Teaching Humans When To Defer to a Classifier via Exemplars

Hussein Mozannar [*] Arvind Satyanarayan [†] David Sontag [‡]

Abstract

Expert decision makers are starting to rely on data-driven automated agents to assist them with various tasks. For this collaboration to perform properly, the human decision maker must have a mental model of when and when not to rely on the agent. In this work, we aim to ensure that human decision makers learn a valid mental model of the agent's strengths and weaknesses. To accomplish this goal, we propose an exemplar-based teaching strategy where humans solve a set of selected examples and with our help generalize from them to the domain. We present a novel parameterization of the human's mental model of the AI that applies a nearest neighbor rule in local regions surrounding the teaching examples. Using this model, we derive a near-optimal strategy for selecting a representative teaching set. We validate the benefits of our teaching strategy on a multi-hop question answering task with an interpretable AI model using crowd workers. We find that when workers draw the right lessons from the teaching stage, their task performance improves. We furthermore validate our method on a set of synthetic experiments.

1 Introduction

Automated agents powered by machine learning are augmenting the capabilities of human decision makers in settings such as healthcare [BBH$^+$20, GSR$^+$21], content moderation [LHL16] and more routine decisions such as asking AI-enabled virtual assistants for recommendations [SC19]. This mode of interaction whereby the automated agent serves only to provide a recommendation to the human decision maker, a setting typically named *AI assisted decision making*, is the focus of our study here. A key question is how does the human expert know when to rely on the AI for advice. In this work, we make the case for the need to initially onboard the human decision maker on when and when not to rely on the automated agent. We propose that before an AI agent is deployed to assist a human decision maker, the human is taught through a tailored onboarding phase how to make decisions with the help of the AI. The purpose of the onboarding is to help the human understand when to trust the AI and how the AI can complement their abilities. This allows the human to have an accurate mental model of the AI agent, and this mental model helps in setting expectations about the performance of the AI on different examples.

Our onboarding phase consists of letting the human predict on a series of specially selected teaching examples in a setting that mimics the deployment use case. The examples are chosen to

[*] Massachusetts Institute of Technology. Email: `mozannar@mit.edu`

[†] Massachusetts Institute of Technology. Email: `arvindsatya@mit.edu`

[‡] Massachusetts Institute of Technology. Email: `dsontag@csail.mit.edu`

give an overview of the AI's strengths and weaknesses especially when it complement's the abilities of the human. After predicting on each example, the human agent then receives feedback on their performance and that of the AI. To allow the human to generalize from each example, we display features of the region surrounding the example. Finally, to enable retention of the example, we let the human write down a lesson indicating whether they should trust the AI in that region and what characterizes the region. Our approach is inspired by research in the education literature that highlight the importance of feedback and lesson retention for learning [ADRW00, HT07].

To select the teaching examples, we need to have a mathematical framework of how the human mental model evolves after we give them feedback. We model the human thought process as first deciding whether to rely on the AI's prediction or not using an internal *rejector* in section 3. This rejector is what we refer to as the human's mental model of the AI. We propose to model the human's rejector as consisting of a prior rejector and a nearest neighbor rule that only applies in local regions surrounding each teaching example in section 4. This novel parameterization is inspired by work in cognitive science that suggests that humans make decisions by weighing similar past experiences [BKSD17]. Assuming this rejector model, we give a near-optimal greedy strategy for selecting a set of representative teaching examples that allows us to control the examples and the region surrounding them in section 5.

We first evaluate the efficacy of our algorithmic approach on a set of synthetic experiments and its robustness to the misspecification of the human's model. For our main evaluation, we conduct experiments on Amazon Mechanical Turk on the task of passage-based question answering from HotpotQA [YQZ+18] in section 6. Crowdworkers first performed a teaching phase and were then tested on a randomly chosen subset of examples. Our results demonstrate the importance of teaching: around half of the participants who undertook the teaching phase were able to correctly determine the AI's region of error and had a resulting improved performance.

2 Related Work

One of the goals of explainable machine learning is to enable humans to better evaluate the correctness of the AI's prediction by providing supporting evidence [LT19, LCH+19, SRFB+20, HB20, ZLB20, KAB19, SLL20, SLGS21, WVW21, GBF+20]. However, these explanations do not inform the decision maker how to weigh their own predictions against those of the AI or how to combine the AI's evidence to make their final decision [KNJ+20]. The AI explanations cannot factor in the effect of the human's side information, and thus the human has to learn what their side information reveals about the performance of the AI or themselves. Moreover, if the AI's explanations are unfaithful or become so due to a distribution shift in the data [DT18], then the human may then over-weigh the AI's abilities. Another direct approach for teaching is presenting the human with a set of guidelines of when to rely on the AI [AWV+19]. However, these guidelines need to be developed by a set of domain experts and no standard approach currently exists for creating such guidelines. As a byproduct of our teaching approach, each human writes a set of unorganized rules that can then be more easily turned into such guidelines.

The reverse setting, of teaching a classifier when to defer to a human, is dubbed as learning to defer (LTD) [MPZ18, RBC+19, MS20, WHK20]. The main goal of LTD is to learn a rejector that determines which of the AI and the human should predict on each example. However, there are numerous legal and accountability constraints that may prohibit a machine from making final decisions in high stakes scenarios. Additionally, the actual test-time setting may differ from that which was used during training, but since in our setting the human makes the final decision, this allows them to adapt their decision making and detect any unexpected model errors. As an example in a clinical use case, factors such as times of substantially increased patient load may affect the human expert's accuracy. The human may also occasionally have side-information that was unavailable to the AI that could improve their decision making. Compared to LTD, deployment may be simplified because the same AI is used for all experts; as new experts arrive, our onboarding phase trains them to use the AI according to their unique abilities. Our teaching setting and LTD also use very different techniques. Although the objective that we present in Equation (2) is closely related to the objective used by [MS20], the main task in our setting is that of teaching the human when to defer. This requires us to develop a formalization of the human mental model and algorithms for selecting a subset of examples that enables accurate learning.

Related work has explored how to best onboard a human to trust or replicate a model's prediction. LIME, a black-box feature importance method, was used to select examples so that crowdworkers could evaluate which of two models would perform better [RSG16, LLT20]. Their selection strategy does not take into account the human predictor, nor does their approach do more than display the examples. On a task of visual question answering, [CPY+18] handpicked 7 examples to teach crowdworkers about the AI abilities and found that teaching improved the ability to detect the AI's failure. [FBG19] on a Quizbowl question answering task highlight the importance of modeling the skill level of the human expert when designing the explanations; this further motivates our incorporation of the human predictor into the choice of the teaching set. Through a study of 21 pathologists, [CWS+19] gathered a set of guidelines of what clinicians wanted to know about an AI prior to interacting with it. [YWVW19] study the effect of initial debriefing of stated AI accuracy compared to observed AI accuracy in deployment and find a significant effect of stated accuracy on trust, but that diminishes quickly after observing the model in practice; this reinforces our approach of building trust through examples that simulate deployment. [BNK+19] investigate the role of the human's mental model of the AI on task accuracy, however, the mental model is formed through test time interaction rather than through an onboarding stage. [BNK+21] propose a theoretical model for AI-assisted decision making, assuming that the human has a perfect mental model of the AI and that the human has uniform error.

Finally, our work was inspired by the literature on machine teaching [ZSZR18, SBB+14, KZSC21, HCMA+18, DHPZ19] and curriculum learning [BLCW09, GBM+17]. Our work differentiates itself from the machine teaching literature by the use of our novel radius neighbor human model and the goal of teaching how to defer to an AI rather than teaching concepts to humans. Studies have also explored the use of reinforcement learning as a tool for online education [RHY+20, DKB19, LRJ+15]. We further expand the related work in Appendix A.

3 Problem Setup

Our formalization is based on the interaction between two agents: the AI, an automated agent, and a human expert who both collaborate to predict a target $Y \in \mathcal{Y}$ based on a given input context. The setup is as follows: the AI perceives a view of the input $X \in \mathcal{X}$, then communicates a message $A \in \mathcal{A}$ that is perceived by the human. The human expert then integrates the AI message A and their own view of the input $Z \in \mathcal{Z}$ to make a final decision $M(Z, A)$ which can either be to predict on their own or allow the AI agent to predict. The input space of the human Z and that of the AI X could be different since the human may have side information that the AI can't observe. This is

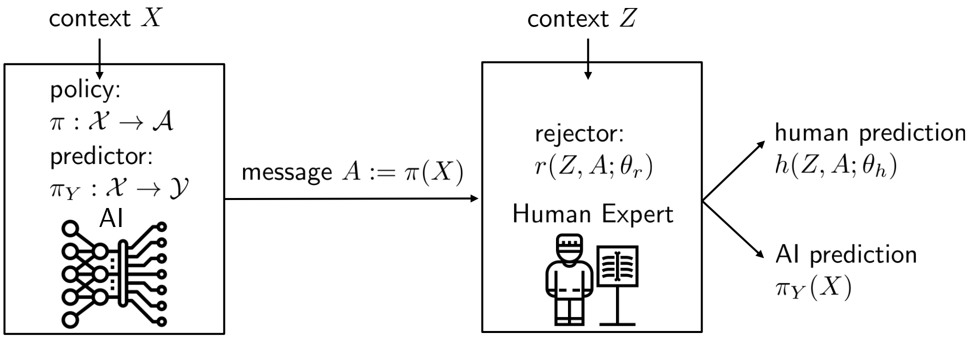


Figure 1: The AI assisted decision making pipeline. The AI first sends to the human a message A, then the human decides with their rejector $r(Z, A)$ if they should follow the AI's advice and predict $\pi_Y(X)$ or they should predict on their own using $h(Z, A)$.

essentially the *AI-Assisted Decision Making* setup illustrated in Figure 1 which is the more common mode of interaction between humans and artificially intelligent agents in high-stakes scenarios.

More formally, the AI consists of a predictor $\pi_Y : \mathcal{X} \to \mathcal{Y}$ that can solve the task on its own and a policy $\pi : \mathcal{X} \to \mathcal{A}$ which serves to communicate with the human. The message space $\mathcal{A}$ may consist for example of the AI's prediction $\pi_Y(X)$ alongside an explanation of their decision. On the other hand, the human when seeing the AI's message consists of a **predictor** $h : \mathcal{Z} \times \mathcal{A} \to \mathcal{Y}$ parameterized by θ_h and the human decides to allow the AI to predict or not according to a **rejector** $r : \mathcal{Z} \times \mathcal{A} \to \{0, 1\}$ parameterized by θ_r, where if $r(Z, A; \theta_r) = 1$ the human uses the AI's answer for its final prediction. This implies that the final human decision M is as follows:

$$M(Z, A) = \begin{cases} \pi_Y(x) & , \text{ if } r(Z, A; \theta_r) = 1 \\ h(Z, A; \theta_h) & , \text{ otherwise} \end{cases} \tag{1}$$

System objective. Given the above ingredients and a performance measure on the label space $l(y, \hat{y}) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ (e.g. 0-1 loss), the loss that we incur is the following:

$$L(\pi, \pi_Y, h, r) = \mathbb{E}_{x,z,y} [\underbrace{l(\pi_Y(x), y)}_{\text{AI cost}} \overbrace{\mathbb{I}_{r(x,\pi(x))=1}}^{\text{AI predicts}} + \underbrace{l(h(z, \pi(x)), y)}_{\text{Human cost}} \overbrace{\mathbb{I}_{r(x,\pi(x))=0}}^{\text{Human predicts}}] \tag{2}$$

We put ourselves in the role of a system designer who has knowledge of both the human and the AI and wishes to minimize the loss of the system L (2).

The central Human-AI interaction problem. Given a fixed AI policy, and human parameters (θ_h, θ_r), the manner in which the human expert integrates the AI's message depends only on the expert context Z and the message itself A. In particular, for two different policies π_1 and π_2 that output the same message A on input Z, our framework tells us that the resulting behavior of the human expert would be identical in both cases. However, if it is known to the human that AI π_1 has very high error compared to AI π_2, then is more likely for them to trust the message if it is coming from π_2 rather than from π_1. Thus it is more realistic to assume that the expert has a *mental model* of the policy π that they have arrived at from either a description of the policy or from previously interacting with it; the rejector here formalizes the *mental model.* This insight forces us to now consider the parameters (θ_h, θ_r) as variables that are learned by the human as a function of the underlying AI policy π. This makes the optimization of the loss now much more challenging as whenever the policy π changes, the human's mental model, (θ_h, θ_r), needs to update. Therefore, we need to first understand how the human's mental model evolves and how we can influence it.

Teaching Humans about the AI. In this work, we focus on exemplar based strategies to allow the human to update their mental models of the AI. The question is then how do we select a minimal set of examples that teaches the human an accurate mental model of the AI. To make progress, we need to first understand the form of the human's rejector and how it evolves, which we elaborate on in the following section. Crucially, we will keep the AI in this work as a fixed policy and not look to optimize for it. Once we understand this first step, future work can then look to close the loop which entails learning an updated AI with the knowledge of the human learner dynamics.

4 Human Mental Model

We now introduce our model of the human's rejector and the elements of the teaching setup. The tasks we are interested in are where humans are *domain experts*, where we define domain experts to mean that their knowledge about the task and their predictive performance are fixed. We further extend this to how they may incorporate the AI message in their prediction, but crucially not how they decide when to use the AI. This assumption translates in our formulation as follows.

Assumption 1. *The human predictor does not vary as they interact with the AI, i.e. we assume θ_h to be fixed.*

While we have assumed θ_h is fixed and have so far spoken about a singular human, in reality, the AI might be deployed in conjunction with multiple human experts. These experts might have different parameters θ_h individually, however; for the rest of this paper, we focus on a singular expert that we are interacting with.

We now move our attention to the human's rejector, which represents their mental model of the AI, and learned after observing a series of labeled examples. Research on human learning from the cognitive science literature has postulated that for complex tasks humans make decisions by sampling similar experiences from memory [BKSD17, GL13, RP14]. Moreover, [BKSD17] makes the explicit comparison with nearest neighbor models found in machine learning. However, standard nearest neighbor models don't allow for prior knowledge to be incorporated. For this reason, we postulate a nearest neighbor model for the human rejector that starts with a prior and updates in local regions of each shown example in the following assumption.

Assumption 2 (Form of Human's rejector)**.** *The human's rejector consists of a prior rejector rule and a nearest neighbor rule learned after observing teaching examples $D_T = \{z_i, a_i, r_i\}_{i=1}^m$.*

Formally, let $g_0(Z, A) : \mathcal{Z} \times \mathcal{A} \to \{0, 1\}$ be the human's prior rejector. Figure 2 illustrates the scenario: the prior is the region at the boundary of the human predictor h. Let $K(.,.) : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$ be the similarity measure that the human employs to measure the degree of similarity between two instances.

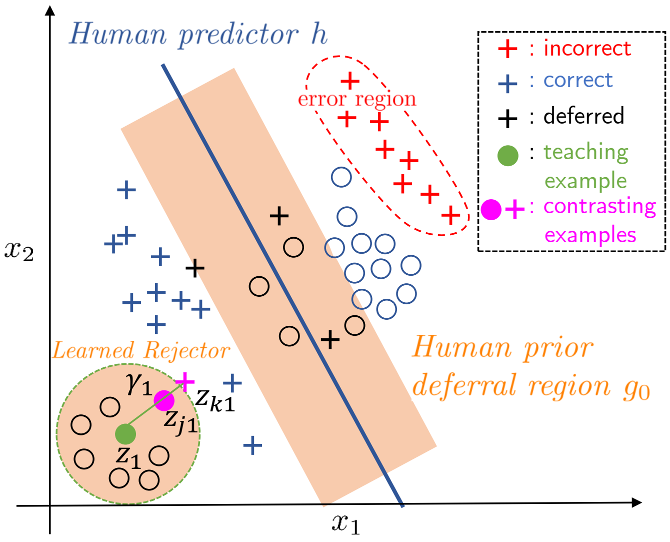


Figure 2: Illustration of human rejector on toy example. The task is classification with labels $\{o, +\}$, the human prediction h is the blue line and the prior g_0 is the shaded orange region surrounding the boundary. Points in red is where the human is incorrect, in blue correct and in black point deferred to the AI. The AI is assumed to be correct on examples far from the human boundary. The human receives a teaching example z_1 (in green) with radius γ_1. Also shown are the two contrasting examples z_{j1} and z_{jk} (in pink) that define the region.

The human's rejector uses a learned rule if they had observed an example similar with respect to $K(.,.)$ during teaching, otherwise falling back on their prior:

$$r(Z, A; \theta_r) = \begin{cases} vote(B(Z)) & , \; if \;\; B(Z) \neq \emptyset \\ g_0(Z, A) & , \; otherwise \end{cases} \tag{3}$$

where $B(Z)$ is the set of all points in D_T that they observed in training sufficiently similar to Z:

$$B(Z) = \{i \in [m] \mid K(Z, z_i) > \gamma_i\} \tag{4}$$

The degree of similarity is measured by a scalar γ_i that the human sets for each teaching example, in figure 2 all the points in the shaded ball have $B(Z) = \{z_1\}$. The rule vote$(B(Z))$ defines the label for all points similar to Z based on a weighted decision:

$$vote(B(Z)) = \arg\max_{k \in \{0,1\}} \frac{\sum_{i \in B(Z)} \mathbb{I}\{r_i = k\} K(Z, z_i)}{\sum_{i \in B(Z)} K(Z, z_i)} \tag{5}$$

Where r_i is the deferral rule that the human has learned on example z_i.

We can possibly further assume that the prior takes a rather simple form of thresholding the predictor's error: $g_0(Z, A) = \mathbb{I}\{\mathbb{P}(h(Z, A) \neq Y | Z, A) \geq \epsilon\}$ for some $\epsilon > 0$. One possibility for ϵ is the error rate of the AI.

Discussion on the Assumptions. In our assumptions above, we assumed knowledge of the following parameters: the human predictor $h(Z, A)$, the prior human rejector $g_0(Z, A)$ and the human similarity measure $K(,, .)$. In fact, as we will see, we only need to know the expert error distribution $\mathbb{E}[l(h(Z, A), Y)|Z, A]$ rather than the full expert predictor; it may be reasonable to estimate the expert's error distribution from previously collected data. The prior rejector g_0 can also be learned by testing the human prior as evidenced by prior work on capturing human priors [KWKH19, BPR+19], otherwise, a reasonable guess is the human deferring by just thresholding their own error rate. Finally to teach the human, we need a proxy for the similarity measure $K(., .)$. This can be obtained in many ways: one can learn this metric with separate interactions with the human, see [Ilv19, QTZ+09], or rely on an AI based similarity measure e.g. from neural network embeddings [RG19]. This last proxy is readily available and in the framework of our study, we believe it is reasonable to use.

An important part of the rejector is the associated radius γ_i with each teaching example i, the radius allows the human to generalize from each teaching example to the entire domain. The human learning process leaves the setting of γ_i completely up to the human and is not observed. However, we hope to directly influence the value of γ_i that the human sets during teaching.

5 Teaching a Student Learner

Formulation. The previous section introduced the model of the human learner, in this section we will set out our approach to select the teaching examples for the onboarding stage. Essentially, our approach is trying to find local regions, balls with respect to $K(., .)$, that best teach the human about the AI. We assume access to a labeled dataset $S = \{x_i, z_i, y_i\}_{i=1}^n$ that is independent from the training data of the AI model. For each point we can assign a deferral decision r_i that the human should undertake that minimizes the system loss. Explicitly, the optimal deferral decision r_i is defined to select who between the human and AI has lower loss on example i:

$$r_i = \mathbb{I}\{\mathbb{E}[l(h(z_i, a_i), y_i)] \geq \mathbb{E}[l(\pi_Y(x_i), y_i]]\} \tag{6}$$

Note that to derive r_i we only need to know the loss of the human on the teaching set and not their predictions. Define then $S^* = \{x_i, z_i, r_i\}_{i=1}^n$ as a set of examples alongside deferral decisions. As mentioned previously, the human is also learning a radius γ_i with each example. The radius γ_i should be set large enough to enable generalization to the domain, but small enough for the region to be coherent so that the human can interpret why should they follow the optimal deferral decision.

Let $D_z \subset S^*$ and let D_γ be the set of radiuses associated with each point in D_z and define $D = (D_z, D_\gamma)$. Define the loss of the human learner $M(., .; D)$ now only parameterized by the teaching set D as follows:

$$L(D) = \sum_{i \in S} l\left(M(z_i, a_i; D), y_i\right) \tag{7}$$

Greedy Selection. Note that since the radiuses set by the human are learned only after observing the example, we try to jointly optimize for the teaching point and the radius to teach. To optimize for D, consider the following greedy algorithm (`GREEDY-SELECT`) which starts with an empty set D_0 , and then repeats the following step for $t = 1, \cdots, m$ to select the example z and radius γ that leads to the biggest reduction of loss if added to the teaching set:

$$z, \gamma = \arg \min_{z_i \in S \setminus D_t, \gamma} L(D_t \cup \{z_i, \gamma\}), \tag{8}$$

$$\text{s.t. } \exists k \in [n] \ s.t. \ \gamma = K(z_i, z_k), \tag{9}$$

$$\text{and } \frac{\sum_{j \in [n], K(z_i, z_j) > \gamma} \mathbb{I}_{r_j = r_i}}{|\{j \in [n], K(z_i, z_j) > \gamma\}|} \geq \alpha \tag{10}$$

Constraint (9) restricts γ to be the similarity between z and another data point and constraint (10) ensures that $\alpha\%$ of all points inside the ball centered at z share the same deferral decision as z. The scalar α is a hyperparameter that controls the consistency of the local region: when $\alpha = 1$, the region is perfectly consistent and we call this setting CONSISTENT-RADIUS, and when $\alpha = 0$ the constraint is void and we dub the algorithm as DOUBLE-GREEDY.

Contrasting examples. Note that the radius γ is actually defined by two points: the point z_k in equation (9) that defines the boundary and an interior point z_j that is the least similar point to z with similarity at least γ; these two points are illustrated in Figure 2 with the color pink. These two points must actually share opposing deferral actions with $r_k \neq r_j$ and thus are contrasting points later used as a way to describe the local region.

Theoretical Guarantees. Let D_t be the solution found by the greedy algorithm and D^* the optimal solution. We now try to see how we can compare D_t to D^*. To do so, we make a further assumption on the choice of radiuses that the human sets.

Assumption 3 (Radius consistency)**.** *We assume that if $j \in B(z_i) \cap S$ then $r_i = r_j$. This implies that if z_j is at least γ_j close to z_i, then the best deferral choice for j is the same as that for i. This assumption is an assumption on the choice of γ_i's for each example in the teaching set.*

Assumption 3 in essence says that the human is always conservative enough such that the lesson drawn from example i is consistent on S. This translates to setting $\alpha = 1$ in our algorithm; when $\alpha < 1$ the guarantees may not hold. Given this assumption we can deduce that our objective function is now submodular and monotone. Furthermore, equipped with the fact that our problem is submodular we can derive the following guarantee on the gap of performance of our algorithm versus the optimal teaching set, as the next theorem demonstrates.

