	SST	IMDB	SST
FastText w/ PL	56.25	31.05	100	100
FastText w/ AL	56.45	29.20	100	100
ARIA + FastText w/ PL	**74.61**	**32.42**	**362.5**	**137.5**
ARIA + FastText w/ AL	**73.83**	**33.11**	**912.5**	**175.0**

5.3 Evaluation on sample efficiency (Q2)

Table 2 and Table 3 demonstrate the results in terms of relative sample efficiency conditioned on the baseline to ease comparison, where the target performance levels pl are set to be the MCAs of ARIA under different data size. In essence, ARIA enjoys much high sample efficiency, where the information of each data is used to not only train the machine but also regarded as analogies, which refines the machine annotation suggestions. More statistically speaking, the average values of relative sample efficiency gains over baseline under different data sizes are reported in Table 1. By estimation, it also saves 9.14% and 32.32% human labors for sentiment analysis and knowledge



(a) WN18RR w/ PL (b) WN18RR w/ AL

(c) FreeBase w/ PL (d) FreeBase w/ AL

(e) IMDB w/ PL (f) IMDB w/ AL

(g) SST w/ PL (h) SST w/ AL

Figure 3: Machine Cumulative Accuracy of ARIA with different weighting strategies. To save space, we denote ARIA and baseline as F and f, respectively. $f1$ means ARIA with the fixed weight being 0.1, and so on.

graph completion tasks respectively, to finish annotation. On average, we estimate that the proposed method enjoys 145.08% gains over all data sets, indicating that analogical reasoning help save relative 145.08% human annotations for the machine to reach the required performance level. As shown in the MCA curve in Figure2, the learning efficiency of the model annotator on FreeBase and IMDB is relatively lower, especially in the early stage of the learning process. This means that the model requires more data to train to reach the same performance level as ARIA. It explains the reason why ARIA largely outperforms baselines on FreeBase and IMDB on FreeBase and IMDB in terms of relative sample efficiency. In our opinion, the improved sample efficiency contributes to the MCA gain of ARIA, allowing to learn and generalize efficiently from only a few data.

5.4 Analysis on dynamic weighting (Q3)

Our dynamic weighting mechanism adjusts λ automatically without human tuning. In this section, we consider the ARIA with a fixed weighting method (See section 4.2.2) for comparison. Here, the the λ (i.e., the weight) is tuned from 0.1 to 0.9 with step size being 0.1. As illustrated in Figure3, our results are in line with previous studies, stating that different tasks with different training data have a different optimal value of λ. According to our tuning experiments, the optimal λ is in $\{0.1, 0.3, 0.5\}$ for different data sets, which are largely different from the previous studies. Primarily, $\lambda = 0.7$ is suggested for QA (Kassner and Schütze, 2020) and $\lambda \in \{0.2, 0.3, 0.8\}$ for machine translation (Khandelwal et al., 2020), it also takes different values ($\lambda \in \{0.2, 0.75, 0.9\}$) on different training data when building language model (Khandelwal et al., 2019). Therefore, the fixed weighting methods are infeasible for the interactive annotation, as they take some trial and error to tune λ accordingly, hence involving more human labor. On the contrary, by treating λ as a trainable parameter, our dynamic weighting mechanism reaches the sub-optimal performance (see section 4.2.3 for explanation). We argue it is the trade-off between annotation performance and human labor. When human labor are budgeted, our dynamic weighting mechanism is a better choice.

6 Conclusion

We call attention to the sample efficiency in the limited data annotation setting. To this end, we propose ARIA and highlight the model-agnostic plug-in module and the dynamic weighting mechanism. They explore a new solution to improve sample efficiency and bring more flexibility in allowing the expert to design any preferred model annotator according to different annotation tasks.

We are devoted to optimizing human-machine utilities by emphasizing the learning of task-specified concepts efficiently from a few human demonstrations. To achieve this long-term goal, we start from the basic idea of sample efficiency. However, there is a loose ending to our discussion. In the future, we would extend our research scope by involving more proactive instructions from the expert, such as machine teaching methods.

References

Zahra Ashktorab, Michael Desmond, Josh Andres, Michael Muller, Narendra Nath Joshi, Michelle Brachman, Aabhas Sharma, Kristina Brimijoin, Qian Pan, Christine T Wolf, et al. 2021. Ai-assisted human labeling: Batching for efficiency without overreliance. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–27.

Jonathan F Bard. 2013. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media.

Paul Bartha. 2013. Analogy and analogical reasoning.

Daniel Beck, Lucia Specia, and Trevor Cohn. 2013. Reducing annotation effort for quality estimation via active learning. In *ACL*, pages 543–548, Sofia, Bulgaria. ACL.

Jaime G Carbonell. 1983. Learning by analogy: Formulating and generalizing plans from past experience. In *Machine learning*, pages 137–161. Springer.

Thomas Carta, Sylvain Lamprier, Pierre-Yves Oudeyer, and Olivier Sigaud. 2022. Eager: Asking and answering questions for automatic reward shaping in language-guided rl. *arXiv preprint arXiv:2206.09674*.

Arantxa Casanova, Pedro O Pinheiro, Negar Rostamzadeh, and Christopher J Pal. 2020. Reinforced active learning for image segmentation. *arXiv preprint arXiv:2002.06583*.

Rui Castro, Charles Kalish, Robert Nowak, Ruichen Qian, Tim Rogers, and Jerry Zhu. 2008. Human active learning. *Advances in neural information processing systems*, 21.

Aditi Chaudhary, Antonios Anastasopoulos, Zaid Sheikh, and Graham Neubig. 2021. Reducing confusion in active learning for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 9:1–16.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *International Conference on Learning Representations*.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2019. Babyai: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations*.

Davide Cucurnia, Nikolai Rozanov, Irene Sucameli, Augusto Ciuffoletti, and Maria Simi. 2021. Matilda-multi-annotator multi-language interactivelight-weight dialogue annotator. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 32–39.

Sanjoy Dasgupta. 2005. Coarse sample complexity bounds for active learning. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pages 235–242.

Dazhen Deng, Jiang Wu, Jiachen Wang, Yihong Wu, Xiao Xie, Zheng Zhou, Hui Zhang, Xiaolong Zhang, and Yingcai Wu. 2021. Eventanchor: reducing human interactions in event annotation of racket sports videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Florian E Dorner. 2021. Measuring progress in deep reinforcement learning sample efficiency. *arXiv preprint arXiv:2102.04881*.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1192–1197.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *EMNLP*, pages 595–605.

Clarissa Forbes, Farhan Samir, Bruce Oliver, Changbing Yang, Edith Coates, Garrett Nicolai, and Miikka Silfverberg. 2022. Dim wihl gat tun: The case for linguistic expertise in nlp for under-documented languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2116–2130.

Dedre Gentner and Keith J Holyoak. 1997. Reasoning and learning by analogy: Introduction. *American psychologist*, 52(1):32.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. 2021. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1567–1574.

Rebecca Hwa. 2000. Sample selection for statistical grammar induction. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 45–52.

Muhammad Abdullah Jamal and Guo-Jun Qi. 2019. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11719–11727.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Yohan Jo, Elijah Mayfield, Chris Reed, and Eduard Hovy. 2020. Machine-aided annotation for fine-grained proposition types in argumentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1008–1018, Marseille, France. European Language Resources Association.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Bert-knn: Adding a knn search component to pretrained language models for better qa. In *EMNLP: Findings*, pages 3424–3430.

Michael J Kearns, Umesh Virkumar Vazirani, and Umesh Vazirani. 1994. *An introduction to computational learning theory*. MIT press.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. *arXiv preprint arXiv:2010.00710*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *ICLR*.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *COLING: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. ACL.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2020. From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains. In *ACL*, pages 6982–6993, Online. ACL.

Thomas Kober, Julie Weeds, Lorenzo Bertolini, and David J. Weir. 2021. Data augmentation for hypernymy detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1034–1048.

Trausti Kristjansson, Aron Culotta, Paul Viola, and Andrew McCallum. 2004. Interactive information extraction with constrained conditional random fields. In *AAAI*, volume 4, pages 412–418.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Florian Laws, Christian Scheible, and Hinrich Schütze. 2011. Active learning with Amazon Mechanical Turk. In *EMNLP*, pages 1546–1556, Edinburgh, Scotland, UK. ACL.

Trung-Nghia Le, Tam V Nguyen, Quoc-Cuong Tran, Lam Nguyen, Trung-Hieu Hoang, Minh-Quan Le, and Minh-Triet Tran. 2021. Interactive video object mask annotation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16067–16070.

Aurélie Leclercq-Vandelannoitte and Emmanuel Bertin. 2018. From sovereign it governance to liberal it governmentality? a foucauldian analogy. *European Journal of Information Systems*, 27(3):326–346.

Ji-Ung Lee, Jan-Christoph Klie, and Iryna Gurevych. 2022. Annotation curricula to implicitly train non-expert annotators. *Computational Linguistics*, 48(2):343–373.