Theorem 1. *Let $F(X) = L(\emptyset) - L(X)$, when $\alpha = 1$, $F(.)$ is submodular, monotone and positive. Moreover, the GREEDY-SELECT algorithm described above achieves the following performance compared to the optimal set D^*:*

$$\underbrace{L(D_m)}_{\textit{loss of chosen set}} \leq (1 - \frac{1}{e}) \underbrace{L(D^*)}_{\textit{loss of optimal set}} + \frac{1}{e} \underbrace{L(\emptyset)}_{\textit{loss of prior rejector}}$$

All proofs can be found in Appendix B.

Theorem 1 gives a guarantee on the subset chosen by the greedy algorithm with an $1 - \frac{1}{e}$ approximation factor, one can ask if we can do better. We prove that a generalization of our problem is in fact NP-hard in the appendix. In what was previously discussed, the dataset that we measure performance on and that we teach from are the same. We generalize to have a separate training set

Algorithm 1 Our Human Teaching Approach

```
Input: Teaching set $D$
for $i = 1, \cdots, m$ do
    Stage 1: Testing. Test the human on example $z_i$ with AI message $a_i$
    Stage 2: Feedback. Show human feedback of actual label $y_i$, AI prediction $\pi_i$, and recommended deferral action $r_i$
    Stage 3: Lesson Generalization. Show the two contrasting examples $z_j$ and $z_k$ and high level features about the region to allow generalization around $z_i$.
    Stage 4: Lesson Reinforcement. We ask the human to write a rule $R_i$ that describes the region surrounding the example $z_i$ and which action they should take.
end for
```

S_T and a validation set S_V and define the loss of the human with respect to S_V and now define our optimization problem in terms of finding a minimal size subset D that achieves a certain loss $\delta \geq 0$:

$$D^*_\delta = \arg\min_{D \subset S_T} |D| \quad s.t. \sum_{i \in S_V} l\left(M(z_i, a_i; D), y_i\right) \leq \delta \tag{11}$$

Proposition 1. *Problem* (11) *is NP-hard.*

The reduction is to the set cover problem and can be found in Appendix B.

Human Teaching Approach. After running our greedy algorithm, we obtain a teaching set D that we now need to teach to the human. We rely on a four stage approach for teaching on each example so that they are able to learn and generalize to the neighborhood around it shown in Algorithm 1. The human first predicts on the example z, then they receive feedback on their prediction and the AI's prediction. We then show them a description of the region around the example that helps them learn the radius. Specifically, we show them the two contrasting examples z_j and z_k defined by γ_i and high level features about the neighborhood. Finally, we ask them to formalize in writing a rule describing the region and the action to take inside that region. This rule that they write per example helps the human in creating a set of guidelines to remember for when to rely on the AI and ensures that they reflect on the teaching material.

6 Experimental User Study

We provide code to reproduce our experiments [1]. Additional experimental details and results are left to Appendix F.

6.1 Experimental Preliminaries

Experimental Task and Dataset. Our focus will be on *passage-based question answering* tasks. These are akin to numerous real world applications such as customer service, virtual assistants and information retrieval. It is of interest as relying on an AI can reduce the time one needs to answer questions by not reading the entire passage and as an experimental setup it allows a greater range in

[1] `https://github.com/clinicalml/teaching-to-understand-ai`

the type of *sub-expertise* we can allow for compared to experimental tasks in the literature. We rely on the HotpotQA dataset [YQZ+18] collected by crowdsourcing based on Wikipedia articles. We slightly modify the HotpotQA examples for our experiment by removing at random a supporting sentence from the two paragraphs. The supporting sentence removed does not contain the answer, so that each question always has an answer in the passage, however, it may not always be possible to arrive at that answer. This was done to make the task harder and create incentives for expert humans to use the AI. We further remove yes/no questions from the dataset and only consider hard multi hop questions from the train set of 14631 examples and the dev set of 6947 examples.

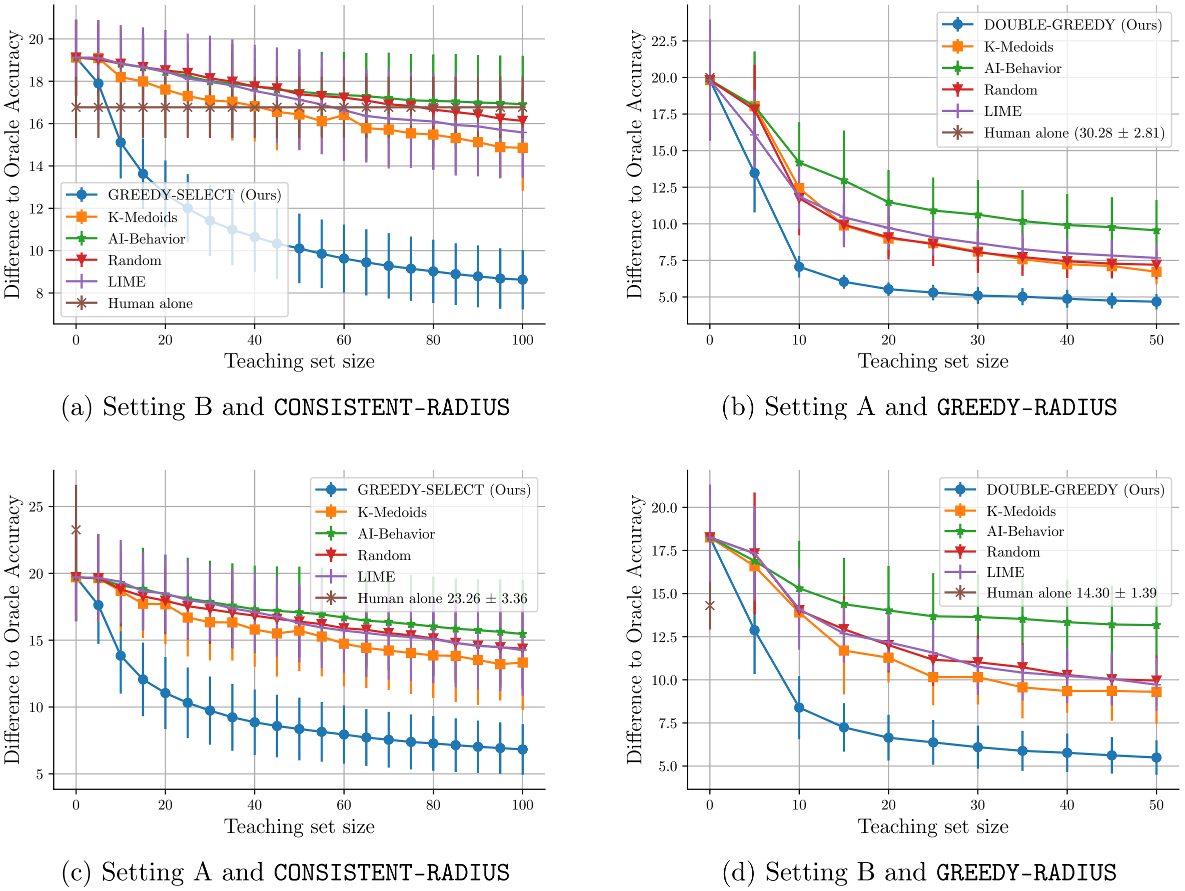
Simulated AI. One of the top performing models on HotpotQA is SAE-large: a graph neural network on top of RoBERTa embeddings [THW+20]. We performed a detailed error analysis in Appendix C of the SAE-large model predictions on the dev set. However, our analysis uncovered only few and small regions of model error. For our experimental study, we want to evaluate the effect of teaching in two ways: 1) through systematically checking the validity of the user lessons and 2) through objective task metrics. The SAE model makes it harder for us to do both especially with a limited number of responses from crowdworkers. For this reason, we decided to create a simulated AI whose error regions are more interpretable. We first cluster the dataset using K-means with k_p clusters based on only the paragraph embeddings obtained from a pre-trained SentenceBERT model [RG19]. The simulated AI model is parameterized by a vector $err_p \in [0, 1]^{k_p}$ where the probability of error of the AI on cluster i by $err_p[i]$. The answer of the AI when it is incorrect is manually constructed to be reasonably incorrect: for example if the answer asks for a date, we provide an incorrect date rather than a random sentence. To summarize, the AI for each cluster in the data has a specified probability of error that is constant on the cluster. To show that each cluster computed has a distinct meaningful theme, we retrieve the top 10 most common Wikipedia categories in each cluster. The full categories are shown in Appendix F; example cluster categories include singers/musicians, movies and soccer (but not football).

Metrics. Our aim will be to measure objective task performance and effort through the proxy of time spent on average per example. Our task performance metric is the F1 score on the token level [RZLL16]; we will measure this when considering the final predictions (Overall F1), on only when the human defers (Defer F1) and when the human does not defer (Non-Defer F1). We will also measure *AI-reliance*: this is calculated as how often they rely on the "Let AI answer for you" button in Figure 4a.

6.2 Simulated Users

Before we experiment with real human users, we evaluate the teaching complexity, i.e. the relation between teaching set size and human accuracy, of our teaching algorithm on simulated human learners that follow our assumptions. We further evaluate the robustness of our approach when we do not have full knowledge of the human parameters.

AI and Human model. We use the simulated AI model with $k_p = 15$ and a vector of errors err_p where for each i, $err_p[i]$ is drawn *i.i.d.* from Beta$(\alpha_{ai}, \beta_{ai})$. The human predictor is analogous to the AI model with a different vector of probabilities err'_p sampled from Beta(α_h, β_h). The human prior thresholds the probability error of the human to a constant ϵ. Finally, the human similarity



(a) Setting B and `CONSISTENT-RADIUS`

(b) Setting A and `GREEDY-RADIUS`

(c) Setting A and `CONSISTENT-RADIUS`

(d) Setting B and `GREEDY-RADIUS`

Figure 3: Teaching set size versus the negative difference between the human's learner test accuracy under the different methods compared to ORACLE. We plot the average result across 10 trials and standard deviation as error bars.

measure is the RBF kernel on the passage embeddings i.e. $K(x,x') = e^{-|x-x|^2}$. In this setup both the human and AI contexts are identical and the AI does not send any messages to the human.

Baselines. We implement a domain cover subset selection baseline in K-Medoids, the LIME selection strategy by [RSG16] with 10 features per example following [LLT20] (LIME), random selection baseline (RANDOM) and a baseline that greedily selects the point that helps a 1-nearest neighbors learner best predict the AI errors (AI-BEHAVIOR). Finally, we also compare to the optimal rejection rule computed with knowledge of the human and AI error rates by picking the lower one (ORACLE). The ORACLE rejector is an upper bound on achievable performance by any possible rejector regardless of the human student model.

Experimental setup. We will compare to the baselines as we vary the size of the teaching set D_T. To illustrate the effectiveness of the teaching methods, we focus on two settings: A) the Human is less accurate than the AI but their prior rejector rarely defers where we set the following and B) the Human is more accurate than the AI but their prior rejector over defers to the AI. These two settings is where teaching is most beneficial as the prior is erroneous. Specifically in setting A) we set

Condition	Oracle Gap @n=30
Full Information	6.38 ± 1.56
Missing g_0	6.90 ± 1.80
Noisy Radius	9.74 ± 3.0
Missing h	13.47 ± 5.07
No Information+Noise	15.12 ± 4.00
Prior only	16.72 ± 1.22
Human Alone	19.8 ± 2.80

Table 1: Test Accuracy gap between `DOUBLE-GREEDY` and ORACLE at teaching set of size 30 under various conditions. This is performed under setting B.

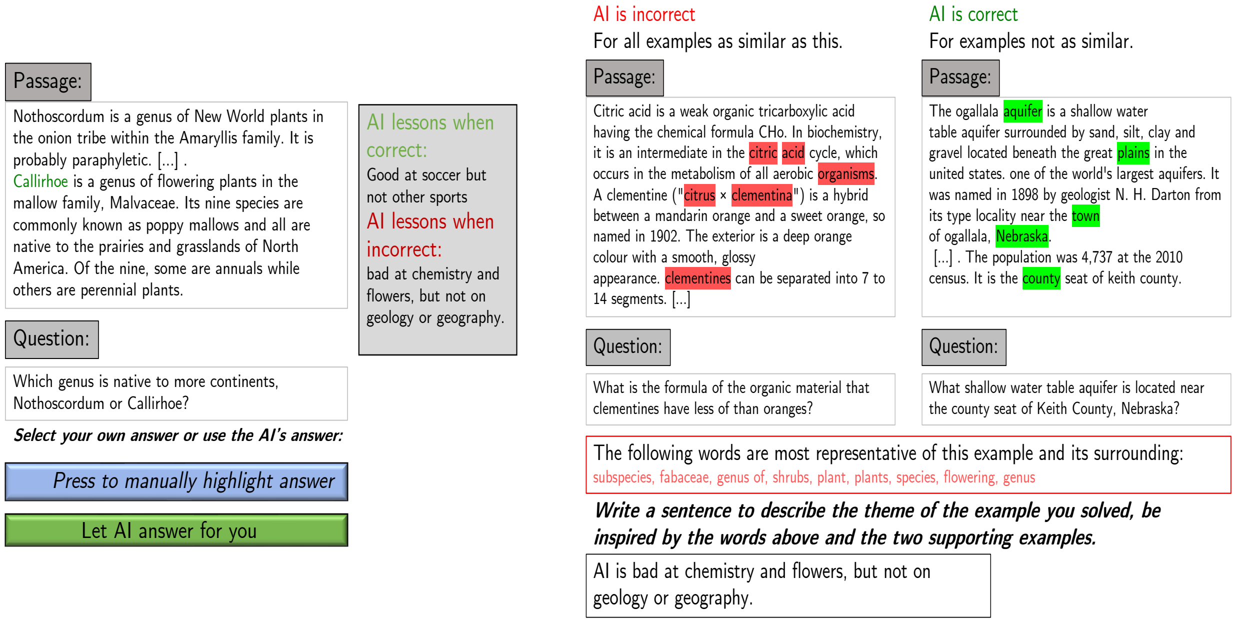
the following: $(\alpha_{ai} = 2, \beta_{ai} = 1)$ (the pdf is a straight line from the origin to $(1, 2)$), $(\alpha_h = 1, \beta_h = 1)$ (uniform distribution) and $\epsilon = 0.1$ and B) we set $(\alpha_{ai} = 1, \beta_{ai} = 1), (\alpha_h = 2, \beta_h = 1)$ and $\epsilon = 0.9$. We evaluate for each setting 10 different random settings of the human and AI error probability vectors and average the results.

Results. Figure 3 shows the gap between Oracle and human accuracy on the dev set compared to the size of the teaching set for each of the methods. We can see that our approach is able to outperform the baselines under setting B with `CONSISTENT-RADIUS`. We observe a wide gap between our method and the baselines, this is because the teaching examples here must focus on only a select number of the clusters and cover them sufficiently. With the greedy radius selection, we require fewer examples to reach high accuracy and the gap between our method and the baselines narrows.

Robustness to Misspecification of Human model. We evaluate accuracy when the human is not learning the correct radius; this simulates noise in the learning process. The radius γ_i that the human learns is a noisy version of $\hat{\gamma_i}$ where we add a uniformly distributed noise $\delta \sim \mathcal{U}(-(1-\hat{\gamma_i})/2, (1-\hat{\gamma_i})/2)$ to it. We then evaluate when we have no knowledge of the prior rejector g_0 or/and no knowledge of the human predictor h. When we don't know either of these parameters, we replace them by a random binary vector Bernoulli$(1/2)^n$ on the teaching set. Results are shown in Table 1. We can see that even if we don't have knowledge about the prior, accuracy is not impacted. However, if we don't have knowledge about the predictor h, then performance drops significantly. To evaluate how much information about h we need to properly teach the human, we learn a teaching set assuming the human's error probability is $err'_p + \boldsymbol{\delta}$ where $\boldsymbol{\delta}$ has each component drawn from $\{-\delta, \delta\}$ uniformly where $\delta > 0$. On setting B with `DOUBLE-GREEDY`, we can tolerate up to 0.25 error in knowledge about cluster error probability with no noticeable drop in performance; full results are in Appendix D. Note that when we don't have any knowledge about the human and the learning process is noisy, teaching is impacted.

6.3 Crowdsourced Experiments Details

Testing user interface. Our user interface during testing is shown in Figure 4a which shows a paragraph and its associated question. The human can either submit their own answer or let the AI answer for them using a special button. However, the interface does not display the AI's answer



Passage:

Nothoscordum is a genus of New World plants in the onion tribe within the Amaryllis family. It is probably paraphyletic. [...] .
Callirhoe is a genus of flowering plants in the mallow family, Malvaceae. Its nine species are commonly known as poppy mallows and all are native to the prairies and grasslands of North America. Of the nine, some are annuals while others are perennial plants.

Question:

Which genus is native to more continents, Nothoscordum or Callirhoe?

Select your own answer or use the AI's answer:

Press to manually highlight answer

Let AI answer for you

AI lessons when correct:
Good at soccer but not other sports
AI lessons when incorrect:
bad at chemistry and flowers, but not on geology or geography.

AI is incorrect
For all examples as similar as this.

Passage:

Citric acid is a weak organic tricarboxylic acid having the chemical formula CHo. In biochemistry, it is an intermediate in the citric acid cycle, which occurs in the metabolism of all aerobic organisms. A clementine ("citrus × clementina") is a hybrid between a mandarin orange and a sweet orange, so named in 1902. The exterior is a deep orange colour with a smooth, glossy appearance. clementines can be separated into 7 to 14 segments. [...]

Question:

What is the formula of the organic material that clementines have less of than oranges?

AI is correct
For examples not as similar.

Passage:

The ogallala aquifer is a shallow water table aquifer surrounded by sand, silt, clay and gravel located beneath the great plains in the united states. one of the world's largest aquifers. It was named in 1898 by geologist N. H. Darton from its type locality near the town of ogallala, Nebraska.
[...] . The population was 4,737 at the 2010 census. It is the county seat of keith county.

Question:

What shallow water table aquifer is located near the county seat of Keith County, Nebraska?

The following words are most representative of this example and its surrounding:
subspecies, fabaceae, genus of, shrubs, plant, plants, species, flowering, genus

Write a sentence to describe the theme of the example you solved, be inspired by the words above and the two supporting examples.

AI is bad at chemistry and flowers, but not on geology or geography.

(a) Testing interface

(b) Teaching interface

Figure 4: On the left in subfigure (a) is the testing interface shown for an example. This is the same interface that is also shown at the beginning of each teaching example. After the human predicts and we are in the teaching phase, we show them the correct answer and transition to the interface in subfigure (b) that shows the two supporting examples for the example in (a), the top weighted words in the region and asks the user to write down their rule for the example.

or any explanation, which forces the user to rely solely on their mental model and the teaching examples to make a prediction. This was done so that we can control for the effect of teaching solely, as showing the AI prediction at test time leaks information about the AI beyond what was shown in the teaching set. Moreover, not showing the AI prediction forces the human to explicitly think about the AI performance. The right panel next to the passage shows the lessons that the user wrote down during teaching.

Teaching user interface. Following our teaching algorithm, during teaching, the worker is first faced with the same user interface as in test time. The difference is that *after* they answer, they receive feedback on the correctness of their answer and can see the AI's answer. We then show the human the two constrasting examples with LIME word highlights. As a high level description of the local region, we show the top 10 most weighted words obtained by LIME in the ball surrounding the original teaching example [RSG16] (see Figure 4b). After they observe the two supporting examples, they are asked to write a sentence that describes the lesson of the example. These sentences are available during test-time for the workers to review as help for answering new questions.

Experimental Design and Baselines. The experimental teaching setup proceeds in three stages. The first stage (Stage 0) is a tutorial that introduces the task with two examples and where we gather the worker's demographic information, knowledge of machine learning and how often they

Metric	Ours-Teaching (all)	No-Teaching	LIME (all)	Ours (acc)	Ours (inacc)	LIME (acc)	LIME (inacc)
Overall F1	58.2 ± 3.4	57.6 ± 3.4	52.9 ± 3.4	62.8 ± 4.7	53.5 ± 4.9	56.5 ± 6.4	52.0 ± 4.2
Defer F1	50.7 ± 4.7	57.8 ± 4.9	48.1 ± 5.3	53.4 ± 6.7	50.0 ± 6.8	44.6 ± 9.0	49.9 ± 6.5
Non-Defer F1	67.6 ± 4.7	57.6 ± 4.7	56.9 ± 4.6	73.92± 6.2	60.6 ± 7.1	70.0 ± 8.6	53.7 ± 5.4
Time/ex (min)	0.60 ± 0.03	0.62 ± 0.03	0.68 ± 0.04	0.54 ± 0.04	0.68 ± 0.05	0.65 ± 0.08	0.69 ± 0.05
AI-Reliance (%)	55.2 ± 3.6	48.9 ± 3.6	45.4 ± 3.6	53.3 ± 4.9	58.9 ± 5.0	52.8 ± 3.6	43.6 ± 4.3

Table 2: Comparison of the metrics between our teaching condition (split into all participants, those who gave accurate lessons (acc) and those who didn't (inacc), see description below), the `No-teaching+AI-prediction` condition and LIME teaching. Shown are averages across all participants with 95% confidence interval error bars. The F1 of the AI alone in this setting is 46.7%; we did not separately measure the F1 of the human in isolation.

visit Wikipedia. Stage 1 is the teaching stage where the worker solves 9 teaching examples and stage 2 is the testing phase where the worker solves 15 questions with no feedback. After the two stages is an exit survey where users are asked about their decision process for using the AI. The two stage experimental design mimics what we believe would be a realistic deployment in practice; we don't expect feedback to be possible during deployment, but rather only in a specialized teaching phase. We randomly assign each participant to one of three conditions.

In the first condition the participants go through the entire pipeline described above (`Ours Teaching`). The second is condition is called (`LIME-Teaching`) where LIME is first used to obtain 18 examples. During teaching, users are asked to solve the first 9 questions and are then shown: LIME highlights of the example, performance feedback and asked to write a lesson of what they learned. Then users view the 9 remaining examples with LIME highlights without needing to solve them or write lessons. The difference with our method is that workers don't see the supporting examples or the word level description of the regions. The third is a baseline condition (`No-teaching+AI-prediction`) that makes the following modifications to the experimental design: the participants skip the teaching stage (Stage 1) and immediately proceed to the testing phase (Stage 2). However, during the testing phase, the participants *can see the AI prediction* before they press the use AI button which gives them an edge compared to the teaching condition.

Participants We recruited 50 US based participants from Amazon Mechanical Turk per each condition (150 total) and initial pilot studies were also conducted with graduate students in computer science at a US university. Participants in the non-teaching baseline were paid $3 for 10 minutes of work and those in the teaching condition received $6 for 20 minutes of work. Any demographic information we gathered in our study is kept confidential and workers were asked to consent to their use of their responses in research studies.

AI and Test Set details. The simulated AI had $k_p = 11$ and was randomly chosen to have probability of error 0 or 1 on each cluster. This means there are clusters where the AI is perfect on and other clusters where the AI is always wrong. We split the HotpotQA dev set into two parts 80:20 for the teaching and testing set respectively. To obtain the 9 teaching examples we run `GREEDY-SELECT` with the consistent radius strategy with no knowledge of g_0 or h. The examples in the testing phase was obtained first by filtering the data using K-medoids with $K = 200$ as a way to get diverse questions. Then each participant received 7 random questions from the filtered set on which the AI was correct and 8 on which the AI is incorrect.

Metric	Ours-Teaching (ID)	No-Teaching (ID)	Ours (OOD)	No-Teaching (OOD)
Overall F1	56.8 ± 3.6	56.0 ± 3.6	70.9 ± 10.5	72.86 ± 10.7
Defer F1	51.42 ± 4.9	57.8 ± 5.2	42.95 ± 17.2	56.7 ± 18.8
Non-Defer F1	63.7 ± 5.2	54.4 ± 5.1	96.05± 5.82	85.0 ± 11.5
AI-Reliance (%)	56.1 ± 3.8	49.4 ± 3.8	47.3 ± 11.6	42.9 ± 11.8

Table 3: Comparison of the metrics on clusters that were seen during teaching with our method (ID for in distribution) compared to performance on clusters that were not seen during teaching (OOD for out of distribution). We also show the performance of the no-teaching baselines on the two cluster sets as a reference point. The errors on the OOD estimates are much higher as there are much fewer samples in the not-seen clusters.

Further details can be found in Appendix F.

6.4 User Study Observations and Results

Teaching enables participants to better know when to predict on their own, but not when to defer to the AI. The first three columns of Table 2 display the metrics measured across both conditions on all participants. We can first note that participants with teaching are able to predict overall just as well as participants in the baseline no-teaching condition who have additional information about the AI prediction at test time. Moreover, participants who received teaching can better recognize when they are able to predict better than the AI. There is a difference significant at p-value 0.05 ($t = 2.9$, from a two sample t-test) of the F1 score when the human doesn't defer between our method and the no-teaching baseline and significant at p-value 0.001 ($t = 3.2$) compared to LIME. However, the participants in the teaching condition deferred to the AI when it was incorrect more often than those in the no-teaching baseline condition. A positive difference significant at p-value 0.05 ($t = -2.0$) in F1 when the humans defers for `No-teaching+AI-prediction` workers. An explanations for this is that the participants might press the use AI button on examples where their own prediction agrees with that of the AI instead of manually selecting the answer which takes more effort.

Accurate teaching lessons might predict improved task performance and our method teaches more participants than LIME. Given our knowledge about the clusters and the AI, the correct form of the teaching lesson of each example is "AI is good/bad at TOPIC" where TOPIC designates the theme of each cluster amongst a set of 11 topics which include soccer, politics, music and more. Manually inspecting the lessons of the 50 participants without seeing their test performance, we found that 25 out of 50 participants in our teaching condition were able to properly extract the right lesson from each teaching example. The remaining 25 participants were split into two camps: those who gave explanations on question/answer type or too broad or narrow of explanations e.g. "AI is good at people" rather than a specific subgroup of musicians for example (14 out of 50), and those who gave irrelevant explanations (11 out of 50, this group performed non trivially and so could not be disqualified). Table 9 in Appendix F gives examples of the actual lessons that users wrote. Results for participants who had accurate vs not accurate lessons are shown in the last four columns of Table 2. The participants who had accurate lessons had a 9 point average overall F1 difference significant at p-value 0.01 compared to those with inaccurate lessons. With

LIME-Teaching we found that only 14 out of 50 participants were able to properly extract the right lessons. The difference between LIME and our method in enabling teaching is significant at p-value 0.02 with $t = 2.3$, however, we observe that accurate teaching has a similar effect in both conditions. Note, that even when participants have accurate lessons, they often don't always follow their own recommendations as evidenced by the low Defer F1 score.

Differences in performance on in-distribution and out-of-distribution examples. During teaching with our method we let the users solve 9 examples, each corresponding to a unique cluster. The data domain is in fact split into 11 clusters where the AI has a different error probability in $\{0, 1\}$ on each of them. Thus, there are 2 clusters where users have not seen examples from, which we call the out-of-distribution examples (OOD), and 9 from which they have, the in-distribution examples (ID). In table 3 we show the different metrics split into ID and OOD distribution for teaching participants in our method and for the no-teaching participants as a reference point. LIME-Teaching participants observe all the clusters during teaching so there is no distinction between ID and OOD. We can first observe a very high F1 for OOD examples where the human predicts (Non-Defer F1) for our method. This is also the case for the non-teaching participants, thus the increase in F1 lies with the nature of the examples in the OOD clusters rather than the distinction of them being ID versus OOD. On the other hand, we observe that Defer F1 is higher by 8.36 points on average for ID examples compared to OOD with our teaching method while we do not observe a difference in Defer F1 for the baseline non-teaching group. However, the results are not significant as the 95% confidence intervals overlap.

7 Additional Synthetic Experiments

Dataset. To complement our NLP-based experiments, we run a study on the CIFAR-10 image classification dataset [KH$^+$09] consisting of 32×32 color images drawn from 10 classes. For CIFAR we use a WideResNet [ZK16] with no data augmentation that achieves 90.46% test accuracy and the model is trained to minimize the cross entropy loss with respect to the target. We split the dataset into three distinct parts: training set for AI model (90% CIFAR train, 45k), teaching set to obtain teaching images (10% of CIFAR train set, 5k) and test set for the human learner (CIFAR test set, 10k).

Setup. We let $X = Z$ and use the respective models' last layer encodings as the input space to the teaching algorithm. The message the AI sends is the pair $A = (\hat{y}, \hat{c})$ consisting of the AI prediction and a confidence score (softmax output of model). We assume the human is following the human rejector Assumption 2 and is perfectly learning the radius and actions. We consider the human expert models considered in [MS20]: let $k \in [10]$, then if the image is in the first k classes the expert is perfect, otherwise the expert predicts randomly. The human prior rejector defers if the AI's confidence $\hat{c}$ is less than $\epsilon = 0.5$.

Results. We show the results in Table 4 for various teaching set sizes for the expert $k = 6$ and a learning curve in Figure 5; full results are in Appendix D. We compare our approach to solving the problem as learning to defer with the AI deferring to the human: we compare to the surrogate loss baseline in [MS20], the confidence baseline in [RBC$^+$19] and a ModelConfidence baseline which optimizes over the prior parameter ϵ. We find that with only 4 teaching examples, `DOUBLE-GREEDY` increases accuracy from 90.98 to 96.3 ± 0.1 on the test set.

Method	CIFAR (acc)
Prior only	90.98 ± 0.0
DOUBLE-GREEDY @T=4	96.3 ± 0.1
DOUBLE-GREEDY @T=8	96.4 ±0.1
DOUBLE-GREEDY @T=14	96.5 ±0.1
K-Medoids @T=4	94.58 ±0.3
K-Medoids @T=8	95.5 ± 0.2
K-Medoids @T=14	96.5 ± 0.2
Random @T=8	95.3 ± 0.5
Oracle	97.91
Surrogate Loss [MS20]	97.1
Confidence [RBC+19]	95.5
ModelConfidence	93.94

Table 4: Synthetic experiment on CIFAR-10, showing the test Accuracy for our method DOUBLE-GREEDY at different teaching set sizes and learning to defer baselines.

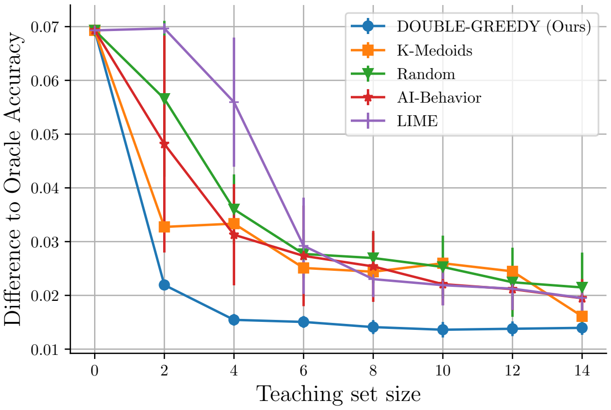


Figure 5: Synthetic experiment on CIFAR-10, showing difference between the performance of the methods and ORACLE (defined as taking the optimal decision at test time) for expert $k = 6$.

8 Discussion

Our work provides a general recipe for onboarding human decision makers to AI systems. We propose an exemplar based teaching strategy where humans are asked to predict on real examples and then with the help of similar examples and top features for the neighborhood, the human derives an explanation for the AI performance.

One limitation of our human experiments is that we used a simulated AI that has an easier to understand error boundary. This enabled us to have a more in-depth study of the crowdworker responses than otherwise would have been possible. Having a simulated AI which we perfectly understand where its error regions are, enables us to define what the "lessons" should be and thus evaluate if users are learning correctly. Future user studies will evaluate with non-simulated

AI models. We hypothesize that the example selection algorithm presented in this work will be sufficient, however, we might require better methods to illustrate the neighborhood for each example. Another limitation is that our test-time interface did not include model explanations, which was done to eliminate additional confounding factors when comparing approaches. Future work will evaluate whether the effect of teaching remains as significant when evaluating with test-time model explanations. Other limitations include the fact that we are using a proxy task of passage based question answering and proxy tasks have been documented to be misleading for evaluating AI systems [BLGG20]. Another limitation is the use of MTurk which may not ensure high quality workers and the final limitation is that our study only focuses on the onboarding phase of AI deployment.

Teaching is used in our work to influence human's perception of an AI model; this can be potentially used to manipulate workers into relying on AI agents in high stakes settings if the AI predictions during teaching were fabricated. While our work was conducted in a low stakes scenario and was designed to portray an accurate reflection of the AI performance, it is possible by manipulating the AI predictions during teaching to have the worker learn any desired rejector. We believe if the data used during teaching is not manipulated, then our approach can serve to give an unbiased overview of the AI.

Acknowledgments

HM and DS were supported by NSF AitF award CCF-1723344.

References

[ADRW00] Robert K Atkinson, Sharon J Derry, Alexander Renkl, and Donald Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of educational research*, 70(2):181–214, 2000.

[Ang05] Fabrizio Angiulli. Fast condensed nearest neighbor rule. In *Proceedings of the 22nd international conference on Machine learning*, pages 25–32, 2005.

[AWV$^+$19] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–13, 2019.

[BBH$^+$20] Emma Beede, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruamviboonsuk, and Laura M Vardoulakis. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[BKSD17] Aaron M Bornstein, Mel W Khaw, Daphna Shohamy, and Nathaniel D Daw. Reminders of past choices bias decisions for reward in humans. *Nature Communications*, 8(1):1–9, 2017.

[BLCW09] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[BLGG20] Zana Bucinca, Phoebe Lin, Krzysztof Z Gajos, and Elena L Glassman. Proxy tasks and subjective measures can be misleading in evaluating explainable ai systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 454–464, 2020.

[BNK+19] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 2–11, 2019.

[BNK+21] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. Is the most accurate ai the best teammate? optimizing ai for teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11405–11414, 2021.

[BPR+19] David D Bourgin, Joshua C Peterson, Daniel Reichman, Stuart J Russell, and Thomas L Griffiths. Cognitive model priors for predicting human decisions. In *International conference on machine learning*, pages 5133–5141. PMLR, 2019.

[BVVD+17] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

[BWZ+20] Gagan Bansal, Tongshuang Wu, Joyce Zhu, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel S Weld. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. *arXiv preprint arXiv:2006.14779*, 2020.

[CMAS+18] Yuxin Chen, Oisin Mac Aodha, Shihan Su, Pietro Perona, and Yisong Yue. Near-optimal machine teaching via explanatory teaching sets. In *International Conference on Artificial Intelligence and Statistics*, pages 1970–1978, 2018.

[CPY+18] Arjun Chandrasekaran, Viraj Prabhu, Deshraj Yadav, Prithvijit Chattopadhyay, and Devi Parikh. Do explanations make vqa models more predictable to a human? *arXiv preprint arXiv:1810.12366*, 2018.

[CRA20] Eric Chu, Deb Roy, and Jacob Andreas. Are visual explanations useful? a case study in model-in-the-loop prediction. *arXiv preprint arXiv:2007.12248*, 2020.

[CWS+19] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. " hello ai": Uncovering the onboarding needs of medical practitioners for human-ai collaborative decision-making. *Proceedings of the ACM on Human-computer Interaction*, 3(CSCW):1–24, 2019.

[DHPZ19] Sanjoy Dasgupta, Daniel Hsu, Stefanos Poulis, and Xiaojin Zhu. Teaching a black-box learner. In *International Conference on Machine Learning*, pages 1547–1555, 2019.

[DKB19] Shayan Doroudi, Ece Kamar, and Emma Brunskill. Not everyone writes good examples but good examples can come from anywhere. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 12–21, 2019.

[DMH$^+$20] Rati Devidze, Farnam Mansouri, Luis Haug, Yuxin Chen, and Adish Singla. Understanding the power and limitations of teaching with imperfect knowledge. *arXiv preprint arXiv:2003.09712*, 2020.

[DT18] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[FBG19] Shi Feng and Jordan Boyd-Graber. What can ai do for me? evaluating machine learning interpretations in cooperative play. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 229–239, 2019.

[Gat72] Geoffrey Gates. The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 18(3):431–433, 1972.

[GBF$^+$20] Ana Valeria Gonzalez, Gagan Bansal, Angela Fan, Robin Jia, Yashar Mehdad, and Srinivasan Iyer. Human evaluation of spoken vs. visual explanations for open-domain qa. *arXiv preprint arXiv:2012.15075*, 2020.

[GBM$^+$17] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. PMLR, 2017.

[GK95] Sally A Goldman and Michael J Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

[GL13] Gyslain Giguère and Bradley C Love. Limits in decision making arise from limits in memory retrieval. *Proceedings of the National Academy of Sciences*, 110(19):7613–7618, 2013.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[GSG$^+$17] Chirag Gupta, Arun Sai Suggala, Ankit Goyal, Harsha Vardhan Simhadri, Bhargavi Paranjape, Ashish Kumar, Saurabh Goyal, Raghavendra Udupa, Manik Varma, and Prateek Jain. Protonn: Compressed and accurate knn for resource-scarce devices. In *International Conference on Machine Learning*, pages 1331–1340. PMLR, 2017.

[GSR$^+$21] Susanne Gaube, Harini Suresh, Martina Raue, Alexander Merritt, Seth J Berkowitz, Eva Lermer, Joseph F Coughlin, John V Guttag, Errol Colak, and Marzyeh Ghassemi. Do as ai say: susceptibility in deployment of clinical decision-aids. *NPJ digital medicine*, 4(1):1–8, 2021.

[HB20] Peter Hase and Mohit Bansal. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831*, 2020.

[HCMA$^+$18] Anette Hunziker, Yuxin Chen, Oisin Mac Aodha, Manuel Gomez Rodriguez, Andreas Krause, Pietro Perona, Yisong Yue, and Adish Singla. Teaching multiple concepts to a forgetful learner. *arXiv preprint arXiv:1805.08322*, 2018.

[HT07] John Hattie and Helen Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

[Ilv19] Christina Ilvento. Metric learning for individual fairness. *arXiv preprint arXiv:1906.00250*, 2019.

[KAB19] Rafal Kocielnik, Saleema Amershi, and Paul N Bennett. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

[KG14] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014.

[KH$^+$09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

[KKK16] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in neural information processing systems*, pages 2280–2288, 2016.

[KNJ$^+$20] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[KTWA14] Matt Kusner, Stephen Tyree, Kilian Weinberger, and Kunal Agrawal. Stochastic neighbor compression. In *International Conference on Machine Learning*, pages 622–630. PMLR, 2014.

[KWKH19] Yea-Seul Kim, Logan A Walls, Peter Krafft, and Jessica Hullman. A bayesian cognition approach to improve data visualization. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–14, 2019.

[KZSC21] Akash Kumar, Hanqi Zhang, Adish Singla, and Yuxin Chen. The teaching dimension of kernel perceptron. In *International Conference on Artificial Intelligence and Statistics*, pages 2071–2079. PMLR, 2021.

[LASS21] Ariel Levy, Monica Agrawal, Arvind Satyanarayan, and David Sontag. Assessing the impact of automated suggestions on decision making: Domain experts mediate model errors but take less initiative. In *CHI Conference on Human Factors in Computing Systems*, 2021.

[LCH$^+$19] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006*, 2019.

[LHL16] Daniel Link, Bernd Hellingrath, and Jie Ling. A human-is-the-loop approach for semi-automated content moderation. In *ISCRAM*, 2016.

[LLT20] Vivian Lai, Han Liu, and Chenhao Tan. " why is' chicago'deceptive?" towards building model-driven tutorials for humans. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[LRJ$^+$15] Min Hyung Lee, Joe Runde, Warfa Jibril, Zhuoying Wang, and Emma Brunskill. Learning the features used to decide how to teach. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 421–424, 2015.

[LT19] Vivian Lai and Chenhao Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 29–38, 2019.

[MPZ18] David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, pages 6150–6160, 2018.

[MS20] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

[PVG$^+$11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[QTZ$^+$09] Guo-Jun Qi, Jinhui Tang, Zheng-Jun Zha, Tat-Seng Chua, and Hong-Jiang Zhang. An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 841–848, 2009.

[RBC$^+$19] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

[RBS$^+$19] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning*, pages 5281–5290, 2019.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[RHY+20] Sherry Ruan, Jiayu He, Rui Ying, Jonathan Burkle, Dunia Hakim, Anna Wang, Yufeng Yin, Lily Zhou, Qianyao Xu, Abdallah AbuHashem, et al. Supporting children's math learning with feedback-augmented narrative technology. In *Proceedings of the Interaction Design and Children Conference*, pages 567–580, 2020.

[RP14] Jennifer J Richler and Thomas J Palmeri. Visual category learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 5(1):75–94, 2014.

[ŘS10] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[RZLL16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[SBB+14] Adish Singla, Ilija Bogunovic, Gabor Bartok, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *International Conference on Machine Learning*, pages 154–162, 2014.

[SC19] Sonia Jawaid Shaikh and Ignacio Cruz. 'alexa, do you know anything?'the impact of an intelligent assistant on team interactions and creative performance under time scarcity. *arXiv preprint arXiv:1912.12914*, 2019.

[SCD+17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[SCMA+17] Shihan Su, Yuxin Chen, Oisin Mac Aodha, Pietro Perona, and Yisong Yue. Interpretable machine teaching via feature feedback. 2017.

[SLGS21] Harini Suresh, Kathleen M Lewis, John V Guttag, and Arvind Satyanarayan. Intuitively assessing ml model reliability through example-based explanations and editing model inputs. *arXiv preprint arXiv:2102.08540*, 2021.

[SLL20] Harini Suresh, Natalie Lao, and Ilaria Liccardi. Misplaced trust: Measuring the interference of machine learning in human decision-making. *arXiv preprint arXiv:2005.10960*, 2020.

[SRFB+20] Alison Smith-Renner, Ron Fan, Melissa Birchfield, Tongshuang Wu, Jordan Boyd-Graber, Daniel S Weld, and Leah Findlater. No explainability without accountability: An empirical study of explanations and feedback in interactive ml. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[THW$^+$20] Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9073–9080, 2020.

[TRA$^+$20] Philipp Tschandl, Christoph Rinner, Zoe Apalla, Giuseppe Argenziano, Noel Codella, Allan Halpern, Monika Janda, Aimilios Lallas, Caterina Longo, Josep Malvehy, et al. Human–computer collaboration for skin cancer recognition. *Nature Medicine*, 26(8):1229–1234, 2020.

[TVYT21] Tony Tohme, Kevin Vanslette, and Kamal Youcef-Toumi. Improving regression uncertainty estimation under statistical change. *arXiv preprint arXiv:2109.08213*, 2021.

[VTYT20] Kevin Vanslette, Tony Tohme, and Kamal Youcef-Toumi. A general model validation and testing tool. *Reliability Engineering & System Safety*, 195:106684, 2020.

[WHK20] Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. *arXiv preprint arXiv:2005.00582*, 2020.

[WVW21] Jennifer Wortman Vaughan and Hanna Wallach. A human-centered agenda for intelligible machine learning. This is a draft version of a chapter in a book to be published in the 2020 - 21 timeframe., May 2021.

[XCK$^+$20] Yao Xie, Melody Chen, David Kao, Ge Gao, and Xiang'Anthony' Chen. Chexplain: Enabling physicians to explore and understand data-driven, ai-enabled medical imaging analysis. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[YQZ$^+$18] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

[YWVW19] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–12, 2019.

[ZGK$^+$17] Kai Zhong, Ruiqi Guo, Sanjiv Kumar, Bowei Yan, David Simcha, and Inderjit Dhillon. Fast classification with binary prototypes. In *Artificial Intelligence and Statistics*, pages 1255–1263. PMLR, 2017.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[ZLB20] Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 295–305, 2020.

[ZSZR18] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N Rafferty. An overview of machine teaching. *arXiv preprint arXiv:1801.05927*, 2018.

A Extended Related Work

Human-AI interaction. A significant amount of research has tried to understand the role of explanations on Human-AI team performance. [LT19] investigates the role of increasing levels of AI explanation on performance and find that beyond showing predicted labels accuracy does not increase. [LCH$^+$19] identify different regularizers that optimize for factors that help humans better simulate and verify AI predictions on recommendation tasks. [SRFB$^+$20] investigates how the ability of humans to provide feedback to the model reduced user frustration on a text classification task. [HB20] evaluated different explanation methods on the adult income dataset and on a movie reviews dataset found that only LIME helped for simulating the model and that subjective user ratings of explanation quality were not predictive of effectiveness. More research on the adult income dataset found that showing AI confidence improved trust but failed to improve AI-assisted accuracy [ZLB20]. [KAB19] studies how different types of errors an AI may have will lead to different perceptions of the AI by the user, and how setting expectations of the AI capabilities (e.g. its accuracy) improves the user experience. [BWZ$^+$20] show on a beer/book reviews sentiment classification task and on LSAT multiple choice questions that AI explanations beyond confidence scores don't improve performance but rather increase blind trust in the AI system. [LASS21] on the task of annotating clinical texts show that clinicians generally build a mental model of when to rely on automation, however, when the AI presents a complete suggestion versus an incomplete one, this causes experts to show less agency and makes them more likely accept wrong answers. In similar lines, [SLL20] studied how do humans incorporate AI recommendations as a function of their correctness and their prior knowledge of machine learning, and showed that people follow incorrect AI recommendations for tasks they predominantly complete correctly and that incorrect-abnormal recommendations were followed significantly less than incorrect normal recommendations. [CRA20] on the task of age prediction from images showed that the addition of explanations in the form of saliency maps did not improve accuracy nor did the quality of the saliency maps have much impact. [SLGS21] propose to visualize a given input's nearest neighbors to help better reason about the model's uncertainty and show an editor that allows users to edit aspects of the input and see how model predictions change, they found that this interface allowed some clinicians to build better intuition about the AI capabilities and limitations. Finally, a line of work has focused on human-AI interaction in healthcare applications: on chest X-rays [XCK$^+$20, GSR$^+$21], diabetic retinopathy [BBH$^+$20], skin cancer [TRA$^+$20] and breast cancer [BVVD$^+$17]. [YWVW19] study the effect of initial debriefing of stated AI accuracy compared to observed AI accuracy in deployment and find a significant effect of stated accuracy on trust, but that diminishes quickly after observing the model in practice; this reinforces our approach of building trust through examples that simulate deployment.

Explainability. Methods for explaining the decisions of ML models range from feature attribution (e.g. LIME [RSG16]), saliency methods for computer vision tasks (Grad-CAM [SCD$^+$17]), Example-based explanations [KKK16] and others. One of the basic forms of model explanations is calibrated confidence scores [GPSW17, TVYT21, VTYT20] These methods for explainability start from a set of desiderata (natural assumptions of what an explainability method should provide) and then formulate a given method that can be implemented without further data requirements. The common pitfall of these methods is that they are agnostic to the downstream expert, the desiderata is formulated from a perspective of a rational expert and are sometimes justified from user studies.

Machine Teaching. Machine teaching (MT) refers to the problem of choosing a minimally sized dataset that enables a student learner to learn a specific target function [ZSZR18]. Given a hypothesis class, its teaching dimension is the smallest sized set that enables an ERM learner to pick out the optimal classifier [GK95, KZSC21]. To mimic human learners, [SBB$^+$14] proposes a Bayesian learner based on a prior over a discrete hypothesis class, the learner maintains a distribution over each hypothesis that updates with each teaching example. They evaluate their approach on an image classification task where crowdworkers learn to distinguish different animals. This setting was extended to include explanations in the form of attention [CMAS$^+$18, SCMA$^+$17] and errors in learning priors and teacher knowledge [DMH$^+$20]. [DHPZ19] aims to teach a consistent black-box learner, while this formulation is attractive in regards to a human learner, the algorithm they provide requires an excessive amount of queries to the human that go beyond the teaching examples presented. [HCMA$^+$18] teaches a forgetful human learner multiple concepts where each concept maps to a single example, but the human may forget the concept later on. Our work separates itself by the use of a novel radius nearest neighbor model to approximate the human learning process.

Human Learning. [BKSD17] make the claim that humans makes decision by sampling similar experiences from memory instead of computing reward estimates for each possible action. Their experimental study involves users performing a two-armed bandit task with each example having a unique identifier. [GL13] make two claims about how humans make decisions: the first is that people often retrieve a limited set of items from memory when making decisions and the second is that training humans on idealized instances is more advantageous than training them on noisy or hard instances. They base their claims on two experiments: one where humans classify horizontal lines of different lengths and the other where they judge outcomes of baseball games. [RP14] review the literature on visual category learning, how we distinguish between different visual objects. They make a distinction between two different models of human decision making. The first is example-based that models assume that a category is represented in terms of the particular exemplars that have been experienced during learning. The other is rule based, people try to explicitly learn categories by forming simple rules. The conjecture is that for hard tasks, the example-based model is more accurate while for simpler ones, the rule-based approach is the driver.

Nearest Neighbor Compression. Our human student model is a more general case of a weighted nearest neighbor learner, this makes the teaching problem equivalent to that of compressing the number of samples nearest neighbors requires. Seminal work on compressing nearest neighbors introduced the condensed nearest neighbor rule [Gat72] and follow-up work introduced more robust versions but that still require the existence of a consistent subset [Ang05]. More recent work has focused on the generation of compressed subsets [KTWA14, ZGK$^+$17, GSG$^+$17].

A.1 Relation to Learning to Defer.

Our framework of Human-AI assisted decision making, dubbed teaching to defer (TTD), and its associated framing can be considered as the analog of the learning to defer framework described in [MS20] (LTD). In our setting, the human observes the AI prediction and then makes a prediction. In LTD, the AI model first decides using a rejector whether to predict on its own or defer to the human. There is no interaction in LTD between the human and the AI as the goal is to reduce the burden on the human expert. We borrow the notion of a rejector to formalize the thought process of the human deciding to or not to use the AI prediction. Table 5 highlights some of the main differences between the two frameworks.

System Objective. The objective in our framework is stated in equation (2), which can be compared to the the system objective from LTD [MS20]:

$$L(h,r) = \mathbb{E}_{(x,y)\sim\mathbf{P},m\sim M|(x,y)} \; [\; l(x,y,h(x))\mathbb{I}_{r(X)=0} + l_{\exp}(x,y,m)\mathbb{I}_{r(x)=1} \;] \tag{12}$$

Beyond the fact that in TTD, the human controls the rejector and in LTD the AI controls the rejector, a technical difference is the input to the rejector function r: in LTD it's the AI domain X, while in TTD it's the human domain Z and the AI prediction $\pi(X)$.

Human-AI Interaction. In LTD, when the AI predicts or defers, it does so without observing the human's prediction, and when the human predicts, they do so without seeing the AI prediction. On the other hand, in our framework, the human observes the AI's prediction and explanation before making their final prediction. This allows the human and AI to combine their predictions in a way that the LTD framework does not allow.

Table 5: Comparison on different dimensions between the teaching to defer framework in this paper (TTD) and the learning to defer framework (LTD) from [MS20, MPZ18, RBS$^+$19].

Dimension	LTD [MS20]	TTD (this paper)
Information at training	Samples from AI domain X, label Y, human prediction M	Samples from AI domain X, Human domain Z, label Y, and error distribution of AI and Human
Information at testing	AI domain X	AI domain X, Human domain Z, AI prediction π
AI training	joint training with rejector	trained without knowledge of human rejector
Knowledge about human	samples of prediction	error distribution
Form of rejector	no constraint	radius nearest neighbor defined by Assumption 2
Interaction between Human and AI	No by design, AI doesn't see the Human prediction and Human doesn't see the AI prediction	Yes
Final decision maker	AI or Human	Human
Does Human observe each example	No, since AI might not defer	Yes
Ease of Deployment	Needs re-training for every human expert	Same deployment for any human expert

B Theoretical Results and Proofs

B.1 Further Derivations

We expand on section "Teaching a Student Learner" (5) and decompose the loss of the human learner.

Since $\pi_Y(x)$ and $h(Z, A)$ are known and fixed, we can assign to each deferral decision at each point i a cost $c_i(r) \in \mathbb{R}^+$ and abstract away the inner classification decisions:

$$L(D) := \sum_{i \in S} l_c(r(z_i, a_i; D); c_i) \tag{13}$$

An example of l_c is $l_c^b := r(x_i; D)c_i(1) + (1 - r(x_i; D))c_i(0)$ which can be made equivalent to the 0-1 classification loss. It may be the case that neither of $c_i(0)$ or $c_i(1)$ are zero since there may be multiple correct decisions or that both be may be non-zero and equal. Now we further decompose the loss L into errors made by the prior and errors due to the learned rejector:

$$L(D) = \sum_{i \in S \ |B(z_i) \neq \emptyset} l_c \left(\frac{\sum_{j \in B(z_i)} \mathbb{I}\{r_j = 1\} K(z_i, z_j)}{\sum_{j \in B(z_i)} K(z_i, z_j)}; c_i \right) \quad \text{(errors by learned rejector)} \tag{14}$$

$$+ \sum_{i \in S \ | \ B(z_i) = \emptyset} l_c \left(g_0(z_i, a_i); c_i \right) \quad \text{(errors by prior)} \tag{15}$$

In the paper, we proved a guarantee in Theorem 1 on the performance of the `GREEDY-SELECT` algorithm when the hyperparameter α is set to 1 when optimizing the loss $L(.)$ (4). The loss L involves the human learner $M(.)$, however, one component of the human learner was left unspecified which is how they set the radius γ following every teaching example z. In what follows, we assume the human is perfectly learning the radius that the teaching process displays to them. Equivalently, when the human is shown the tuple $\{z, \gamma, r\}$ where z is the teaching example, γ is a radius and r is the deferral action, they now follow the deferral action r in the neighborhood of size γ around z.

When we set $\alpha = 1$, this defines a unique radius γ_i to each point $z_i \in S^*$ (the teaching set), this radius defines the largest neighborhood around z_i such that the optimal deferral action in that neighborhood is r_i. Thus our teaching set becomes $S^* = \{x_i, z_i, \gamma_i, r_i\}$ and we can now simplify our optimization problem by only searching for the teaching point z at each step (instead of jointly searching for the radius as well) as the radius is uniquely specified no matter what the current teaching set D_t is.

B.2 Proofs

The following proposition is part of the proof of Theorem 1.

Proposition 2. *Let $F(X) = L(\emptyset) - L(X)$, $F(.)$ is submodular, monotone and positive.*

Proof. **Monotonicity.** We prove that $L(.)$ is monotone decreasing which implies that $F(.)$ is monotone increasing. For notation simplicity we omit the AI message A from the prior rejector and make it only a function of Z, the proof remains valid even if we add A.

Initially $D_0 = \emptyset$ and $L(\emptyset)$ is the error rate of the human's prior rejector g_0 on the set S.

Induction argument: In the first step $D_1 = \{z_{i1}\}$ where z_{i1} is the training example that leads to the biggest error decrease of $L(.)$ (we don't use this fact so that this holds for any training example added, also note that since there is a unique correspondence from z_{i1} to r_{i1} and γ_{i1} we simplify the notation and only write z_{i1}), now note that:

$$L(D_1) - L(D_0) = \sum_{i \in S \ s.t. \ z_{i1} \in B(z_i)} l_c\left(r_{i1}; c_i\right) - l_c(g_0(z_i); c_i) \tag{16}$$

Note that other terms in the difference of equation (16) cancel out, what is left are points in S that the human starts to use their learned rejector on, i.e. those that are sufficiently close to z_{i1} call these set of points $\mathcal{I}$. For each $i \in \mathcal{I}$, if it was the case that $g_0(z_i) \in \arg\min_d l_c(d; c_i)$, then we know that r_{i1} and $g_0(z_i)$ have the same cost since r_{i1} is the optimal decision by definition. Now suppose that $g_0(z_i) \notin \arg\min_d l_c(d; c_i)$, then it must be the case that $r_{i1} = 1 - g_0(z_i)$ and this achieves a lower loss than $g_0(z_1)$. Therefore we have that:

$$L(D_1) - L(D_0) \leq 0$$

Now suppose we are at step $t+1$ of the algorithm and we add example $z_{i(t+1)}$ to obtain $D_{t+1} = \{z_{i1}, \cdots, z_{i(t+1)}\}$. Let us compute the difference:

$$L(D_{t+1}) - L(D_t) = \sum_{i \in S \ s.t. \ B(x_i) = \{z_{i(t+1)}\}} l_c\left(r_{i(t+1)}; c_i\right) - l_c(g_0(z_i); c_i) \tag{17}$$

Note that if there was point $i \in S$ where there exists $j \in D_t$ such that $z_j \in B(x_i)$, then the addition of $z_{i(t+1)}$ cannot change the final cost assigned to example i as if $z_{i(t+1)} \in B(z_i$, then we must have $r_{i(t+1)} = r_j$ by assumption 3. Thus the only element remaining in the difference is points that now have a neighbor in D_{t+1} but not in D_t, meaning those that only have $z_{i(t+1)}$ in their ball. The argument is now exactly as in the base case so that:

$$L(D_{t+1}) - L(D_t) \leq 0$$

which gives us the set of inequalities:

$$L(D_m) \leq \cdots \leq L(D_0)$$

and note that $L(.)$ achieves it's minimum value at $L(S) = L(D_{|S|}) \leq L(D_m)$.

Positivity. Note that $F(.)$ is positive as we assume l_c is positive and we obtain the result from monotonicity.

Submodularity. To make the proof easier, define the teaching ball $\widetilde{B}(D)$ to be the set of points in the training set S that have any teaching point $Z \in D$ in their ball $B(.)$. This implies if $B(z_i) = \{z_j\}$ then $z_i \in (\{z_j\})$; remember that $B(z_i)$ is the set of teaching points that are sufficiently close to z_i. Let $A \subset B \subset S$, let $l \in S \setminus B$, let us compute:

$$F(A \cup \{l\}) - F(A) - F(B \cup \{l\}) + F(B) = L(A) - L(A \cup \{l\}) + L(B \cup \{l\}) - L(B) \tag{18}$$
$$= \sum_{i \in S \ s.t. \ z_i \in \widetilde{B}(z_l) \setminus \widetilde{B}(A)} l_c(g_0(z_i); c_i) - l_c\left(r_l; c_i\right)$$

$$+ \sum_{i \in S \ s.t. \ z_i \in \widetilde{B}(z_l) \setminus \widetilde{B}(B)} l_c\left(r_l; c_i\right) - l_c(g_0(z_i); c_i) \tag{19}$$

$$= \sum_{i \in S \ s.t. \ z_i \in \left(\widetilde{B}(z_l) \cap \widetilde{B}(B)\right) \setminus \widetilde{B}(A)} l_c(g_0(z_i); c_i) - l_c\left(r_l; c_i\right) \geq 0 \tag{20}$$

The last term is positive as the optimal decisions r_i always improve on the prior.

□

Theorem 1. *Let $F(X) = L(\emptyset) - L(X)$, $F(.)$ is submodular, monotone and positive. Moreover, the* ***GREEDY-SELECT*** *algorithm described above achieves the following performance compared to the optimal teaching set D^*:*

$$\underbrace{L(D_m)}_{\text{loss of chosen set}} \leq (1 - \frac{1}{e}) \underbrace{L(D^*)}_{\text{loss of optimal set}} + \frac{1}{e} \underbrace{L(\emptyset)}_{\text{loss of prior rejector}} \tag{21}$$

Proof. The first statement of the theorem is proved in Proposition 2.

For the second statement of the theorem, the proof is simply restating the proof of Theorem 1.5 in [KG14] in the context of our problem which we do here for clarity. Let $D_i = (z_1, \cdots, z_i)$ the set that our algorithm produced at round i and $D^* = (z_1^*, \cdots, z_K^*)$ the optimal set.

For all $i \leq m$:

$$F(D^*) \leq F(D^* \cup D_i) \quad \text{(monotonicity)} \tag{22}$$

$$= F(D_i) + \sum_{j=1}^{m} F(D_i \cup D_{j-1}^* \cup z_j^*) - F(D_i \cup D_{j-1}^*) \quad \text{(telescoping)} \tag{23}$$

$$\leq F(D_i) + \sum_{z \in D^*} F(D_i \cup z) - F(D_i) \quad \text{(submodular } F\text{)} \tag{24}$$

$$\leq F(D_i) + m(F(D_i \cup z^{i+1}) - F(D_i)) \quad \text{(optimality of } z^{i+1}\text{)} \tag{25}$$

$$\tag{26}$$

re-arranging this final inequality with $\delta_{i+1} = F(D^*) - F(D_i)$ we get:

$$\delta_{i+1} \leq \delta_i (1 - \frac{1}{m})$$

iterating this last inequality till m, using the fact that $1 - x \leq e^{-x}$ and restating things in terms of $L(.)$ gets the final result in the theorem. □

B.3 Hardness result

Theorem 1 gives a guarantee on the subset chosen by the greedy algorithm with an $1 - \frac{1}{e}$ approximation factor, one can ask if we can do better. We prove that a generalization of our problem under Assumption 3 is in fact NP-hard.

Proposition 1. *Problem* (11) *is NP-hard.*

Proof. For simplicity we assume that the AI and human domains are identical and don't consider the AI message in the human rejector or predictor. The proof can be straightforwardly extended to the case when the domains differ and including the AI message.

Suppose we are given a collection of finite sets $A_1, \cdots, A_n$ jointly covering a set W. We reduce the problem of finding a smallest subcollection covering W to the teaching problem (11).

Let $S_V = W$, for each A_j, we associate it with a new teaching example $x_j \in S_T$ (unique from all elements of S_V and other elements of S_T) such that its neighbors are exactly the elements of A_j i.e. $K(x_j, x) = \infty$ iff $x \in A_j$ and $K(x_j, x) = 0$ iff $x \notin A_j$ (we construct the function K specifically to satisfy these requirements). Now we set the label $y_i = 1$ for each example $i \in S_V \cup S_T$ and let $h(x) = 1$ (human predictor) and $\pi_Y(x) = 0$ (AI predictor) for all x and we set $g_0(x) = 1$ (human prior rejector) so that the prior is wrong on all example: we should never defer while the prior always defers so the correct deferral decision is $d_i = 0$ (derived deferral decision) for all examples. We set the loss l_c to simply be the $0-1$ deferral loss (cost of 1 incurred if final prediction disagrees with label, otherwise a cost of 0), with this in mind note that $L_V(\emptyset) = |S_V|$ as with $D = \emptyset$ we use the prior rejector on all examples which always errs.

Once we pick a new example x_j (correspondence to the set A_j) to our set D that we are choosing, the only terms that are affected are those that are close to x_j which are exactly the elements of A_j, so that $L_V(\{x_j\}) = |S_V| - |A_j|$. Iteratively, when we add another example x_k to D the only terms affected are those in the neighborhood of x_k which are A_k, but now it may be the case that $A_j \cap A_k \neq \emptyset$, however since the deferral label associated to all examples is the same, which is to not defer, the loss of the elements in the intersection are not affected (in essence there is no double counting of the elements) so that now: $L_V(\{x_j, s_k\}) = |S_V| - |A_j \cup A_k|$. It is now clear to see that solving problem (11) with $\delta = 0$ finds a set cover of W with elements $A_1, \cdots, A_n$ as $L_V(D)$ simply counts how many elements of S_V (correspondence to W) we don't apply the prior rejector to (i.e. elements we cover).

□

B.4 Efficient Implementation of Greedy Selection

When $\alpha = 1$, we provide an efficient implementation of the greedy selection algorithm `GREEDY-SELECT`.

At each round, we have a teaching set D_t from which we can construct a rejector function $g_t(.,.)$, at D_0 we have g_0 is the prior rejector. Now at round t, we calculate for each example on the training set S the following quantity

$$E_i^t = \sum_{j \in S \ | \ K(z_j, z_i) \geq \gamma_i} \mathbb{I}_{g_t(z_j, a_j) \neq r_j} \tag{27}$$

E_i counts the number of points that are in the neighborhood of z_i that the current human rejector g_t misclassifies, in other words it measures for each point i how many points in the training set it will cause their deferral label to flip. Note that we are guaranteed that once a point is close enough to the teaching point z_i, it's deferral decision becomes optimal by Assumption 3. At at each round t we pick the point $i^* = \arg\max_i E_i^t$.

This algorithm has run-time $O(n^2 m)$ where $n = |S|$, while the naive implementation of the algorithm has run-time $O(n^2 m^2)$, the extra m factor comes from having to simulate the human rejector to calculate the resulting loss.

When we are optimizing over the choice of radius jointly with the choice of training point, we have no other choice but to fully simulate the human rejector. But note that the optimization over the radius can be reduced to only looking at radius choices that are equal to kernel similarities on the training set.

C SAE Model Error Analysis

Predictions. The below analysis is performed from allowing the model SAE-large model [THW$^+$20] whose code is available at [2] to predict on the HotpotQA DEV set [YQZ$^+$18] with no distractor paragraphs. The model is ranked 20'th on the public leaderboard, and is the highest ranking model with publicly available code.

C.1 Factors of difference

Presence of distractors. There are two types of question answer types in HotpotQA: yes/no answers and answers that are substrings from the passage. We eliminate yes/no questions and only focus on questions that admit an answer inside the passage which makes the validation set of size 6947 out of an original 7405. We note that the absence of distractor paragraphs does not boost performance by a significant amount. In fact the model SAE first consists of a relevant paragraph extractor that feeds into the RoBERTa reader and that extractor works quite well as evidenced.

Table 6: Performance on the dev set without yes/no questions.

Factor	Exact Match (EM)	F1
8 distractors	66.92	79.62
No distractors	68.79	82.75

Bridge vs comparison questions. The questions in HotpotQA can be categorized into two types: **bridge** e.g. "“when was the singer and songwriter of Radiohead born?”, to answer this question one first has to figure out who is the singer of Radiohead and then look up his date of birth, the other type are **comparison** questions such as that “Who has played for more NBA teams, Michael Jordan or Kobe Bryant?". This categorization is provided already in the dataset.

Table 7: Performance based on question types.

Factor	Exact Match (EM)	F1
Bridge	68.31	83.25
Comparison	71.52	79.86

We can see that there is a difference in how question types affect performance, however it is not consistent across the two metrics to make a definite conclusion.

Passage Lengths. Given the length of the two golden paragraphs, is there a difference in the performance over different sizes? As we can see below we observe no significant difference, in the last bucket of long passages we see a notable increase in F1 but that is due to limited sample size in extremely long passages.

[2] `https://github.com/JD-AI-Research-Silicon-Valley/SAE`

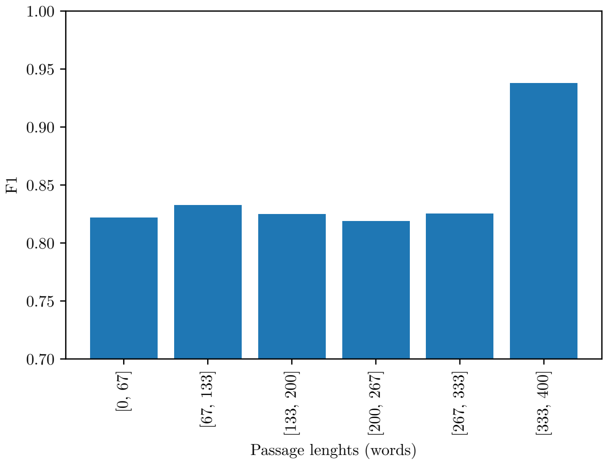


Figure 6: Performance across lengths of passages in terms of words. First bin contains very little samples to be significant.

Supporting fact lengths. We plot the performance versus the number of supporting facts: the number of sentences one must read to answer the question, this is provided in the dataset explicitly. Note there are at least two sentences that one must read since all questions are multi-hop. We can see that there is no real difference across all lengths.

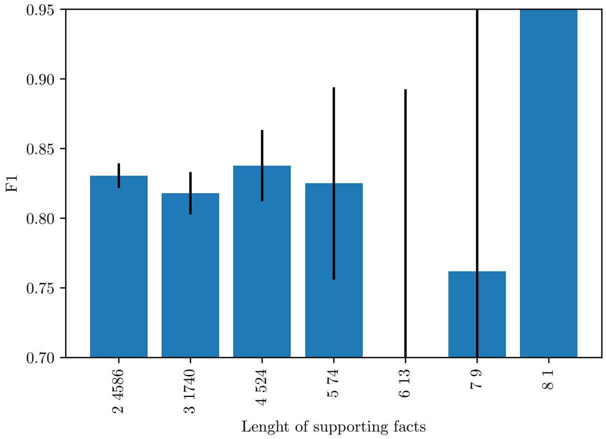


Figure 7: Performance across number of supporting sentences. Black bars indicate 95% confidence interval around the mean, the x axis is: (number of sentences, number of examples with that many sentences)

Passage and Question topics. We try to see if there is a difference in performance when looking into the topics that the examples belong to. We first run an LDA with 15 topics on the passage concatenated with the question (we use the *gensim* package [ŘS10]). We then categorize each example according to the topic with the largest coefficient in the LDA decomposition.

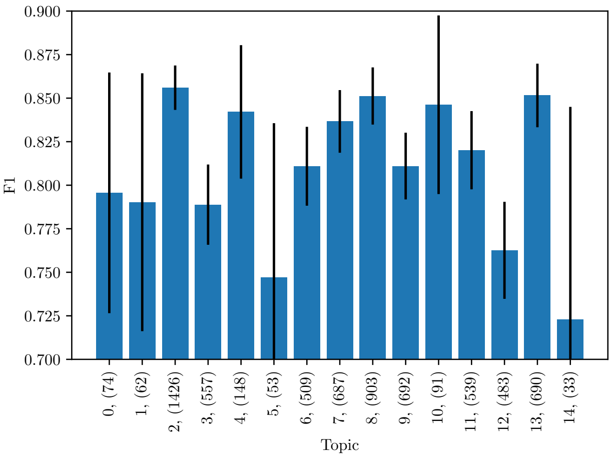


Figure 8: Performance across LDA topics

Plotted in Figure 8 are mean F1 across topic and 90% confidence intervals and we observe no particular topic that has significant difference from others.

Question words. We investigate difference in performance depending on the question word present in the question.

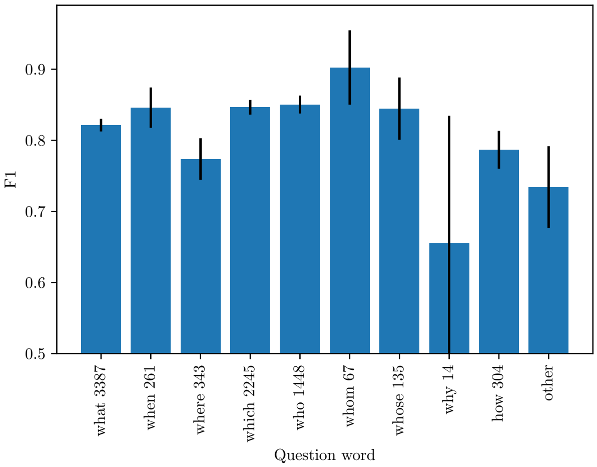


Figure 9: Performance (left EM, right F1) across question words

We can see that there is significant difference with "why" questions (however they are rare) and "how" questions to a lesser degree.

C.2 Embedding clustering

Model embeddings. The SAE model last layer consists of a 512x1024 tensor: a 1024 representation of 512 tokens. This representation is then used to predict for each token the probability that it is the start or end of the answer with a linear layer. To get a vector representation of each example, we average out across tokens to obtain a single 1024 vector for each example. We take these vectors and cluster them using K-means (we do the analysis for multiple k's). We then plot the performance across each cluster below.

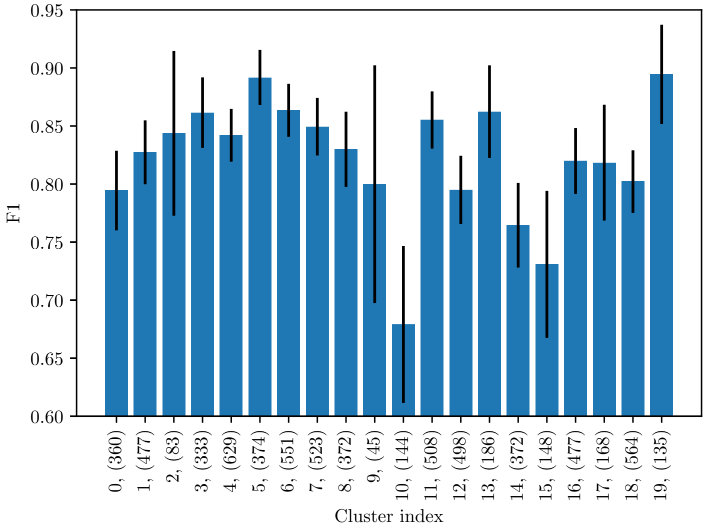


Figure 10: Performance (left EM, right F1) across model embeddings clusters.

We observe that cluster 10 has lower performance than average by a significant amount. Looking at examples from that cluster, no apparent theme emerges.

Passage embeddings. We use the BERT sentence encoder [3] to get embeddings for the passage and cluster them using k-means. We repeat the exact process for the questions and answers.

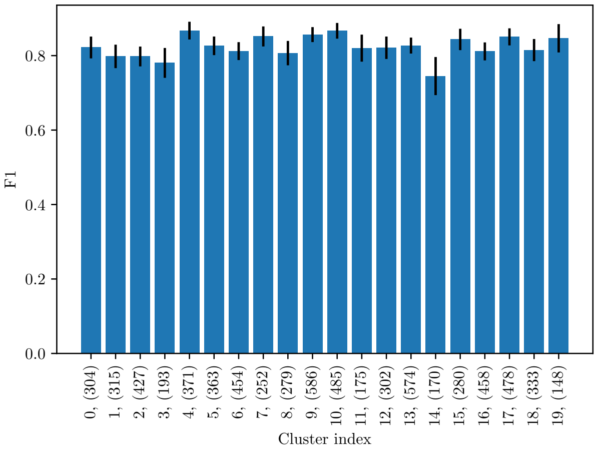


Figure 11: Performance across passage embeddings clusters. No differences emerge significantly.

[3] `https://github.com/UKPLab/sentence-transformers`

Question embeddings. We can see that cluster 13 undeperforms, examining that cluster we can see a pattern of questions like "What city does Paul Clyne and David Soares have in common?", the theme is the "in common" at the end of the question.

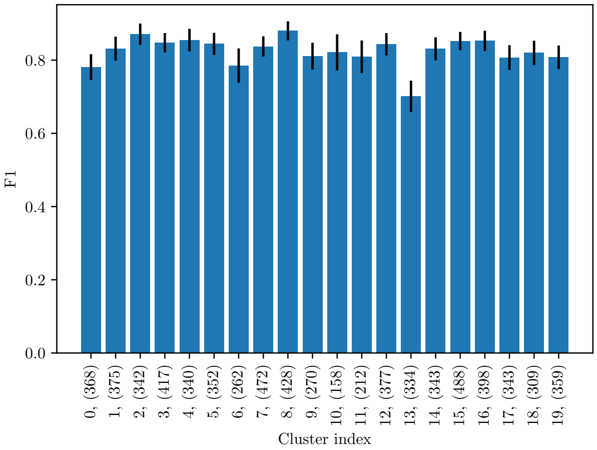


Figure 12: Performance across question embeddings clusters.

Answer embeddings. We observe no observable theme or significant differences.

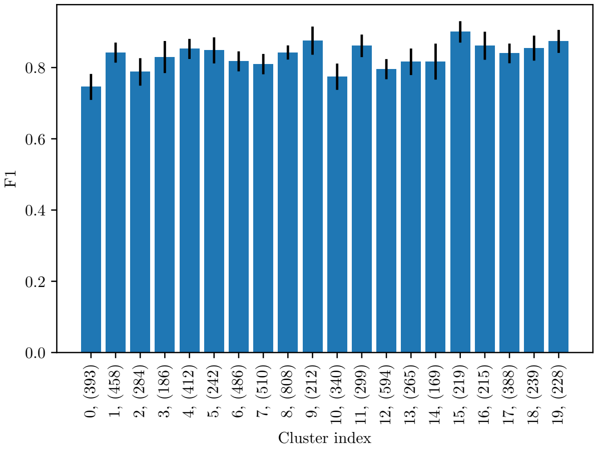


Figure 13: Performance across answer embeddings clusters.

D Synthetic Experiments Details and Results

All experiments were run on a Linux system with a NVIDIA Tesla K80 GPU, 25 GB of RAM on Python 3.7. We use the scikit-learn package to run the clustering algorithms [PVG+11], LIME package for the selection baseline [RSG16] [4], ELI5 package to obtain the text LIME highlights [5] and the Sentence Transformers package for the embedding models [RG19] [6]

D.1 Misspecification results

To evaluate how much information about h we need to properly teach the human, we learn a teaching set assuming the human's error probability is $err'_p + \boldsymbol{\delta}$ where $\boldsymbol{\delta}$ has each component drawn from $\{-\delta, \delta\}$ uniformly where $\delta > 0$. Figure 14 shows the difference to ORACLE accuracy as we increase the misspecification of the human predictor. In this experiment, we assume knowledge of the prior rejector g_0 and that the human is perfectly learning the radius given by the teaching algorithm. What this experiment impacts is the computation of the optimal deferral decision r_i computed by our algorithm to obtain S^*. At the limit when $\delta = 0.5$, we assume that the human expert error rate is uniformly 0.5 across the domain, which is the same as having the human predictions $h \sim Bin(1/2)$ on the teaching set.

In Table 1 in the paper, we evaluate what happens when the human is not learning the radius perfectly, this simulates noise in the learning process. The radius γ_i that the human learns is a noisy version of $\hat{\gamma}_i$, specifically we add a uniformly distributed noise $\delta \sim \mathcal{U}(-(1-\hat{\gamma}_i)/2, (1-\hat{\gamma}_i)/2)$.

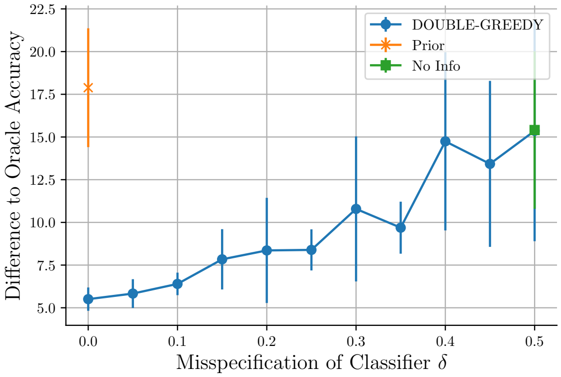


Figure 14: Difference in Oracle accuracy at teaching size @T=30 for the `DOUBLE-GREEDY` method assuming an error in h by $\boldsymbol{\delta}$ in setting B.

[4] `https://github.com/marcotcr/lime`

[5] `https://eli5.readthedocs.io/en/latest/index.html`

[6] `https://github.com/UKPLab/sentence-transformers`

E Additional Synthetic Experiments

E.1 CIFAR-10

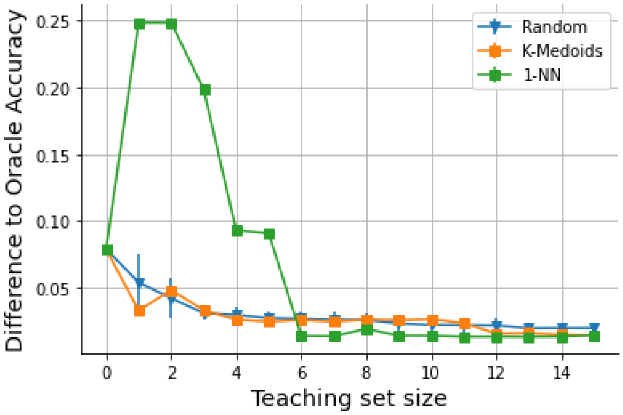


Figure 15: Comparing a 1-nearest neighbor rejector model to the radius nearest neighbor model introduced in Assumption 2 for expert $k = 6$. The "1-NN" line is obtained by first obtaining T points using K-medoids and then running a 1-NN rejector on these points with the label assigned to each point being the optimal deferral decision r_i. We can see that 1-NN struggles with less than 6 examples, but then reaches a steady state that has the same error as the radius nearest neighbor model. The effectiveness of the radius nearest neighbor model when the teaching set is very small is due to the local nature of each update with the addition of a teaching example.

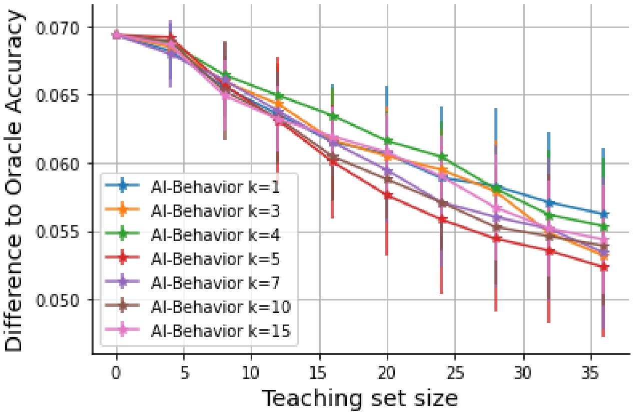
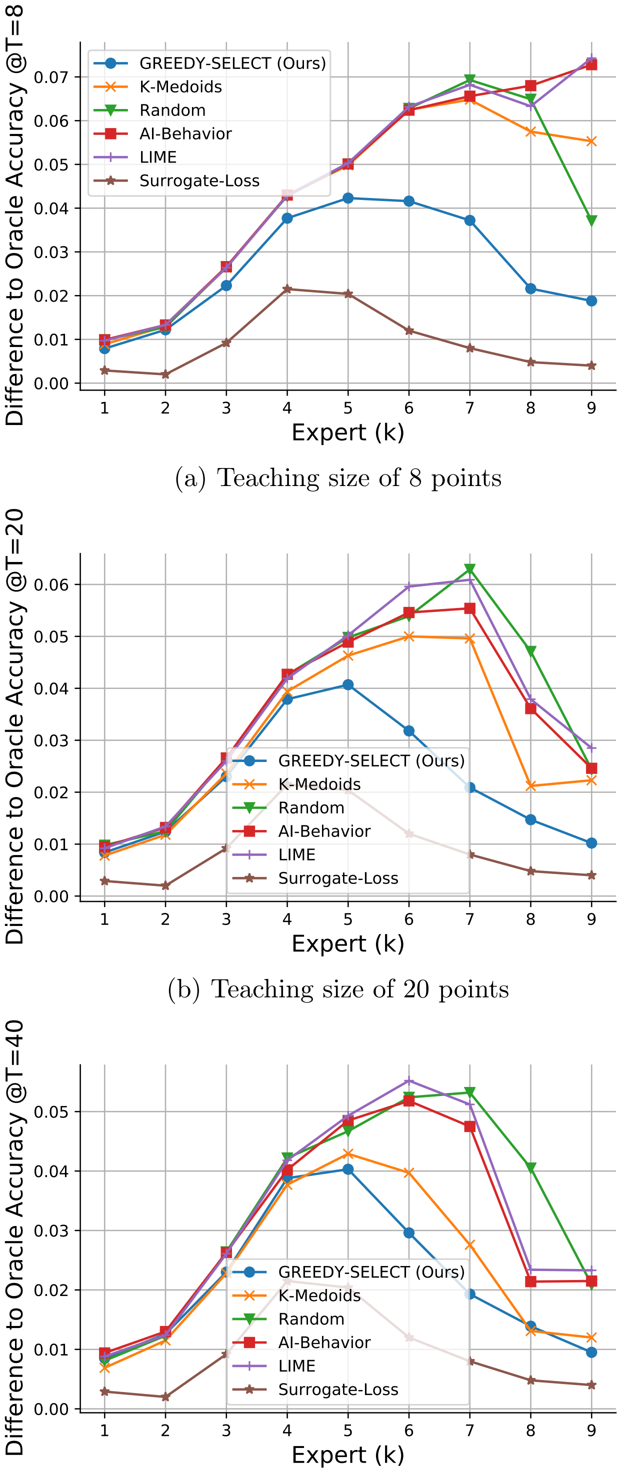


Figure 16: Performance of the AI-Behavior baseline as we vary the parameter K: the AI-Behavior baseline uses a K-nearest neighbor rejector and at each teaching step selects the point that best reduces the error of the rejector at detecting the AI's errors. We show results for the human expert $k = 6$ with the consistent radius strategy $\alpha = 1$. We can see that the parameter K has little effect and thus we use a natural choice of $K = 6$.

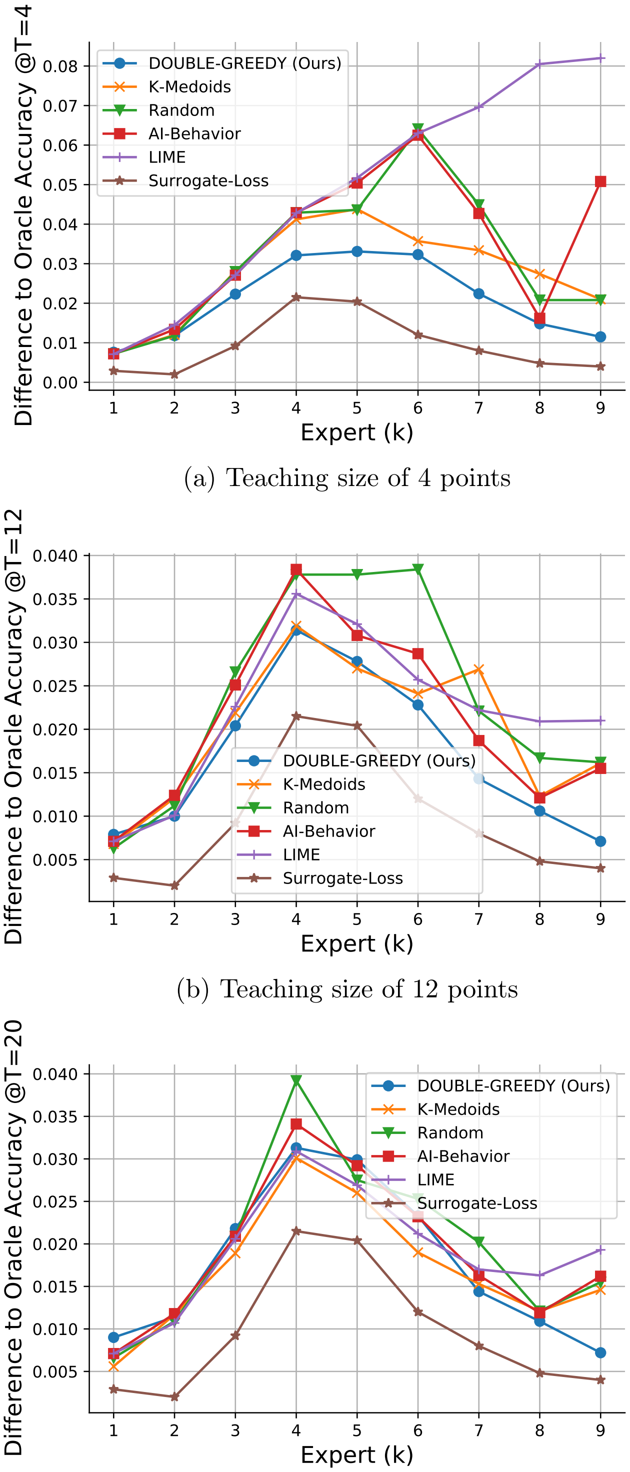


(a) Teaching size of 8 points

(b) Teaching size of 20 points

(c) Teaching size of 40 points

Figure 17: Extended legend: Varying the human parameter k (number of classes human can classify) and plotting the difference to oracle accuracy for all the baselines when using the consistent radius strategy including the surrogate-loss learning to defer method of [MS20] at 3 different teaching set sizes.



(a) Teaching size of 4 points

(b) Teaching size of 12 points

(c) Teaching size of 20 points

Figure 18: Extended legend: Varying the human parameter k (number of classes human can classify) and plotting the difference to oracle accuracy for all the baselines when using DOUBLE-GREEDY including the surrogate-loss learning to defer method of [MS20] at 3 different teaching set sizes.

E.2 Guassian Data Illustration

Figure 1 illustrates the rejector for a linear classification setting, here we formalize this as a mixtures of Gaussian setup and show the performance of our selection algorithm both quantitatively and qualitatively.

Setup. As an illustrative setting where we can visually inspect the teaching set, we perform experiments on two dimensional Gaussian mixture data. The covariate space is $\mathcal{X} = \mathbb{R}^2$ and target $\mathcal{Y} = \{0, 1\}$, we assume that there exists two sub-populations in the data denoted $A = 1$ and $A = 0$. Furthermore, $X|(Y = y, A = a)$ is normally distributed according to $\mathcal{N}(\mu_{y,a}, I)$. The group proportion is $\mathbb{P}(A = 1) = 0.5$ and the means are sampled from a uniform distribution. The AI follows the Bayes solution for group $A = 1$ which here corresponds to a hyperplane and the human classifier follows the Bayes solution for group $A = 0$, which is another hyperplane. We assume the human's prior rejector is to reject based on a tresholding of the predictor confidence i.e. $g_0(x) = \mathbb{I}\{||h(x)|| \leq \epsilon\}$. We assume that the similarity kernel is the RBF kernel $K(x, x') = e^{-||x-x'||^2}$.

Results. For 100 trials, we generate data with random means and measure the difference in system accuracy between our approach and the baselines as we vary the size of the teaching set. Results are shown in Figure 19. Figure 20 shows the points chosen on a given configuration.

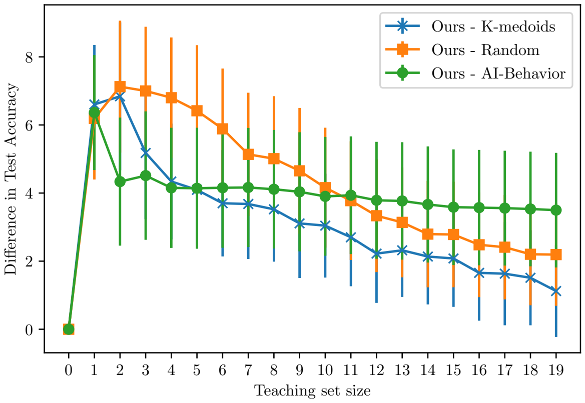
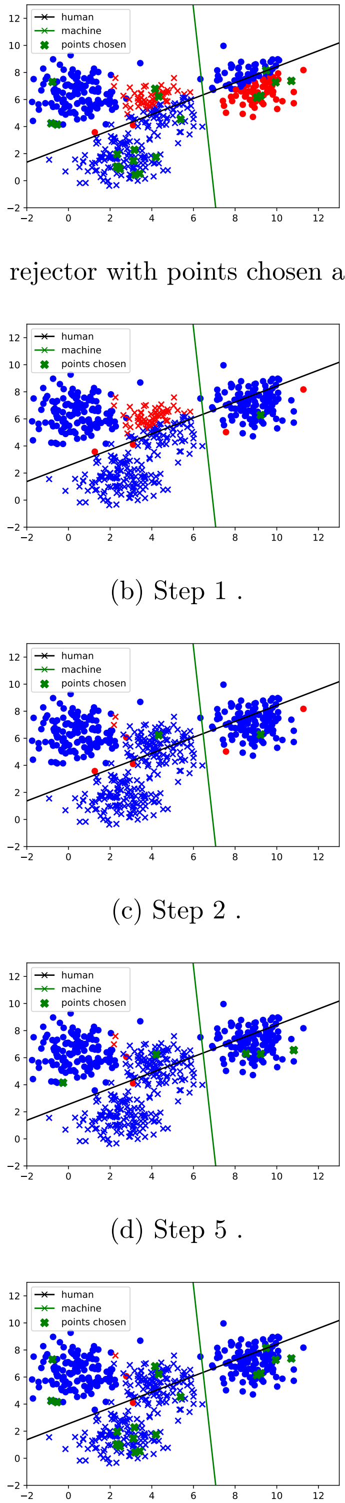


Figure 19: Teaching complexity plot for synthetic Gaussian data setup. The x-axis shows the difference in test human accuracy between our method and the baselines. Plotted are the averages over the 100 trials along with 95% confidence interval error bars for the average.



(a) Prior rejector with points chosen at step 20.

(b) Step 1 .

(c) Step 2 .

(d) Step 5 .

(e) Step 20 .

Figure 20: Extended legend: blue dots indicate a correct decision while red dots indicate mistakes. Points with an "x" are labels 1 while points with an "o" are labels 0 (in the Y space). The lines labeled human and machine are the respective classifiers.

F Crowdsourced Experiments Details and Results

F.1 Experiment Details

Participants. We recruited 50 US based participants from Amazon Mechanical Turk per each condition (100 total), workers were required to have a HIT approval rate higher than 95% and over 100 HITs approved. Initial pilot studies were also conducted with graduate students in computer science at a US university. Participants in the baseline were paid \$3 for 10 minutes of work and those in the teaching condition received \$6 for 20 minutes of work. Any demographic information we gathered in our study is kept confidential and workers were asked to consent to their use of their responses in research studies. We submitted an IRB application and the IRB declared it exempt as is. We followed standard protocol and additionally provided the IRB exemption and details to our user study participants. We filter participants who don't answer the tutorial questions correctly and we also filter for all baselines that workers at least answer one question correctly on their own beyond the first question.

AI and Test Set details. The simulated AI used in the study was obtained by first performing K-means with $K = 25$ on the dev set of HotpotQA, and then manually filtering the data to obtain 11 clusters that are more distinct. The test set used in the testing phase was obtained first by filtering the data using K-medoids with $K = 200$ as a way to get diverse questions. We then created 20 test sets by sampling 7 random questions from the filtered set on which the AI was correct and 8 on which the AI is incorrect. The order of the examples in the test set was shuffled for each participant.

Cluster Topics. The AI used in the study had 11 different clusters on which it's errors were defined. Table 8 shows the main theme and most common Wikipedia categories for each cluster.

User Lessons. In Table 9 we show examples of the lessons that the crowdworkers wrote during the teaching phase for the proposed teaching method. We show examples of the lessons on the first 3 examples in the teaching phase and separate the participant lessons into 4 categories: participants who wrote accurate lessons, participants who wrote irrelevant lessons (not relevant to the question or required no effort to write), participants who wrote complex lessons that don't pertain to the example topic and finally participants who wrote narrow lessons that are on topic but only apply to the example and not the neighborhood of the example. In Table 2 we separated user metrics into two groups accurate lessons and inaccurate lessons, this corresponds to grouping accurate lessons versus the rest in the lesson categorization of Table 9. Furthermore, in the body of section 6.4 we distinguish between accurate lessons, narrow and complex lessons (combined into one group) and finally irrelevant lessons.

G Extended Discussion

One limitation of our human experiments is that we used a simulated AI that has an easier to understand error boundary. This enabled us to have a more in-depth study of the crowdworker responses than otherwise would have been possible.

Having a simulated AI to which we perfectly understand where it's error regions are (but note this is highly non trivial for someone who doesn't know how it was trained), enables to define what

Table 8: Cluster main theme (manually obtained) and top 3 Wikipedia categories of examples in clusters for the AI used in the MTurk study.

Cluster ID	Main Theme	Wikipedia Categories
1	Plants	Poaceae genera, Flora of Mexico, Dioecious plants
2	Singers, Musicians	21st-century American singers, Grammy Award winners, American male guitarists
3	Movies, Actors	American films, British films, American male film actors
4	Sites, Hotels	Casino hotels, Casinos in the Las Vegas Valley, Resorts in the Las Vegas Valley
5	Writers, Magazines	20'th-century American novelists, American male non-fiction writers, American women novelists
6	Composers, Plays	19th-century classical composers, Operas, Male classical pianists
7	Games	Windows games, PlayStation 4 games, Xbox One games
8	Universities	Universities and colleges, Colonial colleges, Private universities in New York
9	Soccer	Premier League players, English Football League players, Association football midfielders
10	Sports (non soccer)	American men's basketball players, NFL player, NBA All-Stars
11	Politics	21st-century American politicians, Presidential Medal of Freedom recipients, Republican Party members

the "lessons" should be and thus evaluate if users are learning correctly. This ability to evaluate if users are actually learning through their written lessons enables to test two things:

1. Do people learn the correct lessons using our teaching method?

2. Do those who learn the correct lessons apply them perfectly?

And our answers in our paper to these questions are: 1) yes but only half the people are able to, 2) not quite, since even those with perfect lessons don't show perfect accuracy (in Defer F1). What is interesting about this last observation tell us that even if people know the rules, and have them written and shown on the screen, they might still apply it incorrectly. With a non simulated AI, it would have been difficult for us to figure out the answers to the questions as the underlying lessons are not pre-determined. For an initial experimental study on teaching, we need to understand better how do humans make decisions and how we can try to use their lessons to possibly provide feedback and better guide them.

Another limitation is that our test-time interface did not include model explanations or predictions. This was done for multiple reasons:

- The AI predictions and explanations reveal information about it's underlying performance at test time. If two different crowdworkers received different test sets, then their knowledge about the AI may be different. Therefore if the participants belonged to two different experimentation conditions, then the test set becomes a confounding factor we need to control for.

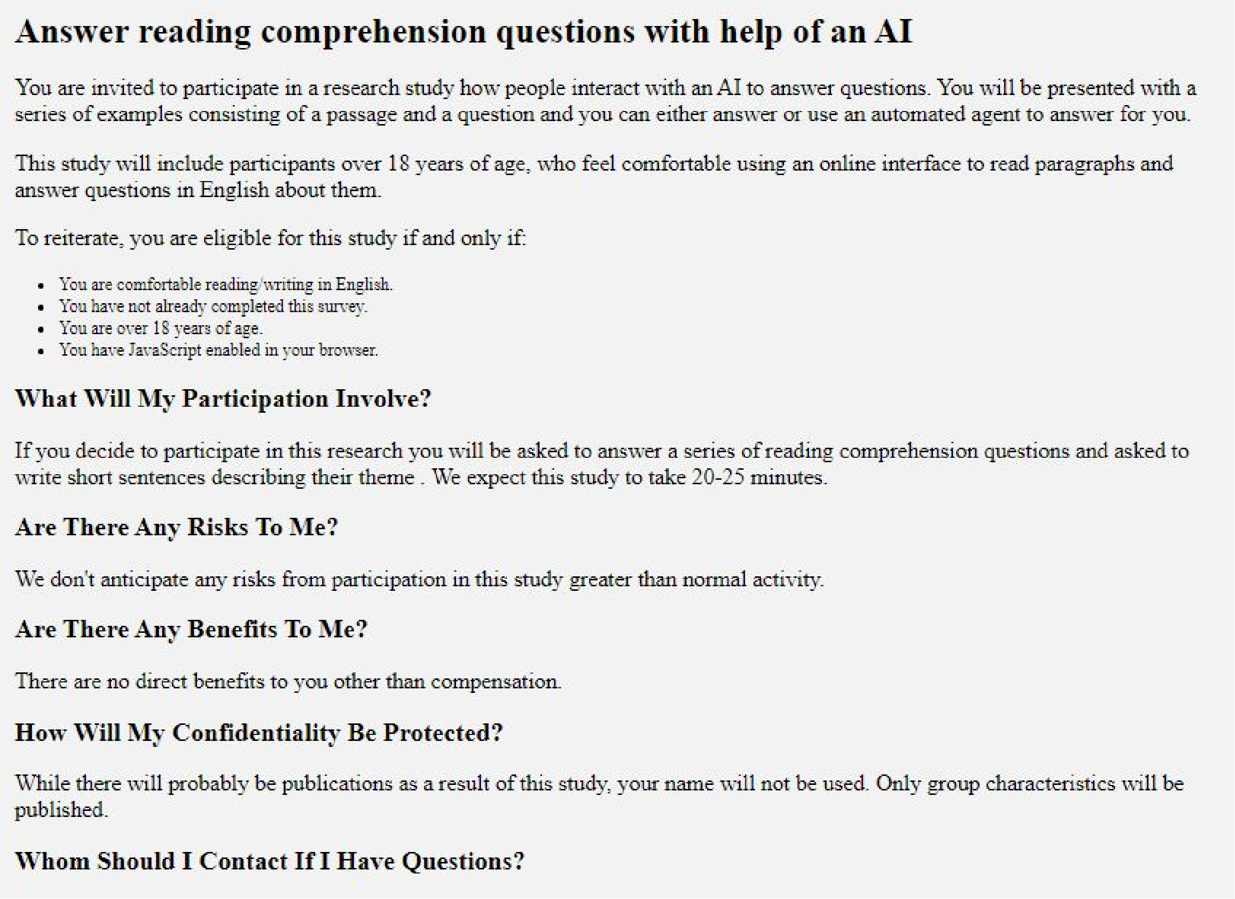
- When model explanations are not available or are not effective, the effect of teaching becomes more important as it is the only way the human's mental model is formed. Thus the choice of the teaching method becomes more important.

- If the AI prediction is available at test time and workers press on the "Use AI answer" button, there is an unobservability issue that arises: are workers pressing the button because they trust the AI, or are workers pressing the button because they came up with the same answer on their own? Removing their ability to see the AI prediction alleviates the problem.

Table 9: Example of lessons that users in the Ours-Teaching condition wrote during the teaching phase. We show examples of the lessons on the first 3 examples in the teaching phase and separate the participant lessons into 4 categories: participants who wrote accurate lessons, participants who wrote irrelevant lessons (not relevant to the question or required no effort to write), participants who wrote complex lessons that don't pertain to the example topic and finally participants who wrote narrow lessons that are on topic but only apply to the example and not the neighborhood of the example.

Lesson Type	Example ID	Actual Lesson
Accurate Lessons	1	The AI is not good at answering questions about plants.
Accurate Lessons	2	The AI is better at Politics and geography than at sports.
Accurate Lessons	3	The AI is bad at answering questions about movies
Irrelevant Lessons	1	I understood AI is good at answering
Irrelevant Lessons	2	AI focus on the institution
Irrelevant Lessons	3	AI omitted important terms
Complex Lessons	1	It seems to be better at answering questions where the absolute same phrases are used in the question as the passage and where both answers are in the question, maybe?
Complex Lessons	2	The ai is good at answering questions that has to do with cities and numbers though not good with words that has to do with repeated words.
Complex Lessons	3	The AI can't decipher clues, example, the other movie was based on a book that came out after the other movie but the AI couldn't figure out that that must mean the movie based on that book must then also have come out after the other movie.
Narrow Lessons	1	The AI isn't good at multi-faceted questions about continental species.
Narrow Lessons	2	The topic was politics and the AI is good at answering questions about specific areas when the question can be answered by looking for specific information about one section but not when it involves integrating multiple pieces of information from the paragraph.
Narrow Lessons	3	The AI isn't good at comparing media release dates..

H User Interface Screenshots



Answer reading comprehension questions with help of an AI

You are invited to participate in a research study how people interact with an AI to answer questions. You will be presented with a series of examples consisting of a passage and a question and you can either answer or use an automated agent to answer for you.

This study will include participants over 18 years of age, who feel comfortable using an online interface to read paragraphs and answer questions in English about them.

To reiterate, you are eligible for this study if and only if:

- You are comfortable reading/writing in English.
- You have not already completed this survey.
- You are over 18 years of age.
- You have JavaScript enabled in your browser.

What Will My Participation Involve?

If you decide to participate in this research you will be asked to answer a series of reading comprehension questions and asked to write short sentences describing their theme . We expect this study to take 20-25 minutes.

Are There Any Risks To Me?

We don't anticipate any risks from participation in this study greater than normal activity.

Are There Any Benefits To Me?

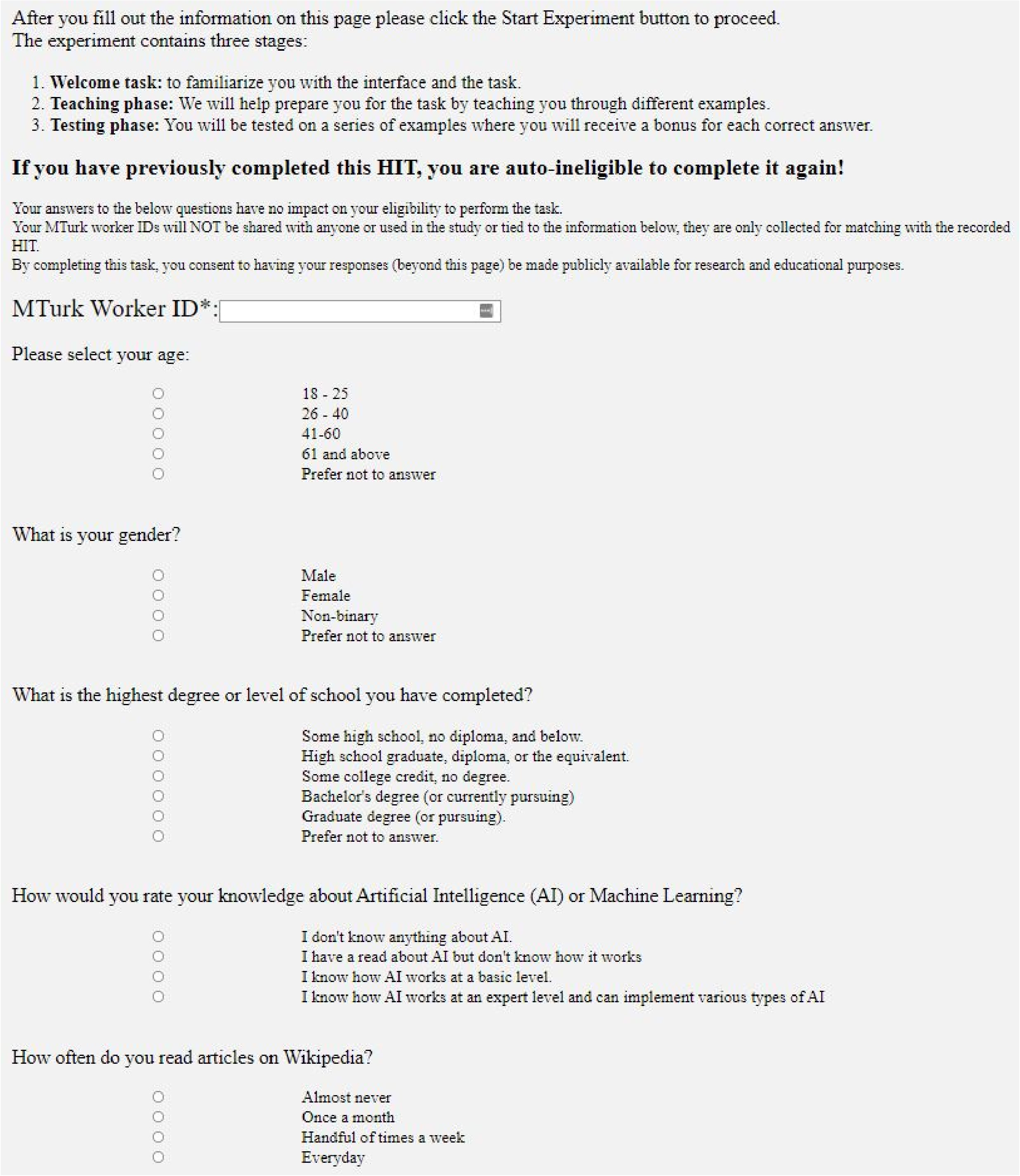
There are no direct benefits to you other than compensation.

How Will My Confidentiality Be Protected?

While there will probably be publications as a result of this study, your name will not be used. Only group characteristics will be published.

Whom Should I Contact If I Have Questions?

Figure 21: Consent form to be confirmed before entering experiment



After you fill out the information on this page please click the Start Experiment button to proceed.
The experiment contains three stages:

1. **Welcome task:** to familiarize you with the interface and the task.
2. **Teaching phase:** We will help prepare you for the task by teaching you through different examples.
3. **Testing phase:** You will be tested on a series of examples where you will receive a bonus for each correct answer.

If you have previously completed this HIT, you are auto-ineligible to complete it again!

Your answers to the below questions have no impact on your eligibility to perform the task.
Your MTurk worker IDs will NOT be shared with anyone or used in the study or tied to the information below, they are only collected for matching with the recorded HIT.
By completing this task, you consent to having your responses (beyond this page) be made publicly available for research and educational purposes.

MTurk Worker ID*:

Please select your age:

○ 18 - 25
○ 26 - 40
○ 41-60
○ 61 and above
○ Prefer not to answer

What is your gender?

○ Male
○ Female
○ Non-binary
○ Prefer not to answer

What is the highest degree or level of school you have completed?

○ Some high school, no diploma, and below.
○ High school graduate, diploma, or the equivalent.
○ Some college credit, no degree.
○ Bachelor's degree (or currently pursuing)
○ Graduate degree (or pursuing).
○ Prefer not to answer.

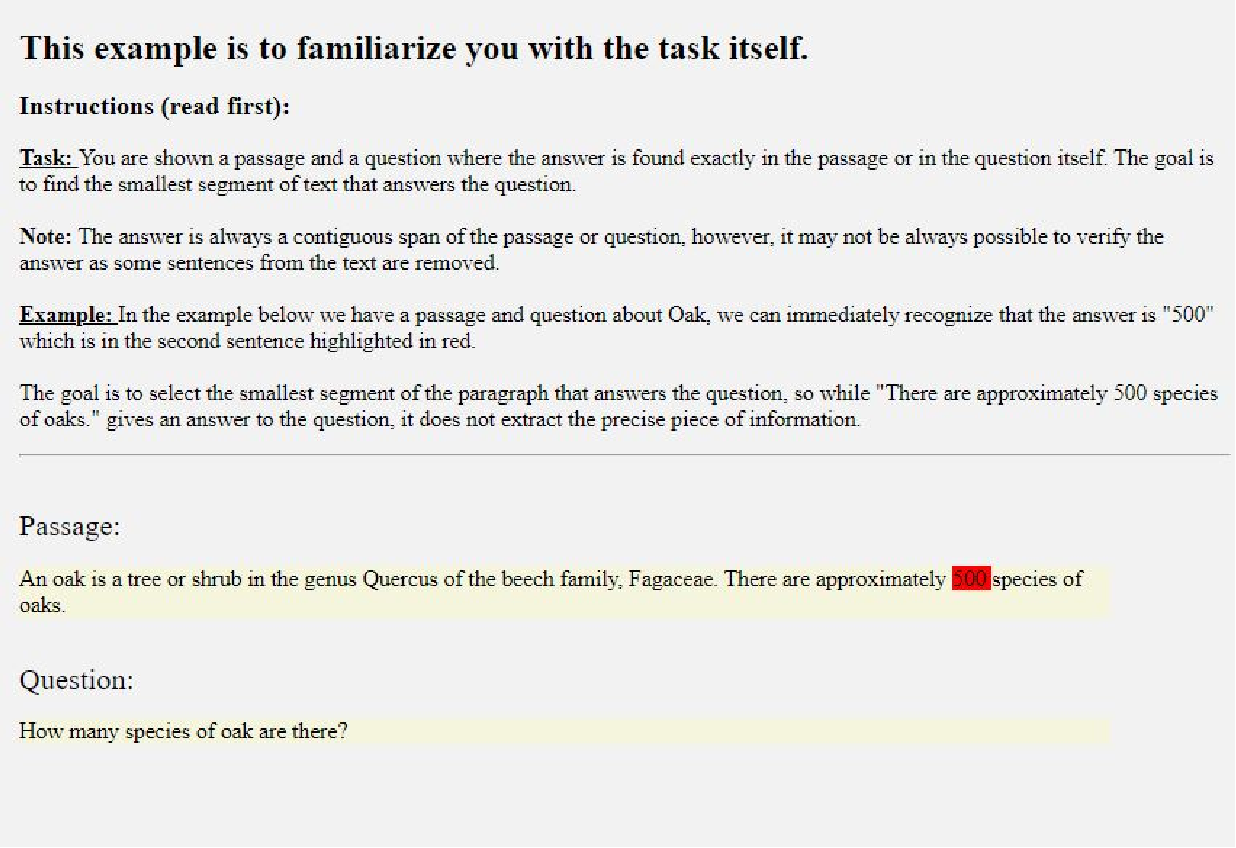
How would you rate your knowledge about Artificial Intelligence (AI) or Machine Learning?

○ I don't know anything about AI.
○ I have a read about AI but don't know how it works
○ I know how AI works at a basic level.
○ I know how AI works at an expert level and can implement various types of AI

How often do you read articles on Wikipedia?

○ Almost never
○ Once a month
○ Handful of times a week
○ Everyday

Figure 22: Information collected about workers prior to experiment. MTurk worker ID was only saved for cross-checking and then deleted.



This example is to familiarize you with the task itself.

Instructions (read first):

Task: You are shown a passage and a question where the answer is found exactly in the passage or in the question itself. The goal is to find the smallest segment of text that answers the question.

Note: The answer is always a contiguous span of the passage or question, however, it may not be always possible to verify the answer as some sentences from the text are removed.

Example: In the example below we have a passage and question about Oak, we can immediately recognize that the answer is "500" which is in the second sentence highlighted in red.

The goal is to select the smallest segment of the paragraph that answers the question, so while "There are approximately 500 species of oaks." gives an answer to the question, it does not extract the precise piece of information.

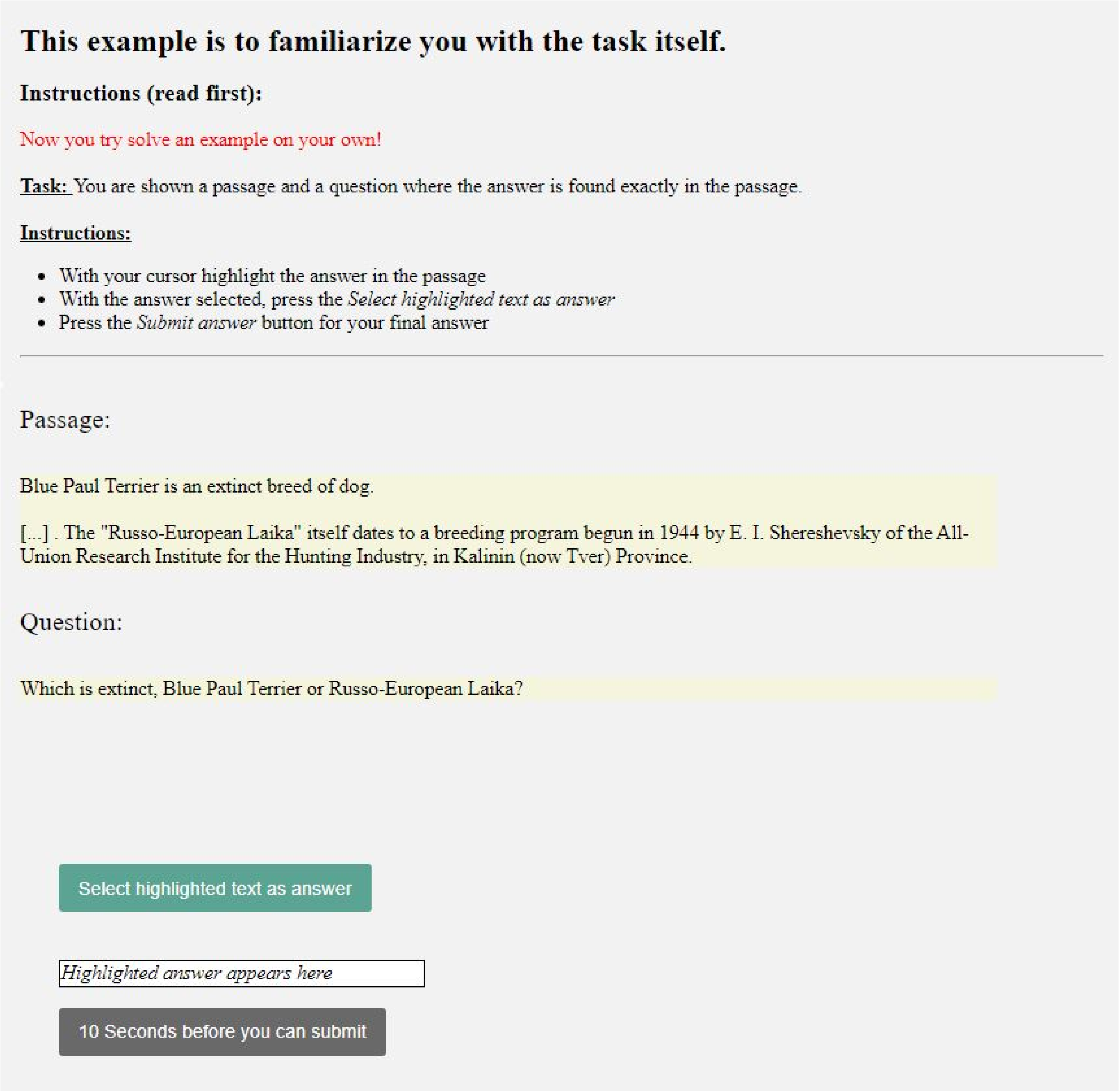
Passage:

An oak is a tree or shrub in the genus Quercus of the beech family, Fagaceae. There are approximately 500 species of oaks.

Question:

How many species of oak are there?

Figure 23: First step of the tutorial introducing the task



This example is to familiarize you with the task itself.

Instructions (read first):

Now you try solve an example on your own!

Task: You are shown a passage and a question where the answer is found exactly in the passage.

Instructions:

- With your cursor highlight the answer in the passage
- With the answer selected, press the *Select highlighted text as answer*
- Press the *Submit answer* button for your final answer

Passage:

Blue Paul Terrier is an extinct breed of dog.

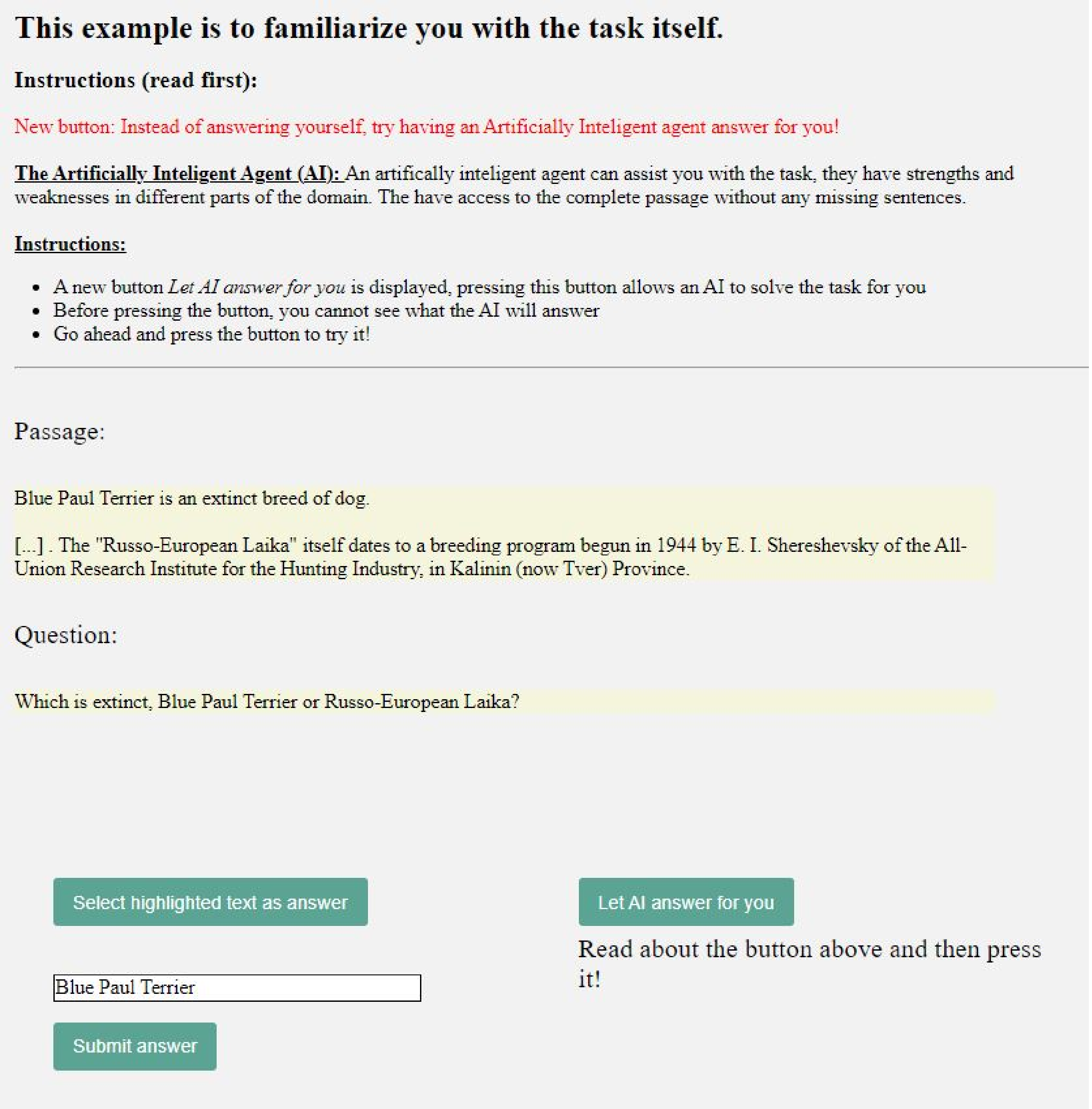
[...] . The "Russo-European Laika" itself dates to a breeding program begun in 1944 by E. I. Shereshevsky of the All-Union Research Institute for the Hunting Industry, in Kalinin (now Tver) Province.

Question:

Which is extinct, Blue Paul Terrier or Russo-European Laika?

Select highlighted text as answer
Highlighted answer appears here
10 Seconds before you can submit

Figure 24: Second step of the tutorial solving without AI help



This example is to familiarize you with the task itself.

Instructions (read first):

New button: Instead of answering yourself, try having an Artificially Inteligent agent answer for you!

The Artificially Inteligent Agent (AI): An artifically inteligent agent can assist you with the task, they have strengths and weaknesses in different parts of the domain. The have access to the complete passage without any missing sentences.

Instructions:

- A new button *Let AI answer for you* is displayed, pressing this button allows an AI to solve the task for you
- Before pressing the button, you cannot see what the AI will answer
- Go ahead and press the button to try it!

Passage:

Blue Paul Terrier is an extinct breed of dog.

[...] . The "Russo-European Laika" itself dates to a breeding program begun in 1944 by E. I. Shereshevsky of the All-Union Research Institute for the Hunting Industry, in Kalinin (now Tver) Province.

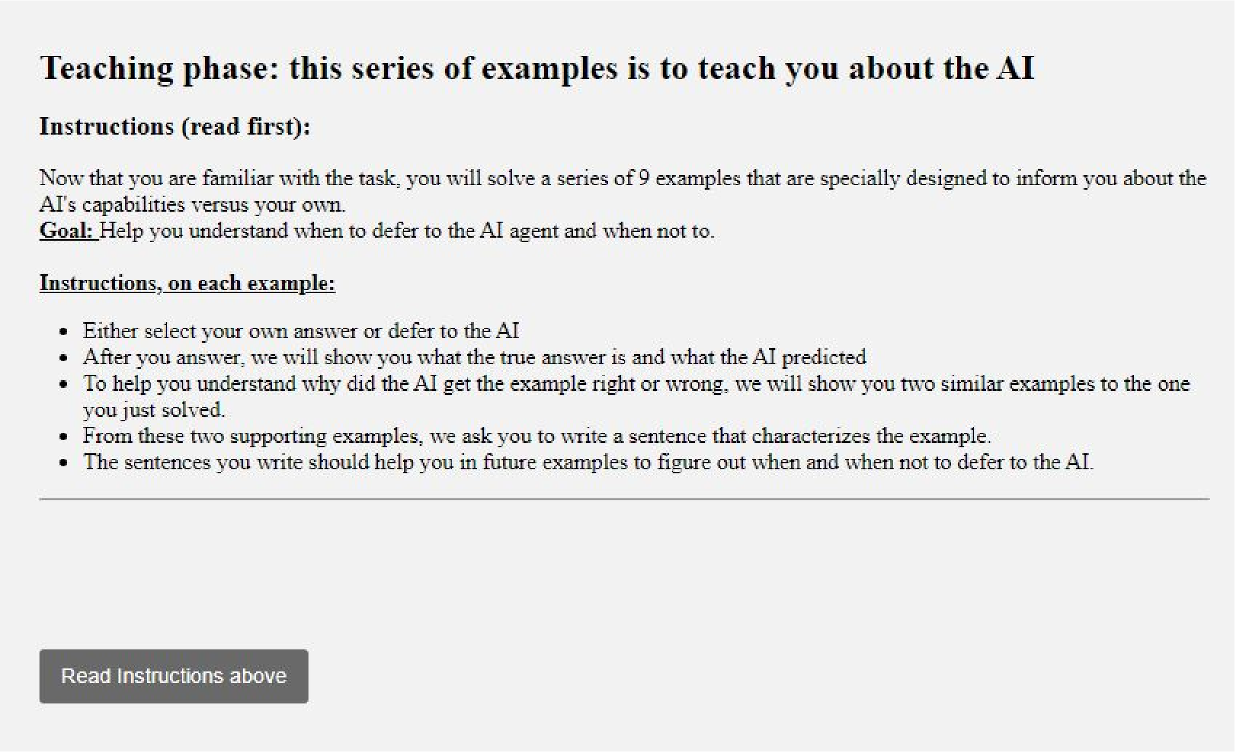
Question:

Which is extinct, Blue Paul Terrier or Russo-European Laika?

Blue Paul Terrier

Read about the button above and then press it!

Figure 25: Third step of the tutorial solving with AI help



Teaching phase: this series of examples is to teach you about the AI

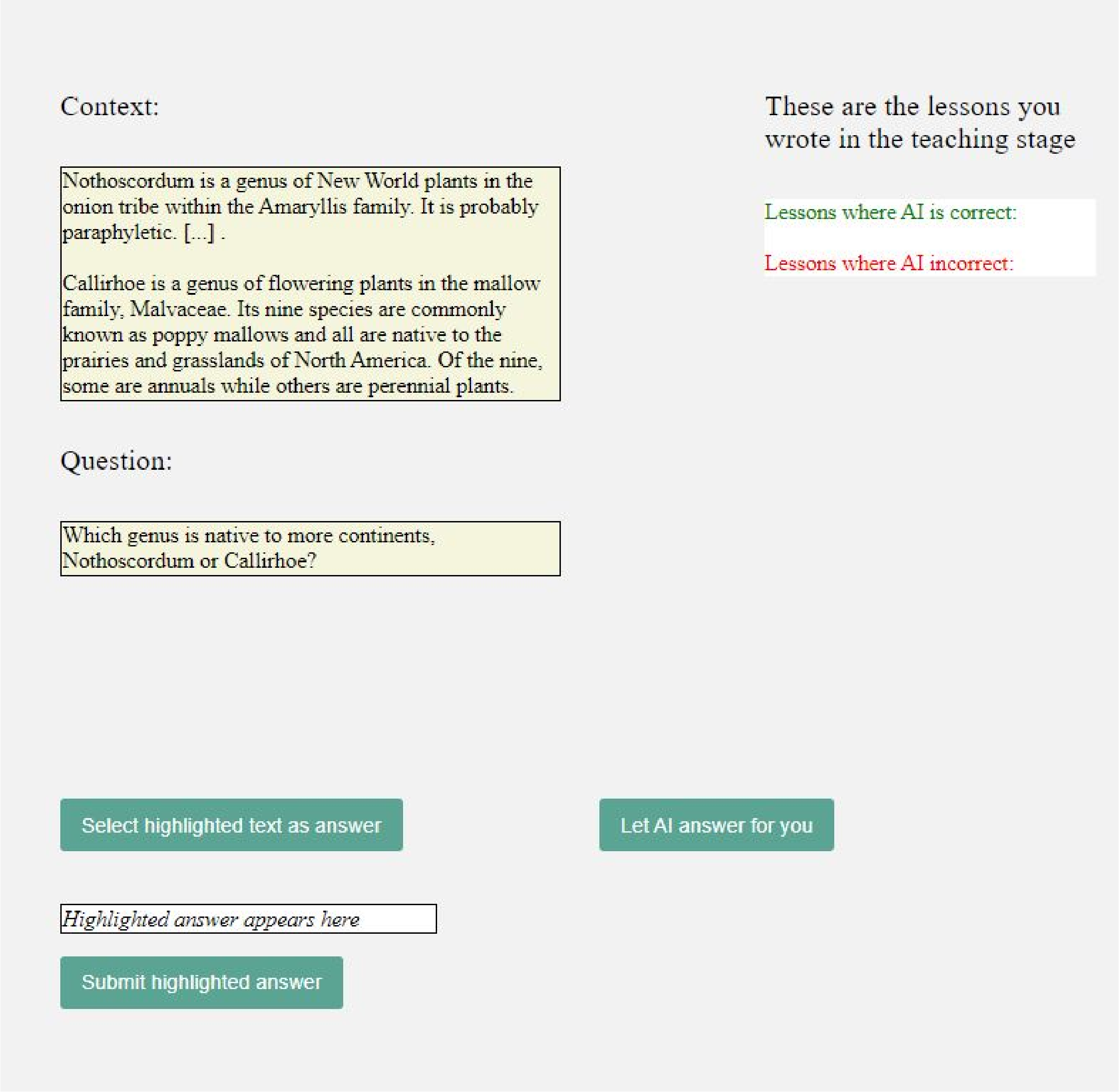
Instructions (read first):

Now that you are familiar with the task, you will solve a series of 9 examples that are specially designed to inform you about the AI's capabilities versus your own.
Goal: Help you understand when to defer to the AI agent and when not to.

Instructions, on each example:

- Either select your own answer or defer to the AI
- After you answer, we will show you what the true answer is and what the AI predicted
- To help you understand why did the AI get the example right or wrong, we will show you two similar examples to the one you just solved.
- From these two supporting examples, we ask you to write a sentence that characterizes the example.
- The sentences you write should help you in future examples to figure out when and when not to defer to the AI.

Figure 26: Teaching instructions



Context:

Nothoscordum is a genus of New World plants in the onion tribe within the Amaryllis family. It is probably paraphyletic. [...] .

Callirhoe is a genus of flowering plants in the mallow family, Malvaceae. Its nine species are commonly known as poppy mallows and all are native to the prairies and grasslands of North America. Of the nine, some are annuals while others are perennial plants.

Question:

Which genus is native to more continents, Nothoscordum or Callirhoe?

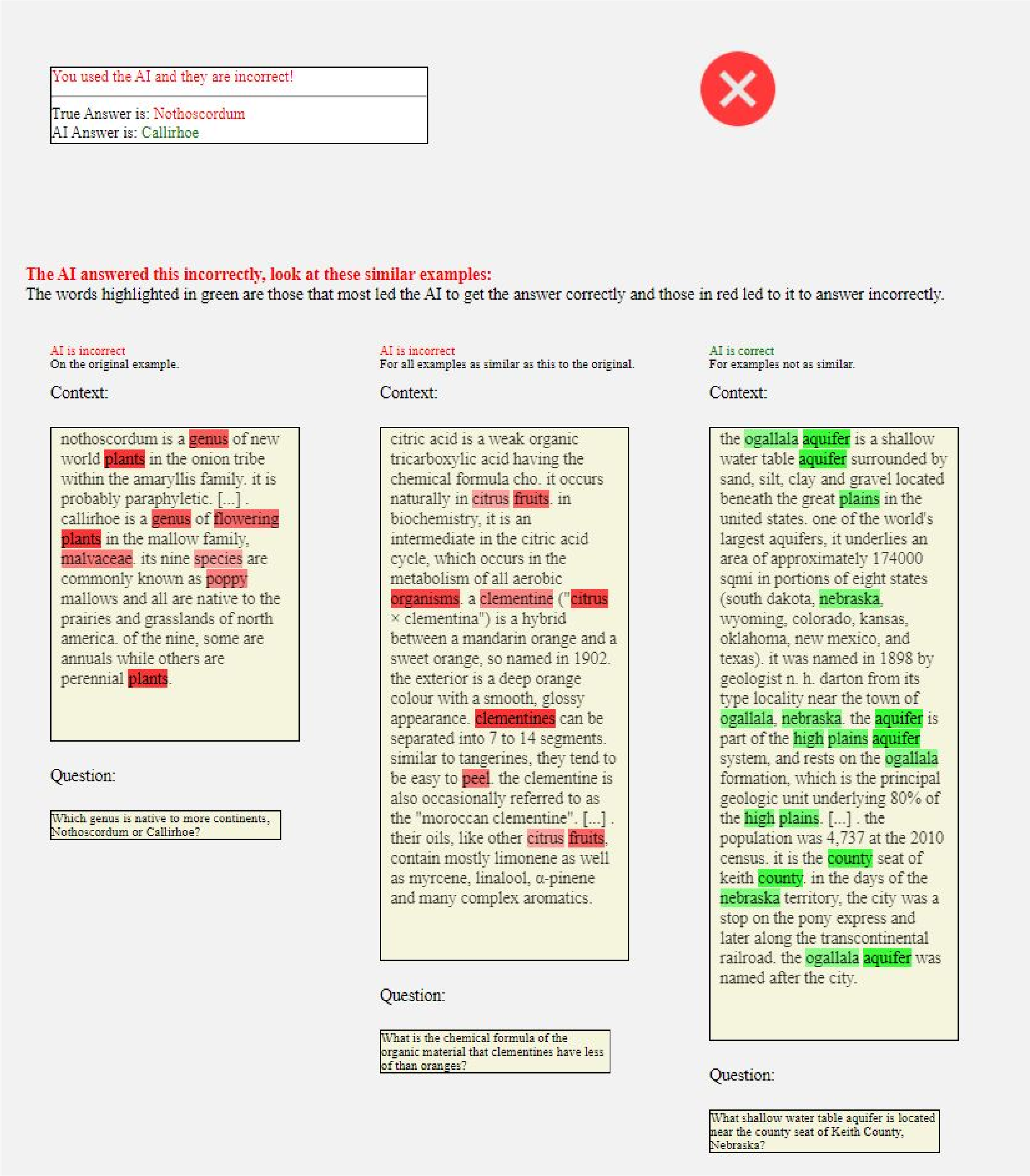
These are the lessons you wrote in the teaching stage

Lessons where AI is correct:

Lessons where AI incorrect:

Select highlighted text as answer
Let AI answer for you
Highlighted answer appears here
Submit highlighted answer

Figure 27: Teaching initial example to be solved by the human.



You used the AI and they are incorrect!

True Answer is: Nothoscordum
AI Answer is: Callirhoe

The AI answered this incorrectly, look at these similar examples:
The words highlighted in green are those that most led the AI to get the answer correctly and those in red led to it to answer incorrectly.

AI is incorrect
On the original example.

Context:

nothoscordum is a genus of new world plants in the onion tribe within the amaryllis family. it is probably paraphyletic. [...] . callirhoe is a genus of flowering plants in the mallow family, malvaceae. its nine species are commonly known as poppy mallows and all are native to the prairies and grasslands of north america. of the nine, some are annuals while others are perennial plants.

Question:

Which genus is native to more continents, Nothoscordum or Callirhoe?

AI is incorrect
For all examples as similar as this to the original.

Context:

citric acid is a weak organic tricarboxylic acid having the chemical formula cho. it occurs naturally in citrus fruits. in biochemistry, it is an intermediate in the citric acid cycle, which occurs in the metabolism of all aerobic organisms. a clementine ("citrus × clementina") is a hybrid between a mandarin orange and a sweet orange, so named in 1902. the exterior is a deep orange colour with a smooth, glossy appearance. clementines can be separated into 7 to 14 segments. similar to tangerines, they tend to be easy to peel. the clementine is also occasionally referred to as the "moroccan clementine". [...] . their oils, like other citrus fruits, contain mostly limonene as well as myrcene, linalool, α-pinene and many complex aromatics.

Question:

What is the chemical formula of the organic material that clementines have less of than oranges?

AI is correct
For examples not as similar.

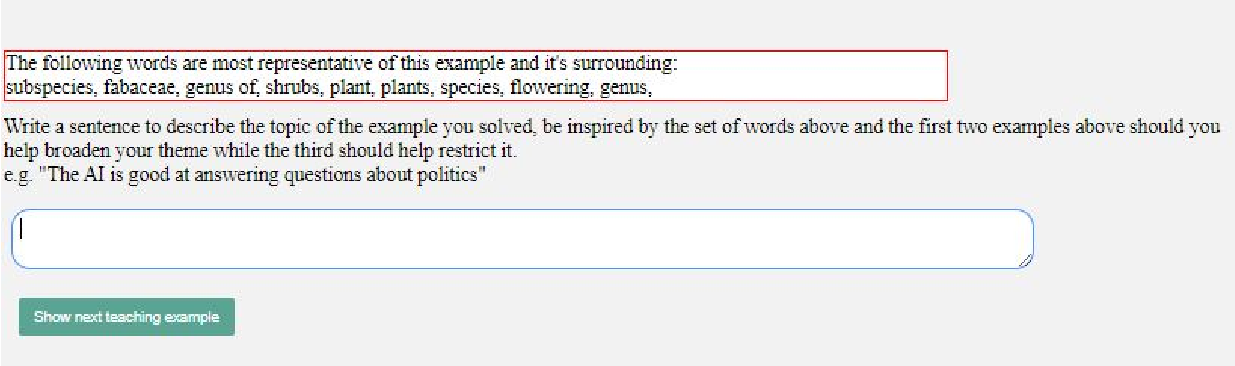
Context:

the ogallala aquifer is a shallow water table aquifer surrounded by sand, silt, clay and gravel located beneath the great plains in the united states. one of the world's largest aquifers, it underlies an area of approximately 174000 sqmi in portions of eight states (south dakota, nebraska, wyoming, colorado, kansas, oklahoma, new mexico, and texas). it was named in 1898 by geologist n. h. darton from its type locality near the town of ogallala, nebraska. the aquifer is part of the high plains aquifer system, and rests on the ogallala formation, which is the principal geologic unit underlying 80% of the high plains. [...] . the population was 4,737 at the 2010 census. it is the county seat of keith county. in the days of the nebraska territory, the city was a stop on the pony express and later along the transcontinental railroad. the ogallala aquifer was named after the city.

Question:

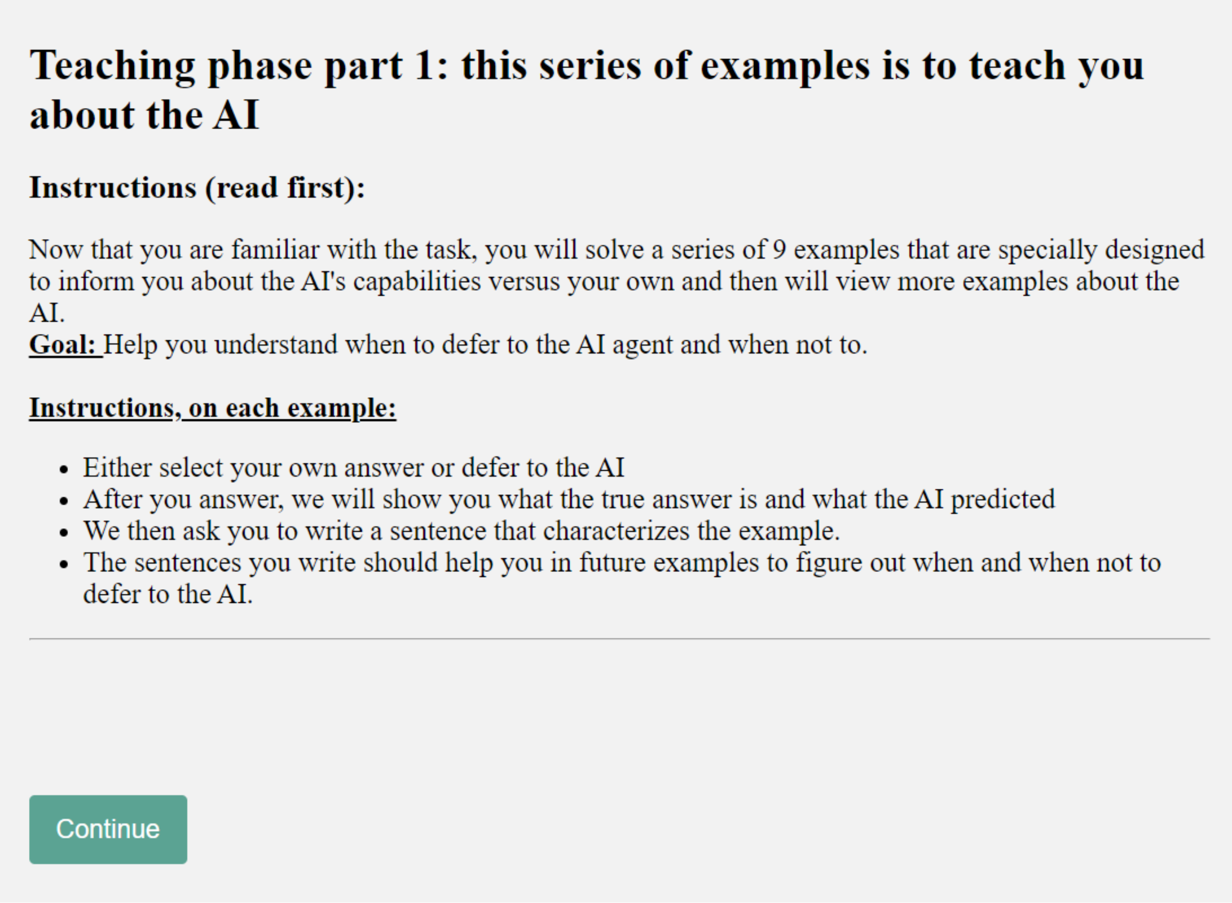
What shallow water table aquifer is located near the county seat of Keith County, Nebraska?

Figure 28: Feedback shown after human solves the example along with supporting examples.


The following words are most representative of this example and it's surrounding:
subspecies, fabaceae, genus of, shrubs, plant, plants, species, flowering, genus,

Write a sentence to describe the topic of the example you solved, be inspired by the set of words above and the first two examples above should you help broaden your theme while the third should help restrict it.
e.g. "The AI is good at answering questions about politics"

Figure 29: Top words for the teaching example along with instructions for lesson writing


Teaching phase part 1: this series of examples is to teach you about the AI

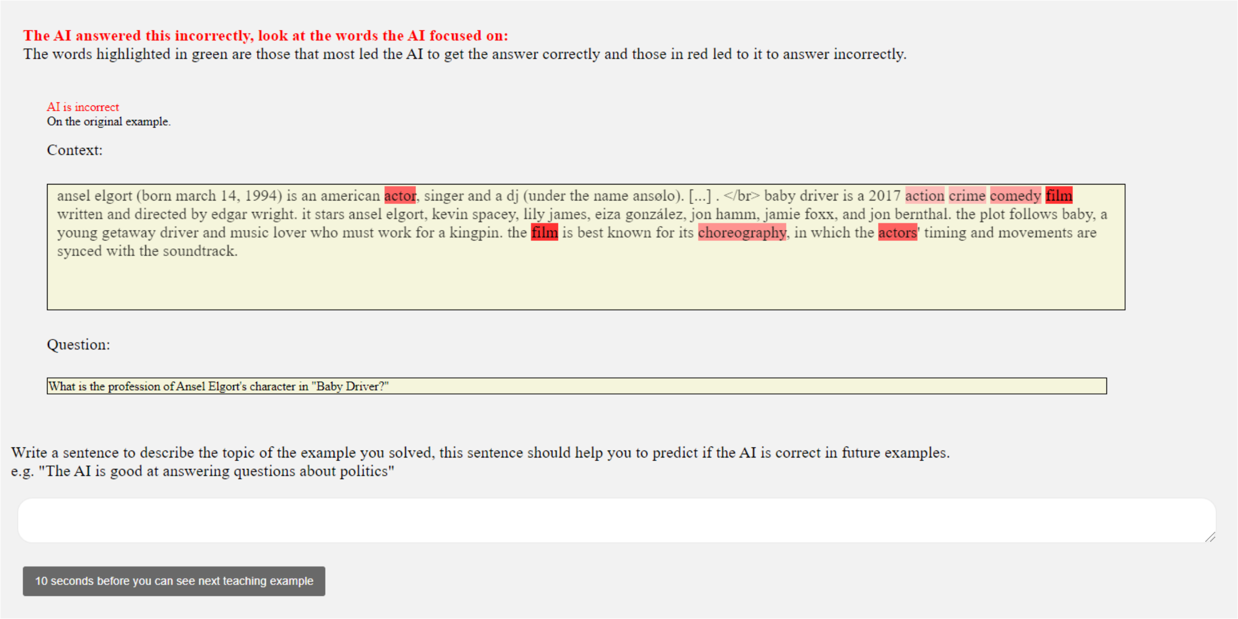
Instructions (read first):

Now that you are familiar with the task, you will solve a series of 9 examples that are specially designed to inform you about the AI's capabilities versus your own and then will view more examples about the AI.
Goal: Help you understand when to defer to the AI agent and when not to.

Instructions, on each example:

- Either select your own answer or defer to the AI
- After you answer, we will show you what the true answer is and what the AI predicted
- We then ask you to write a sentence that characterizes the example.
- The sentences you write should help you in future examples to figure out when and when not to defer to the AI.

Figure 30: The LIME-Teaching user teaching introduction



The AI answered this incorrectly, look at the words the AI focused on:
The words highlighted in green are those that most led the AI to get the answer correctly and those in red led to it to answer incorrectly.

AI is incorrect
On the original example.

Context:

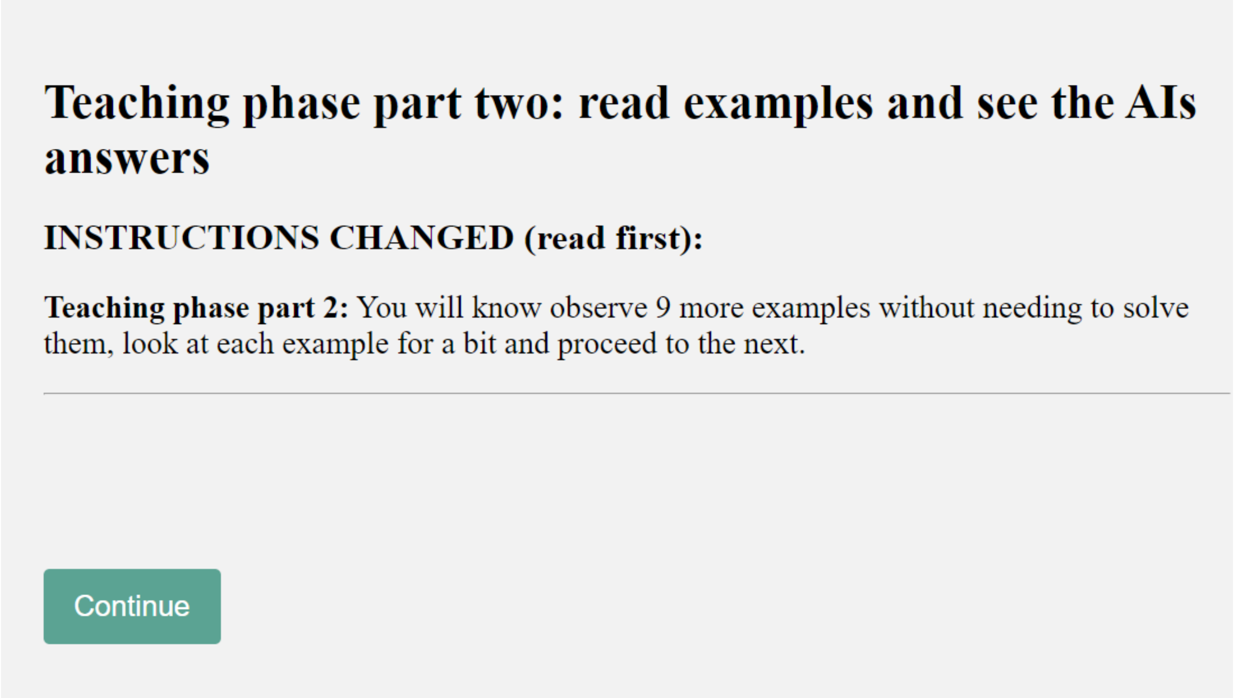
ansel elgort (born march 14, 1994) is an american actor, singer and a dj (under the name ansølo). [...] . </br> baby driver is a 2017 action crime comedy film written and directed by edgar wright. it stars ansel elgort, kevin spacey, lily james, eiza gonzález, jon hamm, jamie foxx, and jon bernthal. the plot follows baby, a young getaway driver and music lover who must work for a kingpin. the film is best known for its choreography, in which the actors' timing and movements are synced with the soundtrack.

Question:

What is the profession of Ansel Elgort's character in "Baby Driver?"

Write a sentence to describe the topic of the example you solved, this sentence should help you to predict if the AI is correct in future examples.
e.g. "The AI is good at answering questions about politics"

Figure 31: The LIME-Teaching feedback after answering teaching question.

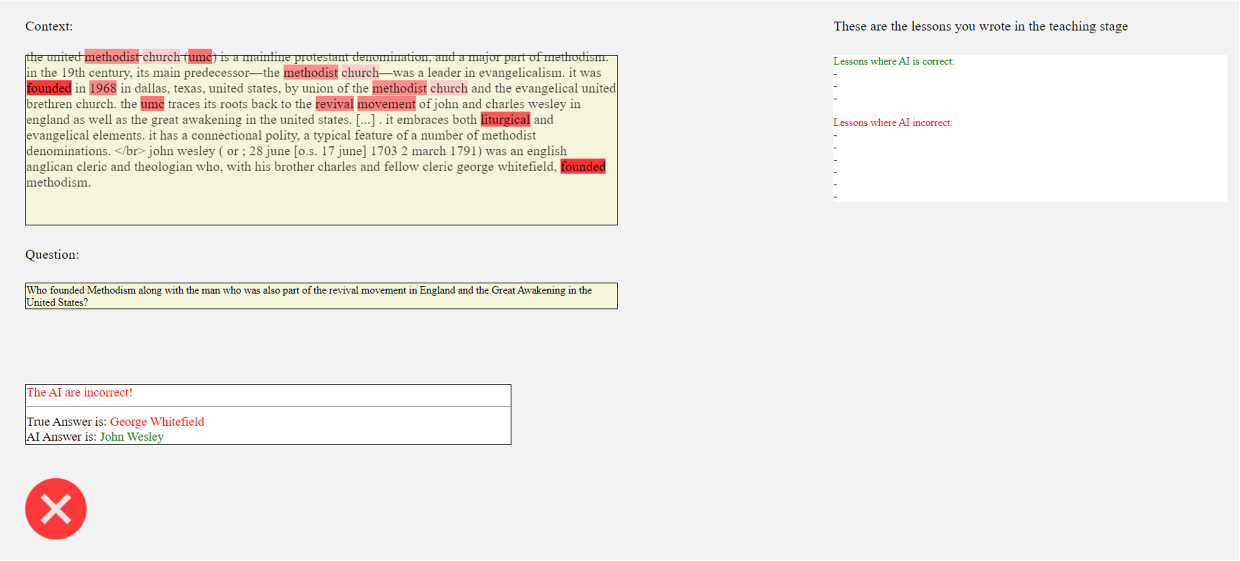


Teaching phase part two: read examples and see the AIs answers

INSTRUCTIONS CHANGED (read first):

Teaching phase part 2: You will know observe 9 more examples without needing to solve them, look at each example for a bit and proceed to the next.

Figure 32: The LIME-Teaching teaching introduction to second part of the teaching phase



Context:

the united methodist church (umc) is a mainline protestant denomination, and a major part of methodism. in the 19th century, its main predecessor—the methodist church—was a leader in evangelicalism. it was founded in 1968 in dallas, texas, united states, by union of the methodist church and the evangelical united brethren church. the umc traces its roots back to the revival movement of john and charles wesley in england as well as the great awakening in the united states. [...] . it embraces both liturgical and evangelical elements. it has a connectional polity, a typical feature of a number of methodist denominations. </br> john wesley (or ; 28 june [o.s. 17 june] 1703 2 march 1791) was an english anglican cleric and theologian who, with his brother charles and fellow cleric george whitefield, founded methodism.

Question:

Who founded Methodism along with the man who was also part of the revival movement in England and the Great Awakening in the United States?

The AI are incorrect!

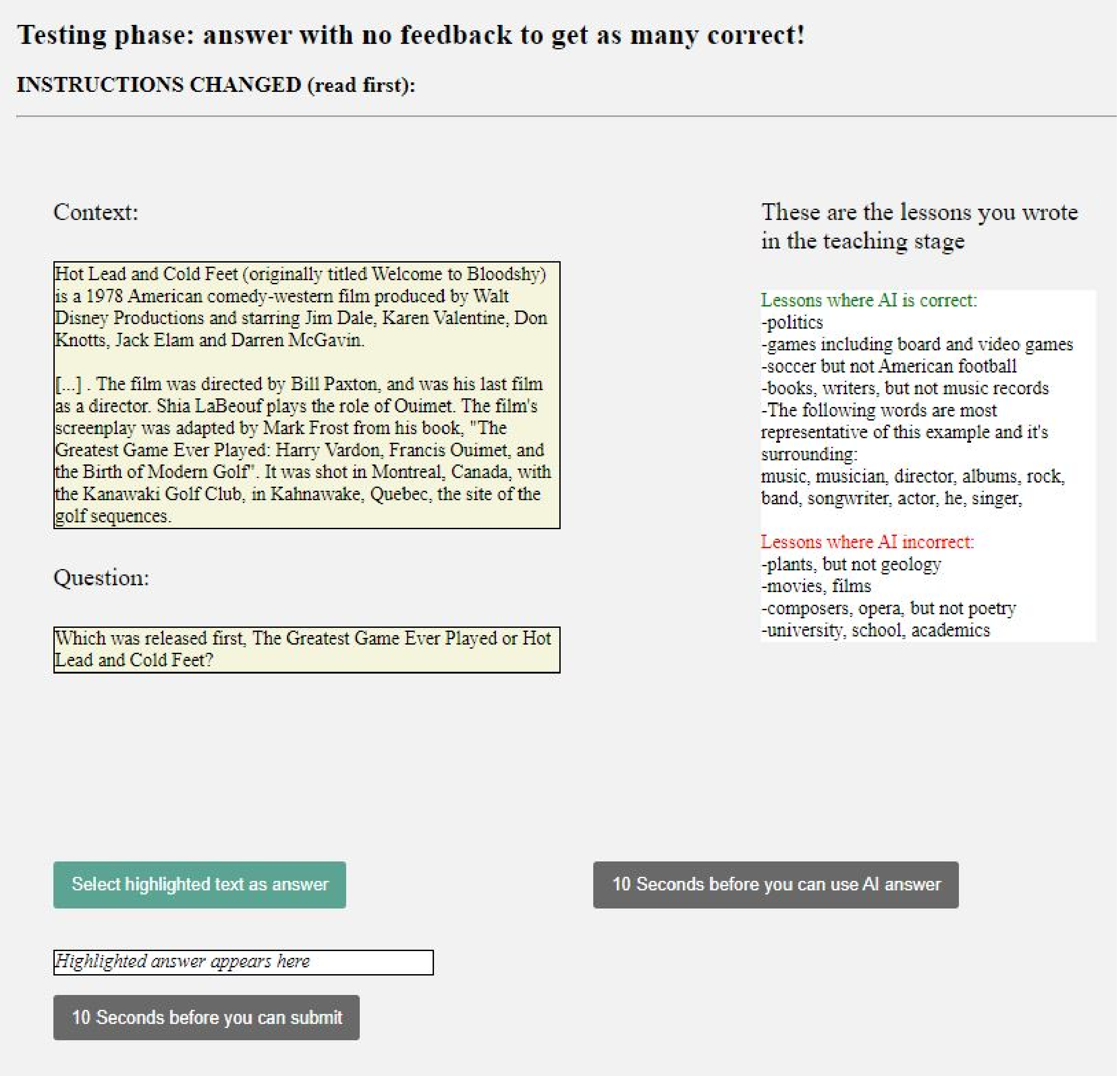
True Answer is: George Whitefield
AI Answer is: John Wesley

These are the lessons you wrote in the teaching stage

Lessons where AI is correct:
-
-
-

Lessons where AI incorrect:
-
-
-
-
-
-

Figure 33: The LIME-Teaching user interface of the second part of the teaching phase where users observe examples and the AI answers.



Testing phase: answer with no feedback to get as many correct!

INSTRUCTIONS CHANGED (read first):

Context:

Hot Lead and Cold Feet (originally titled Welcome to Bloodshy) is a 1978 American comedy-western film produced by Walt Disney Productions and starring Jim Dale, Karen Valentine, Don Knotts, Jack Elam and Darren McGavin.

[...] . The film was directed by Bill Paxton, and was his last film as a director. Shia LaBeouf plays the role of Ouimet. The film's screenplay was adapted by Mark Frost from his book, "The Greatest Game Ever Played: Harry Vardon, Francis Ouimet, and the Birth of Modern Golf". It was shot in Montreal, Canada, with the Kanawaki Golf Club, in Kahnawake, Quebec, the site of the golf sequences.

Question:

Which was released first, The Greatest Game Ever Played or Hot Lead and Cold Feet?

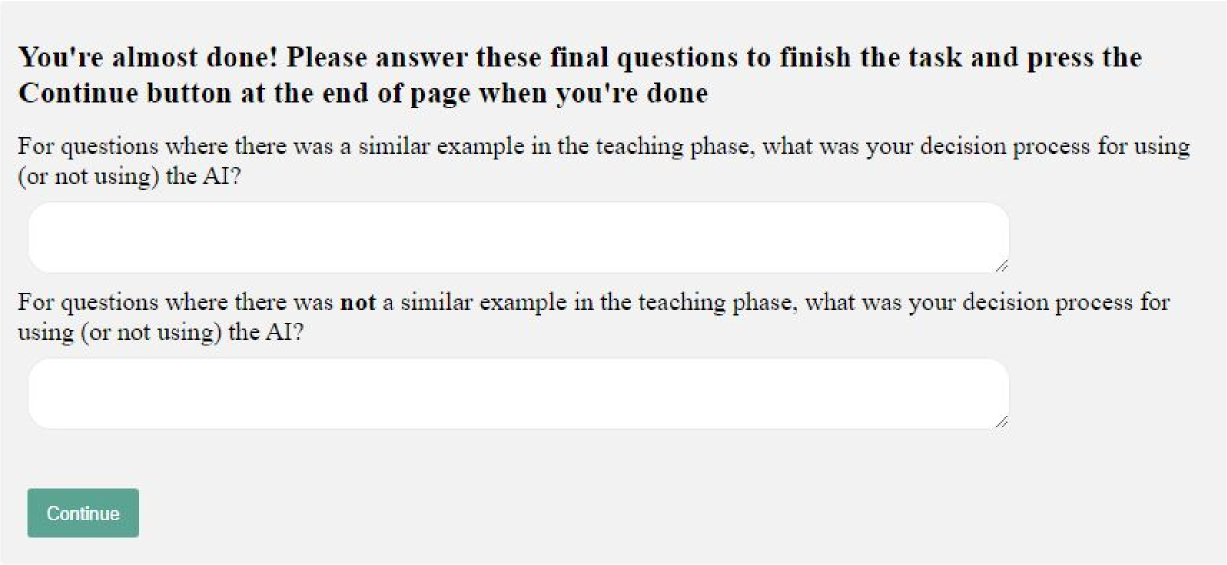
These are the lessons you wrote in the teaching stage

Lessons where AI is correct:
-politics
-games including board and video games
-soccer but not American football
-books, writers, but not music records
-The following words are most representative of this example and it's surrounding:
music, musician, director, albums, rock, band, songwriter, actor, he, singer,

Lessons where AI incorrect:
-plants, but not geology
-movies, films
-composers, opera, but not poetry
-university, school, academics

Select highlighted text as answer
10 Seconds before you can use AI answer
Highlighted answer appears here
10 Seconds before you can submit

Figure 34: Interface during testing.



You're almost done! Please answer these final questions to finish the task and press the Continue button at the end of page when you're done

For questions where there was a similar example in the teaching phase, what was your decision process for using (or not using) the AI?

For questions where there was **not** a similar example in the teaching phase, what was your decision process for using (or not using) the AI?

Figure 35: Questions collected after workers complete experiment for the Teaching condition.