Wei-Hong Li, Xialei Liu, and Hakan Bilen. 2021a. Universal representation learning from multiple domains for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9526–9535.

Yanzeng Li, Bowen Yu, Li Quangang, and Tingwen Liu. 2021b. Fitannotator: A flexible and intelligent text annotation system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 35–41.

Christina Lohr, Johannes Kiesel, Stephanie Luther, Johannes Hellrich, Tobias Kolditz, Benno Stein, and Udo Hahn. 2019. Continuous quality control and advanced text segment annotation with WAT-SL 2.0. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 215–219, Florence, Italy. ACL.

Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. 2018. Challenges of language technologies for the indigenous languages of the americas. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 55–69.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR.

Suvir Mirchandani, Siddharth Karamcheti, and Dorsa Sadigh. 2021. Ella: Exploration through learned language abstraction. *Advances in Neural Information Processing Systems*, 34:29529–29540.

Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101.

Thomas Müller, Guillermo Pérez-Torró, Angelo Basile, and Marc Franco-Salvador. 2022. Active few-shot learning with fasl. *arXiv preprint arXiv:2204.09347*.

Akihiro Nakamura and Tatsuya Harada. 2019. Revisiting fine-tuning for few-shot learning. *arXiv preprint arXiv:1910.00216*.

Khanh X Nguyen, Dipendra Misra, Robert Schapire, Miroslav Dudík, and Patrick Shafto. 2021. Interactive learning from activity description. In *International Conference on Machine Learning*, pages 8096–8108. PMLR.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. 2019. Detecting unseen visual relations using analogies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1981–1990.

Pouya Pezeshkpour, Zhengli Zhao, and Sameer Singh. 2020. On the utility of active instance selection for few-shot learning. *NeurIPS HAMLETS*.

Jing Qian, Qi Sun, Curtis Wigington, Han L Han, Tong Sun, Jennifer Healey, James Tompkin, and Jeff Huang. 2022. Dually noted: Layout-aware annotations with smartphone augmented reality. In *CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2018. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and X. Wang. 2020. A survey of deep active learning. *ArXiv*, abs/2009.00236.

Tim Rietz and Alexander Maedche. 2021. Cody: An ai-based system to semi-automate coding for qualitative research. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Eric Ringger, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, Kevin Seppi, and Deryle Lonsdale. 2007. Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 101–108.

Frank Röder, Manfred Eppe, and Stefan Wermter. 2022. Grounding hindsight instructions in multi-goal reinforcement learning for robotics. *arXiv preprint arXiv:2204.04308*.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING: Technical Papers*, pages 1025–1036.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Sam Spilsbury and Alexander Ilin. 2022. Compositional generalization in grounded language learning via induced model sparsity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 143–155.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Ying-Peng Tang and Sheng-Jun Huang. 2019. Self-paced active learning: Query the right thing at the right time. In *AAAI*, volume 33, pages 5117–5124.

Stephen Tratz and Nhien Phan. 2018. A web-based system for crowd-in-the-loop dependency treebanking. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Carl Vondrick and Deva Ramanan. 2011. Video annotation and tracking with active learning. *Advances in Neural Information Processing Systems*, 24.

Erik Wallen, Jan L Plass, and Roland Brünken. 2005. The function of annotations in the comprehension of scientific texts: Cognitive load effects and the impact of verbal ability. *Educational Technology Research and Development*, 53(3):59–71.

Ding Wang, Shantanu Prabhat, and Nithya Sambasivan. 2022. Whose ai dream? in search of the aspiration in data annotation. In *CHI Conference on Human Factors in Computing Systems*, pages 1–16.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. 2019. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.

Olivia Watkins, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Jacob Andreas. 2021. Teachable reinforcement learning via advice distillation. *Advances in Neural Information Processing Systems*, 34:6920–6933.

Serena Yeung, Francesca Rinaldo, Jeffrey Jopling, Bingbin Liu, Rishab Mehra, N Lance Downing, Michelle Guo, Gabriel M Bianconi, Alexandre Alahi, Julia Lee, et al. 2019. A computer vision system for deep learning-based detection of patient mobilization activities in the icu. *NPJ digital medicine*, 2(1):1–5.

Konrad Żołna, Chitwan Saharia, Leonard Boussioux, David Yu-Tung Hui, Maxime Chevalier-Boisvert, Dzmitry Bahdanau, and Yoshua Bengio. 2021. Combating false negatives in adversarial imitation learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE.