Meta-Learning for Better Learning: Using Meta-Learning Methods to Automatically Label Exam Questions with Detailed Learning Objectives

Amir Zur*
Stanford University
Department of Computer Science
amirzur@stanford.edu

Isaac Applebaum*
Stanford University
Department of Biology
iapple23@stanford.edu

Jocelyn Elizabeth Nardo
Stanford University
Graduate School of Education
jnardo@stanford.edu

Dory DeWeese
Stanford University
Department of Chemistry
dory@stanford.edu

Sameer Sundrani
Stanford University
Department of Biomedical Computation
sundrani@stanford.edu

Shima Salehi
Stanford University
Graduate School of Education
salehi@stanford.edu

ABSTRACT

Detailed learning objectives foster an effective and equitable learning environment by clarifying what instructors expect students to learn, rather than requiring students to use prior knowledge to infer these expectations. When questions are labeled with relevant learning goals, students understand which skills are tested by those questions. Labeling also helps instructors provide personalized feedback based on the learning objectives each student struggles to master. However, developing detailed learning objectives is time-consuming, making many instructors unable to pursue it. Labeling course questions with learning objectives can be even more time-intensive. To address this challenge, we develop a benchmark for automatically labeling questions with learning objectives. The benchmark comprises 4,875 questions and 1,267 expert-verified learning objectives from college physics and chemistry textbooks. This dataset provides a large library of learning objectives, and, to the best of our knowledge, is the first benchmark to measure performance on labeling questions with learning objectives. We use meta-learning methods to train classifiers and test them against our benchmark in a few-shot classification setting. These classifiers achieve acceptable performance on a test set with previously unseen questions (AUC 0.84), as well as a course with previously unseen questions and unseen learning objectives (AUC 0.84). Our work facilitates labeling questions with learning objectives to help instructors provide better feedback and create equitable learning environments[1].

*Equal contribution.
[1]Repository: `https://github.com/AmirZur/smartstem-ai`

A. Zur, I. Applebaum, J. Nardo, D. DeWeese, S. Sundrani, and S. Salehi. Meta-learning for better learning: Using meta-learning methods to automatically label exam questions with detailed learning objectives. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 224–233, Bengaluru, India, July 2023. International Educational Data Mining Society.

`https://doi.org/10.5281/zenodo.8115677`

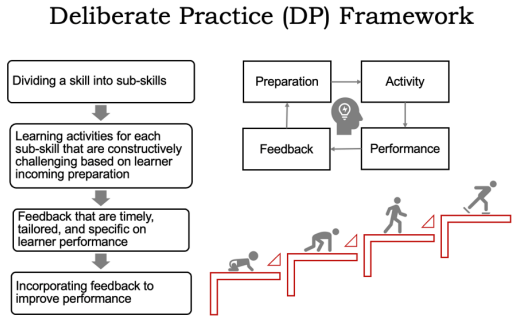

Figure 1: Deliberate practice framework adapted from [6].

Keywords

educational equity, assessment, learning objectives, pedagogical tool, personalized feedback, meta-learning

1. INTRODUCTION

Ericsson and colleagues argue that instructors can maximize their students' learning and improvement over time by facilitating deliberate practice [6]. To facilitate deliberate practice, instructors should break targeted skills into separate subskills, and design learning activities to practice each subskill in a way that takes students' prior knowledge into account. Importantly, students should receive "immediate informative feedback" about their performance on these tasks. Afterwards, students should be given the opportunity to improve their performance, whether by revising their work or by applying what they learned to a similar task. Our version of the deliberate practice framework is shown in Figure 1 [16].As shown by Glaser and Chi, breaking down larger skills into smaller subskills can also facilitate development of mental schema to organize domain knowledge, a key characteristic of expertise [4]. Deliberate practice provides a useful theoretical framework for understanding the benefits of detailed learning objectives and labeling course materials with these objectives.

Implementing deliberate practice in a classroom environment requires providing effective feedback. Ramaprasad argues that true feedback entails clearly articulating a goal, providing information about the gap between current performance and this goal, and ensuring that this information is used to bring current performance closer to the goal [19]. Ruiz-Primo and colleagues apply these criteria in their study of formative assessment [20]. Ruiz-Primo et al. argue that instructors should address three questions when teaching: "Where are we going?", "Where are we now?", and "How will we get there?". Completing the "Where are we going?" step involves writing learning objectives and clarifying what is considered evidence of achieving these learning objectives. Detailed learning objectives therefore provide a clear goal to measure student performance against. The "Where are we now?" step involves assessment, which provides a measure of students' current and prior knowledge. If assessments are intentionally designed around relevant learning objectives, and questions are labeled with the learning objectives they assess, this clarifies the gap between students' performance and the goals defined by the learning objectives. The "How will we get there?" step involves instructors tailoring their instructional practices to meet students' specific needs, which can include reinforcing concepts that a student may be struggling with and allowing students to revise their work [20, 24]. Labeling questions with learning objectives allows instructors to analyze the specific areas where each student needs help, and more effectively tailor instruction to the needs of their students. Finally, exam questions labeled with detailed learning objectives can particularly benefit students with less prior preparation, since these students may be less able to independently identify the skills tested by questions [17, 21, 22].

However, developing detailed learning objectives is difficult and time-consuming, which causes many educators to avoid writing learning objectives altogether, or to write only a few general learning objectives that do not communicate the specific skills that they expect students to demonstrate. Labeling questions with the relevant learning objectives is even more challenging and time-intensive, making it harder to provide effective feedback to students. To address such challenges, this work uses data mining and AI techniques to help instructors reap the benefits of learning objectives to facilitate equitable learning outcomes. We develop a benchmark for automatically labeling questions with learning objectives, using a custom dataset comprising a total of 4,875 questions and 1,267 expert-verified learning objectives drawn from four OpenStax college physics and chemistry textbooks, a widely-used college chemistry textbook, and Stanford University's general chemistry course materials (hereafter, Chem 31A). This dataset provides educators with a large library of learning objectives and questions, and, to the best of our knowledge, is the first benchmark to measure performance on labeling questions with detailed learning objectives. We use our benchmark to train and test three different types of classifiers: a multi-class multi-label (MCML) classifier, a ProtoTransformer, and a classifier adapted from GPT-3 embeddings. The ProtoTransformers and GPT-3 classifiers perform few-shot classification, a meta-learning task in which a classifier predicts the class of an input out of previously unseen classes given a few example items for each class (see Section 2.1 for more detail). Our results show that these few-shot classifiers achieve acceptable performance on our held-out test set, which consists of previously unseen course questions (AUC 0.84). Furthermore, the ProtoTransformer and GPT-3 classifiers generalize to a held-out course, which consists of previously unseen course questions and previously unseen learning objectives (AUC 0.84). Our work facilitates labeling questions with learning objectives, which can help instructors to incorporate learning objectives into their courses, provide better feedback, and create more equitable learning environments for students.

2. RELATED WORKS

Although previous research has supported educators' efforts to generate and analyze learning objectives [3, 12, 18], there has been limited research on facilitating the automatic labeling of questions with learning objectives. Some relevant work has been done on automatic exam grading, which can be viewed as labeling questions with rubric items [14, 30, 31]. Our work follows most closely the ProtoTransformer [31], which uses prototypical networks [25] to train a transformer-based model [29] to automatically grade computer science exams. In this section, we reintroduce the problem of few-shot classification, expand on the ProtoTransformer approach to this problem, and compare it with two other classification methods: multiclass-multilabel (MCML) classifiers and GPT-3 adapted as a few-shot classifier [2].

2.1 Few-Shot Classification

Few-shot classification is a meta-learning task in which, given a few training examples of each class, a classifier must adapt to predict new classes that were not previously seen in training [10, 11, 15, 25]. In our work, we formulate the task of labeling questions with learning objectives as a few-shot classification problem in which the classifier is trained to label questions with learning objectives, and the set of learning objectives and questions can vary from course to course.

In our learning setting, we consider a distribution D consisting of task indicators, input examples, and output labels. Formally, let $(t, x, y) \sim D$ be a task indicator, input example, and label drawn from a distribution of meta-learning tasks. We consider learning objectives $t \in T$ to be task indicators, and questions $x \in X$ to be model inputs. The task label, $y \in Y = \{0, 1\}$, is such that $y = 1$ if question x is labeled with learning objective t, and $y = 0$ otherwise. In this work, our goal is to train a model f_θ to accurately predict y given question x and learning objective t.

To perform few-shot classification, we are given a support set of k examples for each of the n prediction classes, $S = \{(x_1, y_1), \ldots, (x_{k \times n}, y_{k \times n})\}$. This work considers binary classification ($n = 2$), and so we interpret our support set as follows: S contains k examples of questions that are labeled with learning objective t (i.e., examples where $y = 1$), and k examples of questions *not* labeled with learning objective t (i.e., examples where $y = 0$). The goal of a few-shot classifier f_θ is, given S and an unlabeled question x, to classify whether or not it should be labeled with learning objective t. Note that the classes predicted by a few-shot classifier may not be the same between training time and inference time. In fact, the classes differ with each task type. That is, for the same input question x, the correct label may sometimes

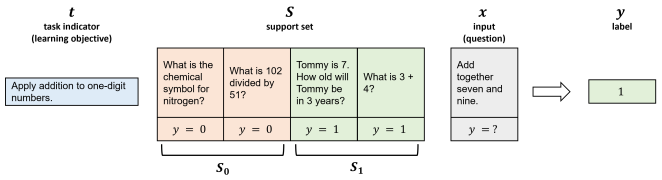

Figure 2: Example few-shot learning tasks in our setting, with $k = 2$. Each task consists of a task indicator (learning objective) t, a support set S containing k negative examples of questions (i.e., questions *not* labeled with learning objective t) and k positive examples of questions (i.e., questions labeled with learning objective t), an input question x, and a label y, where $y = 1$ if x should be labeled with t, and 0 otherwise.

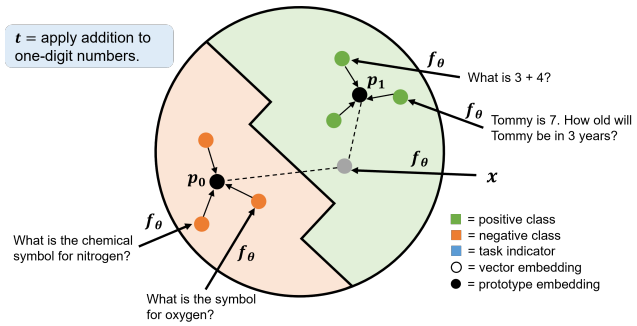

Figure 3: Visualization of prototypical networks adapted from Snell et al., 2017 [25], with $n = 2, k = 3$. For each class (i.e., questions labeled with t and questions not labeled with t) we are provided three examples of questions. The network f_θ maps each question to an embedding, and computes the prototype p_c of each class by averaging the class embeddings. A new input question, x, is then classified by taking the closest prototype to its embedding (in this figure, x is labeled 1).

be 0 and sometimes be 1, depending on the learning objective indicator t. Hence, a few-shot classifier must rely on the support set S, which consists of example questions for each class (i.e., examples of questions that should and shouldn't be labeled with t). As long as we can provide a support set S, a few-shot classifier can classify new questions with new learning objectives that do not appear during training. An example of few-shot classification is provided in Figure 2.

2.2 Prototypical Networks

One method for few-shot learning classification is prototypical networks [25], which serves as the basis of the ProtoTransformer and adapted GPT-3 classifiers [2, 31]. Prototypical networks embed inputs into vectors, such that similar inputs are closer together within the network's embedding space. For each prediction class, prototypical networks create a prototype embedding by taking the average embedding of all support examples in that class. New inputs are then classified by finding the closest class prototype within the network's embedding space.

Here we formalize the prototypical network algorithm. Given a support set S and class label c, let $S_c = \{(x_i, y_i) \in S \mid y_i = c\}$ be all examples of class c in S. For example, S_0 contains all questions in the support set *not* labeled with learning objective t. The prototype embedding of class c is $p_c = \frac{1}{k} \sum_{x_i \in S_c} f_\theta(x_i)$. That is, the prototype of each class represents the mean embedding of inputs with the same class label. The prototypical network then predicts the label y of an unseen question x by taking a softmax over the distance of the model's embedding of x, $f_\theta(x)$, from each prototype p_c (see Equation 1). In our setting, this is equivalent to asking, "Is the network's embedding of our question closer to the average embedding of questions labeled with learning objective t or questions not labeled with t?"

$$p(y = c \mid x) = \frac{\exp(-\text{dist}(f_\theta(x), p_c))}{\sum_{c'} \exp(-\text{dist}(f_\theta(x), p_{c'}))} \quad (1)$$

The network is trained to minimize the negative log-probability $-\log p(y = y \mid x)$ of the true class y. In our setting, dist is the L_2 distance function.

2.3 ProtoTransformer Classifier

The ProtoTransformer classifier is a prototypical network with a transformers-based architecture [31]. One key feature of the ProtoTransformer classifier is its ability to incorporate textual information from the task indicator (i.e., learning objective), which we expand upon in this section.

Prototypical networks generalize to previously unseen input examples (i.e., course questions) and to previously unseen task indicators (i.e., learning objectives). However, representing learning objectives as task indicators does not allow our model to utilize textual information from the learning objectives themselves. Note that as illustrated in Figure 3, prototypical networks do not make use of the content of the task indicator t – they only use the positive and negative examples of the task – in order to classify a new input x. The ProtoTransformer classifier addresses this problem by incorporating information from the task indicator in its embedding layer. The ProtoTransformer uses a separate embedding function g_ϕ, a pre-trained transformers model [29] with frozen parameters, to compute a vector representation of the learning objective, and adds this vector representation to the beginning of its model embeddings. The resulting embedded representation (i.e., learning objective token concatenated with question embedding) is passed into the transformers architecture, so that the interaction between the learning objective and question information can be used to construct an output vector. That is, the ProtoTransformer treats a learning objective as a sort of "task token," which informs the model of the relation between the input question and its learning objective. An example ProtoTransformer embedding layer is illustrated in Figure 4.

2.4 MCML Classifier

Another approach to labeling questions with learning objectives utilizes multi-class multi-label (MCML) classifiers [28]. MCML classifiers, given an input question x, learn to predict a binary vector y with an entry for each learning objective t, such that the t-th entry of y is 1 for all learning objectives that x should be labeled with, and 0 otherwise. Although MCML classifiers are not few-shot learners, in that they do not use the support set S in their predictions, they contribute to a field of prior research on fine-tuning transformer classifiers [29]. In our setting, we fine-tune an MCML model with a transformers-based architecture on a collected

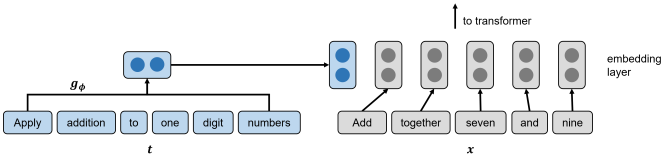

Figure 4: Figure adapted from Wu et al., 2021 [31], illustrating the embedding-space architecture used in our model in order to incorporate textual information from the task indicator (learning objective) t.

Table 1: Cross-comparison of classifiers used in this work to label questions with learning objectives. Both ProtoTransformers and GPT-3 perform few-shot classification, while MCML does not. Although GPT-3 does not require any fine-tuning or additional training, it is not freely accessible and must be accessed through a monetized API.

Classifier	Few-Shot	Fine-tuned	Free Access
ProtoTransformer	✓	✓	✓
MCML	✗	✓	✓
GPT-3	✓	✗	✗

dataset of questions labeled with learning objectives. The MCML model serves as our baseline, since it is not trained by the meta-learning algorithm for prototypical networks.

2.5 GPT-3 Classifier

Recent research has investigated the potential of large language models to perform few-shot classification without additional training [1, 2]. In our work, we adapt a recent large generative model, GPT-3 [2], as a prototypical network. That is, when run on an input question x, the adapted GPT-3 model output $f_{\text{GPT-3}}(x)$ is the activation of the last hidden layer within the GPT-3 model. The final layer activation constitutes a vector representation that is used to compute the prototype embedding for each class within the support set during few-shot classification. One advantage of GPT-3 is that since it is a large pre-trained language model, we expect its hidden layers to provide rich embeddings of text across various domains, including our collected course questions and learning objectives. Hence, GPT-3 does not require any fine-tuning nor additional training in our setting. On the other hand, GPT-3 is not publicly available, and, as of time of writing, is only accessible through a monetized API. This restriction does not apply to the ProtoTransformer and MCML classifiers, and is further discussed in Section 7.2.

In summary, we are not aware of prior research which has focused on the task of labeling course questions with learning objectives. Nevertheless, recent research on ProtoTransformer, MCML classification, and large language models such as GPT-3 provides avenues for developing models to label questions with learning objectives from previously unseen courses. We summarize the key attributes of prototypical networks, MCML classifiers, and few-shot GPT-3 as pertains to our work in Table 1.

3. METHODOLOGY

Our work introduces a benchmark for automatically labeling questions with learning objectives, on which we analyze the ProtoTransformer, MCML, and adapted GPT-3 classifiers. In this section, we provide details on the benchmark data collection process and the classifier training process.

3.1 Benchmark Creation

We collected 4,875 questions and 1,267 expert-verified learning objectives from four publicly available OpenStax textbooks (Chemistry 2e [8], University Physics I, II, and III [13]), a commonly-used university chemistry textbook (Principles of Chemistry 3rd edition [27]), and a Stanford University introductory chemistry course (Chem 31A). The questions from all OpenStax textbooks, as well as from the university chemistry textbook, are labeled with the corresponding list of learning objectives included in each textbook. To collect data from Chem 31A, we worked with members of the course teaching team to manually develop a list of 75 specific learning objectives for the course. For reliability, we independently labeled 30 exam questions (30 percent of the total dataset) and reached an agreement of 98 percent with a Cronbach's alpha score of 0.90, consistent with excellent inter-rater reliability [26]. We then labeled 98 exam questions from the 2021 offering of the course, consisting of four assessments, with the relevant learning objectives from our list. After coding all 98 exam questions, we found that only 53 of our learning objectives were covered by these exam questions. Although other repositories of learning objectives are available [3, 12, 18], to the best of our knowledge this is the first dataset to allow for training and benchmarking machine learning models on the labeling of course questions with relevant learning objectives. Example data points from our dataset can be found in Table 4, and are further discussed in Section A in the Appendix.

3.2 Classifier Training

Our main contribution in this work, besides the creation of the benchmark, is a collection of classifiers (ProtoTransformer, MCML, GPT-3), trained and tested on our benchmark for labeling questions with learning objectives. We use a ProtoTransformer with a BERT architecture, keeping the default settings from the original paper [29] (∼110M parameters). We train the ProtoTransformer with an Adam optimizer [9] and a learning rate of 1×10^{-5} for 8 epochs on our training dataset, which consists of ∼950 of k-shot classification tasks. The k value during training is 5, although we vary k during inference time. Our implementation of MCML is a BERT model (same hyperparameters as the ProtoTransformer) fine-tuned on our training data. We train the MCML classifier with an Adam optimizer and a learning rate of 1×10^{-5} for 5 epochs on our training dataset. Lastly, we adapt GPT-3 using the OpenAI curie model [2] (∼6.7B parameters) as described in Section 2.5, without additional training.

4. EXPERIMENTS

4.1 Experiment 1: Held-Out Test Set

We evaluate our model on a held-out test set, which consists of previously unseen learning objectives. Although questions were shared with the training set, the support set and query set consist of previously unseen combinations of questions and corresponding learning objectives, hence constituting

a previously unseen task. In our benchmark test dataset, positive examples (i.e., question-learning objective pairs in which the question is labeled by that learning objective) are balanced with negative examples (i.e., question-learning objective pairs in which the question is not labeled by that learning objective).

4.2 Experiment 2: Held-Out Course

Our second experiment considers using the trained classifiers to automatically label questions with learning objectives on a full course. We use a held-out course, Chem 31A, which consists of 53 previously unseen learning objectives and 98 previously unseen questions. We note that the MCML classifier is inapplicable in this setting, since the learning objective class labels are unavailable to it during training. Hence, we only compare the ProtoTransformer and GPT-3 classifiers. Unlike the test set, our held-out course is unbalanced with regards to positive and negative examples. A course question in Chem 31A is labeled with one to eight learning objectives of the total 53 available; therefore, our held-out course data is skewed towards negative examples.

Due to the imbalance in our dataset and multiple learning objective labels per question, we evaluate models with respect to ROC-AUC and F1 scores in addition to accuracy [7, 23]. The ROC-AUC metric considers a moving decision boundary, allowing us to better interpret the tradeoff between precision, or the ability to predict a short list of learning objectives that match the true learning objectives per question (with the risk of excluding true learning objectives), and recall, or the ability to predict all learning true objectives per question (with the risk of providing a long list containing unrelated learning objectives). Likewise, the F1 score balances precision and recall in its computation, accounting for class imbalances. We use the accuracy, AUC, and F1 evaluation metrics in both the held-out test and held-out course experiments.

4.3 Experiment 3: Recall Over Top-m

Due to the imbalanced nature of our held-out course dataset, in which questions are labeled with one to three of 53 learning objectives, we expect our model to over-predict the list of learning objectives with which to label a question (i.e., generate an overly-long list of candidate learning objectives for a single question). Interestingly, error types in our setting are imbalanced as well. A false positive (type I error) in our setting occurs when our model labels a question with an incorrect learning objective, meaning that an educator would need to filter a longer list of predicted learning objectives in order to label a question. Meanwhile, a false negative (type II error) occurs when our model fails to label a question with one of its correct learning objectives, meaning that an educator would need to search through the entire course list of learning objectives in order to find the correct learning objective. As a result, false negative errors would be far more time-consuming for an educator to correct. Hence, our last experiment considers recall, which measures a classifier's protection against false negative errors. We consider a graph of recall over top-m, where m represents the number of positive labels that the classifier assigns (i.e., the number of learning objectives labeled per question), chosen by taking the m learning objectives with the highest probability predicted by the classifier. A higher m represents more post-processing on behalf of educators (e.g., filtering from a list of five vs. ten predicted learning objectives); meanwhile, a higher recall score indicates that a greater percentage of true learning objectives are contained in the list of m learning objectives.

Table 2: Comparison of classifier performances on held-out test set. Highest scores are in bold.

k	Classifier	Accuracy	AUC	F1
0	MCML	$0.52 \pm .02$	$0.51 \pm .00$	$0.34 \pm .00$
1	GPT-3	$0.53 \pm .02$	$0.55 \pm .03$	$0.43 \pm .02$
	ProtoTransformer	$0.68 \pm .01$	$0.79 \pm .02$	$0.63 \pm .02$
2	GPT-3	$0.53 \pm .02$	$0.54 \pm .03$	$0.43 \pm .02$
	ProtoTransformer	$0.74 \pm .01$	$0.83 \pm .02$	$0.71 \pm .02$
5	GPT-3	$0.52 \pm .02$	$0.53 \pm .03$	$0.44 \pm .02$
	ProtoTransformer	$\mathbf{0.77 \pm .01}$	$\mathbf{0.84 \pm .01}$	$\mathbf{0.74 \pm .01}$

5. RESULTS

Below we detail classifier performance across each of our experiments. We also provide example classifier outputs in Table 5, and a preliminary qualitative analysis of classifier behavior in Section B in the Appendix.

5.1 Experiment 1: Held-Out Test Set

We report accuracy, ROC-AUC, and F1 scores on our held-out test set for the ProtoTransformer, MCML, and GPT-3 classifiers, across varying values of k (see Table 2). Higher values of k denote more examples provided to a few-shot learner per classification (in our case, more examples of questions that are labeled with a certain learning objective), and hence a greater manual effort to label questions with learning objectives. Since the MCML model is not a few-shot classifier, we treat it as a zero-shot classifier with $k = 0$. Both the ProtoTransformer and GPT-3 classifiers significantly outperform the MCML classifier, with the ProtoTransformer achieving the strongest performance at $k = 5$ (AUC of 0.84).

5.2 Experiment 2: Held-Out Course

We compare the ProtoTransformer and GPT-3 classifiers on a held-out course, Chem 31A, which consists of previously unseen questions and previously unseen learning objectives. We report results across varying values of k, corresponding to the number of example questions per learning objective that the classifier requires in order to label the remaining course's questions (see Table 3). The ProtoTransformer model requires at least $k = 1$ example per learning objective. Meanwhile, GPT-3 can be used as a zero-shot learning model, where each learning objective class is represented by the GPT-3 embedding of the learning objective itself. In this experiment, GPT-3 outperforms the ProtoTransformer classifier, achieving an AUC of 0.80 on the $k = 1$ setting.

5.3 Experiment 3: Recall Over Top-m

Figure 5 illustrates the trade-off between m, the total number of learning objective labels that a model assigns to a single input question, and the model's recall. A larger m means that an educator would need to filter between a longer list of outputted learning objectives. Meanwhile, a larger recall means that the list of outputted learning objectives contains

Table 3: Comparison of classifier performances on held-out course. Highest scores are in bold.

k	Classifier	Accuracy	AUC	F1
0	GPT-3	0.76 ± .02	0.66 ± .05	0.49 ± .02
1	GPT-3	0.63 ± .03	**0.80 ± .04**	0.46 ± .02
	ProtoTransformer	0.47 ± .05	0.73 ± .04	0.36 ± .03
2	GPT-3	0.77 ± .03	0.75 ± .05	0.55 ± .03
	ProtoTransformer	0.63 ± .02	0.74 ± .05	0.46 ± .02
5	GPT-3	**0.84 ± .03**	0.79 ± .05	**0.61 ± .03**
	ProtoTransformer	0.66 ± .03	0.77 ± .04	0.48 ± .02

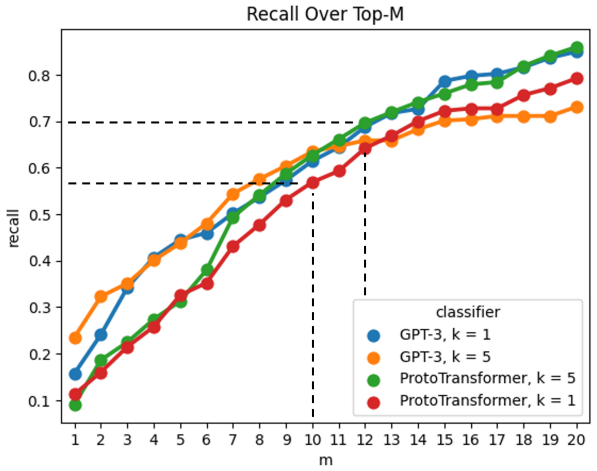

Figure 5: Model performance, measured as recall of true learning objectives, over m, or the number of learning objectives predicted by the model.

a higher percent of the learning objectives that match the input question. The plot below shows that at larger m values and $k = 5$, the ProtoTransformer model achieves stronger recall than GPT-3. Nevertheless, GPT-3 achieves higher recall at $k = 1$ and $k = 5$ when limited to lower m values (between 8 and 12).

6. DISCUSSION

The main contribution of our work is a custom benchmark and a collection of classifiers trained on our benchmark to facilitate the process of labeling questions with learning objectives. Our classifiers generalize to a held-out course, Chem 31A, with previously unseen questions and learning objectives. We therefore believe that these classifiers can be applied to other courses to help educators introduce learning objectives in their classrooms.

Our experiments evaluate an MCML classifier and two few-shot classifiers, the adapted GPT-3 and ProtoTransformer. When benchmarked on our held-out test set, which consists of previously unseen course questions and seen learning objectives, the ProtoTransformer significantly outperforms both the GPT-3 and MCML classifiers (AUC of 0.77, 0.52, 0.52, respectively). These results suggest that the ProtoTransformer model generalizes to new few-shot classification tasks, and is suitable for use in courses that share similar learning objectives to our dataset (e.g. university-level STEM courses). Meanwhile, the MCML classifier, without the ability to perform few-shot classification, is not as suitable as the meta-learning approaches of the adapted GPT-3 and ProtoTransformer classifiers.

In the second experiment, we analyze classifier performance on our held-out course, Chem 31A, with previously unseen questions and learning objectives. The results of this experiment demonstrate how few-shot classifiers could be used to automatically label new questions with new learning objectives. Both the ProtoTransformer and GPT-3 classifiers achieve acceptable performance on the $k = 5$ setting (AUC 0.77, 0.79, respectively), in which the instructor would need to provide 5 examples of questions for each learning objective. Interestingly, the GPT-3 classifier tested on the $k = 1$ setting – requiring only one example question per learning objective – achieves the strongest AUC score of 0.80. This is a promising result which showcases the capability of GPT-3 to perform few-shot classification without any additional training. Therefore, the GPT-3 classifiers can be a better choice for labeling questions with learning objectives of a new course with unseen learning objectives and questions.

Our recall over top-m plot, seen in Figure 5, confirms the strength of GPT-3 as a few-shot classifier. The ProtoTransformer achieves the strongest recall given a larger m value, meaning that when allowed to tag a question with 20 learning objectives, the ProtoTransformer is the most likely model to include the correct learning objectives within the list of 20 predictions. Nevertheless, the $k = 5$ GPT-3 classifier achieves acceptable recall (0.63) at $m = 10$, striking a balance between overly-long lists of learning objectives and the retrieval of accurate learning objectives. We note that the GPT-3 classifier in the $k = 1$ setting, which requires less manual question labeling on behalf of an educator, achieves an acceptable recall (0.69) at $m = 12$. Figure 5, then, illustrates the power of the ProtoTransformer and adapted GPT-3 classifiers to label previously unseen questions with previously unseen learning objectives.

7. LIMITATIONS

7.1 Benchmark Limitations

While the results in this work suggest that our dataset of questions labeled with relevant, specific learning objectives is a reliable and useful benchmark, it is limited by the specificity of the OpenStax learning objectives and their corresponding questions. An inspection of the OpenStax portion of our benchmark, which constitutes the training dataset for our models, reveals that a question is labeled by each of its subchapter's learning objectives, not all of which may be relevant. This limitation also means that questions spanning multiple concepts are only labeled with learning objectives from a particular course unit. See Section A in the Appendix for a detailed analysis of the OpenStax dataset.

The fact that OpenStax questions are not labeled with subsidiary learning objectives from other sub-chapters while Chem 31A questions are labeled with such subsidiary learning objectives may help explain why classifiers trained on OpenStax questions perform better on the held-out course, Chem 31A, than on the held-out OpenStax dataset (see Tables 2 and 3). Another potential explanation is that the OpenStax dataset contains 1,267 learning objectives while

the Chem 31A dataset contains 53 learning objectives, meaning that the classifiers need to choose from fewer learning objectives when labeling Chem 31A questions. The smaller number of learning goals in Chem 31A is likely more representative of a single course, rather than a textbook that could be used to teach a series of courses. Therefore, the OpenStax dataset could pose a more challenging labeling task than the intended use case of assisting course instructors.

Another notable limitation is that we automatically collected the OpenStax data using a custom web scraping program (available on our GitHub repository), without any data preprocessing such as removing special unicode characters or addressing typos. While this limitation does not seem to prevent our classifers from performing effectively on the OpenStax dataset, systematically correcting typos could improve classifier performance and increase the dataset's usefulness to both instructors and researchers.

7.2 Classifier Limitations

Key differences between our ProtoTransformer and GPT-3 models, beyond classification performance, include model size and accessibility. Our trained ProtoTransformer model is an order of magnitude smaller than the respective GPT-3 classifier ($\sim$110M vs. $\sim$6.7B parameters), and is freely accessible for usage and further training. As of time of writing, GPT-3 is only accessible through a monetized API[2], and, partly due to its size, is not readily available for additional training. Hence, we encourage further use and exploration of the ProtoTransformer classifier.

At the same time, we acknowledge that although achieving an AUC score of 77% on a held-out course is promising, the ProtoTransformer classifier may not be accurate enough for use in all introductory STEM courses. We hope that future research using our benchmark will improve classifier accuracy, and potentially generalizability to different course subjects. For immediate use in classroom settings, we recommend that instructors investigate model outputs carefully, and filter its predicted learning objectives down to the ones most relevant to the question at hand. Instructors can use Figure 5 to determine the number of learning objectives that they would like to filter from (we recommend an m between 12 and 16). Furthermore, future research could ensemble multiple classifiers together (e.g. ProtoTransformer at $k = 5$, GPT-3 at $k = 1$, and GPT-3 at $k = 5$) in order to improve classifier accuracy [5].

Lastly, our preliminary qualitative analysis of example model behavior (see Section B in the Appendix) suggests that the performance of our few-shot classifiers is limited by the provided input. That is, access to high-quality support examples during inference time (i.e., questions that are already labeled with learning objectives by the instructor) is essential for accurate prediction. Future work on decreasing k while maintaining high accuracy, along with work on identifying learning objectives that do not receive as much course coverage, can significantly enhance the capabilities of our classifiers.

[2]Information about the OpenAI API can be found here: `https://openai.com/blog/openai-api`

8. FUTURE WORKS

Our results enable many exciting future works for educators in chemistry, physics, and other STEM fields. By facilitating the process of labeling questions with learning objectives, we aim to help educators introduce learning objectives into their classrooms and label course materials with these objectives, actions that support students towards mastery-based learning approaches and promote equity [17]. Since our classifiers can label new questions with existing learning objectives and our dataset includes expert-verified learning objectives from multiple fields, our classifiers can be used to generate lists of detailed learning objectives for courses that currently have none. Rather than designing learning objectives from scratch, instructors could use our classifiers to label their existing course materials with the relevant learning objectives from our dataset. The list of learning objectives chosen by the classifiers can serve as a draft list of learning objectives for the course, which instructors can adapt to fit their needs. To facilitate these applications, our research team is currently developing an interactive web-based tool to allow instructors to experiment with our trained classifiers. This tool will allow instructors to automatically label their own questions with learning objectives from our datasets, or with other learning objectives that they provide. In addition, the tool will allow instructors to choose the value of hyperparameters such as m, the number of learning objectives that they would like the model to recommend as potentially relevant to each question, in order to best align with their needs. Furthermore, the performance of GPT-3 in this work as a prototypical neural network and as a zero-shot classifier motivates further exploration of GPT-3 as a meta-learning model, and its use within educational domains. Lastly, we encourage data scientists and educators to use and expand on our dataset of learning objectives, which we believe is the first benchmark of its kind to label questions with learning objectives.

9. CONCLUSIONS

Questions labeled with learning objectives can help students use feedback to better navigate their course, particularly benefiting students with less prior preparation. However, the task of labeling questions with learning objectives is time-consuming, making many instructors unable to pursue it. In this paper, we introduce a benchmark and trained classifiers for automatically labeling course questions with learning objectives. We show that meta-learning classifiers trained on our benchmark achieve acceptable performance on a test set with previously unseen questions (AUC 0.84), as well as a previously unseen course (AUC 0.84). We believe that our work, and future research in this realm, can support educators by facilitating the process of developing and utilizing learning objectives in their courses to create more effective and equitable learning environments.

10. REFERENCES

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are

few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] S. Chasteen, K. Perkins, P. Beale, S. Pollock, and C. Wieman. A thoughtful approach to instruction: Course transformation for the rest of us. 2011.

[4] M. T. Chi, R. Glaser, and M. J. Farr. *The nature of expertise*. Psychology Press, 2014.

[5] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

[6] K. A. Ericsson, R. T. Krampe, and C. Tesch-Römer. The role of deliberate practice in the acquisition of expert performance. *Psychological review*, 100(3):363, 1993.

[7] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[8] P. Flowers and K. Theopold. [etextbook] chemistry-2e, 2019.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] G. Koch, R. Zemel, R. Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[11] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[12] Y. Li, M. Rakovic, B. X. Poh, D. Gaševic, and G. Chen. Automatic classification of learning objectives based on bloom's taxonomy. *International Educational Data Mining Society*, 2022.

[13] S. J. Ling, J. Sanny, W. Moebs, G. Friedman, S. D. Druger, A. Kolakowska, D. Anderson, D. Bowman, D. Demaree, E. Ginsberg, et al. University physics volume 2. 2016.

[14] A. Malik, M. Wu, V. Vasavada, J. Song, J. Mitchell, N. Goodman, and C. Piech. Generative grading: Neural approximate parsing for automated student feedback. *arXiv preprint arXiv:1905.09916*, 2019.

[15] E. G. Miller, N. E. Matsakis, and P. A. Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000.

[16] J. E. Nardo. Ideal pedagogy stem presentation, 2021. Power Point Presentation, https://ideallabresearch.stanford.edu/.

[17] J. E. Nardo, N. C. Chapman, E. Y. Shi, C. Wieman, and S. Salehi. Perspectives on active learning: Challenges for equitable active learning implementation. *Journal of Chemical Education*, 99(4):1691–1699, 2022.

[18] R. Pepper, S. Chasteen, S. Pollock, and K. Perkins. Facilitating faculty conversations: Development of consensus learning goals. In *Physics Education Research Conference 2011*, volume 1413 of *PER Conference*, pages 291–294, Omaha, Nebraska, August 3-4 2011.

[19] A. Ramaprasad. On the definition of feedback. *Behavioral science*, 28(1):4–13, 1983.

[20] M. A. Ruiz-Primo, E. M. Furtak, C. Ayala, Y. Yin, and R. J. Shavelson. Formative assessment, motivation, and science learning. In *Handbook of formative assessment*, pages 139–158. Routledge, 2010.

[21] S. Salehi, E. Burkholder, G. P. Lepage, S. Pollock, and C. Wieman. Demographic gaps or preparation gaps?: The large impact of incoming preparation on performance of students in introductory physics. *Physical Review Physics Education Research*, 15(2):020114, 2019.

[22] S. Salehi, S. Cotner, and C. J. Ballen. Variation in incoming academic preparation: Consequences for minority and first-generation students. In *Frontiers in Education*, volume 5, page 552364. Frontiers Media SA, 2020.

[23] Y. Sasaki et al. The truth of the f-measure. *Teach tutor mater*, 1(5):1–5, 2007.

[24] L. A. Shepard. Classroom assessment to support teaching and learning. *The ANNALS of the American Academy of Political and Social Science*, 683(1):183–200, 2019.

[25] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[26] H. E. Tinsley and D. J. Weiss. Interrater reliability and agreement of subjective judgments. *Journal of Counseling Psychology*, 22(4):358, 1975.

[27] N. J. Tro. *Chemistry in focus: A molecular view of our world*. Cengage Learning, 2018.

[28] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] M. Wu, R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman. Variational item response theory: Fast, accurate, and expressive. *arXiv preprint arXiv:2002.00276*, 2020.

[31] M. Wu, N. Goodman, C. Piech, and C. Finn. Prototransformer: A meta-learning approach to providing student feedback. *arXiv preprint arXiv:2107.14035*, 2021.

APPENDIX

A. SAMPLE DATA

In this section we provide an overview of the OpenStax portion of our benchmark. Table 4 provides example input questions and their corresponding learning objective labels, sampled from our OpenStax training dataset.

We note that all questions from each sub-chapter of a given OpenStax textbook were labeled with every learning objective that the authors included for that subsection, rather than each question being hand-labeled with unique learning objectives. For example, all questions from the OpenStax Chemistry 2e [8] sub-chapter "6.1 Solving Problems

with Newton's Laws" would be labeled with all five of the learning objectives for this sub-chapter (see Table 4). This simplifying assumption allowed us to create a much larger dataset, including 4,875 labeled questions spanning 1,267 specific learning objectives from OpenStax university-level science textbooks, than would have been possible had we hand-labeled each question individually. While not every question in each sub-chapter focuses on every learning objective for that sub-chapter, the key learning objectives for each question are likely to be included in the learning objectives for that question's sub-chapter. Similarly, it is likely that all of a sub-chapter's learning objectives are relevant in varying degrees to the questions from that sub-chapter, so questions from the OpenStax portion of our dataset are unlikely to be labeled with off-topic learning objectives.

The most significant limitation resulting from this simplifying assumption is that questions in our OpenStax dataset are never labeled with subsidiary learning objectives from other sub-chapters. In theory, this could limit the usefulness of the OpenStax portion of our dataset in training classifiers to label questions that focus on integrating multiple course topics. As shown by our experimental results (see Table 3), classifiers trained on our OpenStax dataset were able to perform effectively on our Chem 31A dataset, where chemistry experts individually hand-labeled questions with learning objectives. Additionally, many questions from the Chem 31A dataset focus on integrating skills from multiple course topics, particularly the longer free-response questions and questions from the final exam. Our classifiers' ability to generalize to the Chem 31A dataset after being trained on the OpenStax dataset suggests that the benefits of the method we used to label the OpenStax questions, such as enabling the creation of a much larger dataset for training, outweigh the limitations mentioned above.

B. SAMPLE OUTPUTS

Table 5 presents the outputs of the ProtoTransformer classifier with $k = 5$ on a sample of questions from the held-out Chem 31A course.

A brief inspection suggests that the ProtoTransformer classifier does not solely rely on semantic keywords. For example, although the second question contains the phrase "vapor pressure" multiple times, the top three classifier predictions do not contain this phrase. Meanwhile, the first question does not explicitly state the ideal gas law, $PV = nRT$; however, the classifier infers the learning objective label.

Although a thorougher investigation is required to interpret the ProtoTransformer classifier's behavior, we hypothesize that the classifier more accurately identifies core learning objectives (e.g. "use the ideal gas law", "interpret a phase diagram") which appear in many course questions, and less accurately predicts learning objectives that are specific to a sub-unit (e.g. "apply the concept of percent by mass"). This is because few-shot classification requires access to high-quality samples of related questions. Since the pool of questions related to the ideal gas law in Chem 31A is richer than the pool of questions related to the concept of percent by mass, the ProtoTransformer classifier is likely to achieve higher accuracy on the former than on the latter.

Table 4: Sample questions and their corresponding learning objective labels from the OpenStax training dataset.

Course + *subchapter*	Question	Learning Objectives
University Physics I *6.1 Solving Problems with Newton's Laws*	A 30.0-kg girl in a swing is pushed to one side and held at rest by a horizontal force F so that the swing ropes are 30.0° with respect to the vertical. (a) Calculate the tension in each of the two ropes supporting the swing under these conditions. (b) Calculate the magnitude of F	Apply problem-solving techniques to solve for quantities in more complex systems of forces Use concepts from kinematics to solve problems using Newton's laws of motion Solve more complex equilibrium problems Solve more complex acceleration problems Apply calculus to more advanced dynamics problems
Chemistry 2e *4.3 Reaction Stoichiometry*	What mass of silver oxide, Ag_2O, is required to produce 25.0 g of silver sulfadiazine, $AgC_{10}H_9N_4SO_2$, from the reaction of silver oxide and sulfadiazine? 2 $C_{10}H_{10}N_4SO_2$ + Ag_2O → 2 $AgC_{10}H_9N_4SO_2$ + H_2O	Explain the concept of stoichiometry as it pertains to chemical reactions Use balanced chemical equations to derive stoichiometric factors relating amounts of reactants and products Perform stoichiometric calculations involving mass, moles, and solution molarity

Table 5: Presents the outputs of the ProtoTransformer classifier with $k = 5$, run on four sample questions from the Chem 31A course. The top $m = 3$ learning objectives predicted by the classifier are shown for each question, in order of model confidence. Correct predictions by the model are highlighted in green, while incorrect predictions are highlighted in red.

Question	True Learning Objectives	Predicted Learning Objectives ($m = 3$)
A mixture of 20.0 g of Ne and 20.0 g Ar have a total pressure of 1.60 atm and temperature of 298K. What is the partial pressure of Ar?	Apply the concept of percent by mass and percent by volume when solving problems Use the ideal gas law (PV=nRT) to solve problems	Use gas laws with stoichiometry to analyze chemical reactions of gasses Use the ideal gas law (PV=nRT) to solve problems Write and balance chemical and net-ionic equations
Decreasing the external pressure on a liquid at constant temperature will do which of the following:(a) Increase the boiling point, but not affect the vapor pressure(b) Decrease the boiling point, but not affect the vapor pressure(c) Increase the vapor pressure, therefore decreasing the boiling point(d) Increase the amount of heat required to boil a mole of the liquid(e) Both B and D are true	Calculate how vapor pressure will change as the pressure, volume, temperature, or amount are varied Interpret a phase diagram to determine what phase change may occur for a given change in pressure or temperature	Calculate changes in energy, enthalpy, and temperature that result from a chemical reaction Interpret a phase diagram to determine what phase change may occur for a given change in pressure or temperature Know the difference between systems and surroundings
At a constant external pressure, if work was done by the system on the surroundings, would you expect ΔE for the system to be greater than, less than or the same as the ΔH° for the system?(a) ΔE for the system would be greater than ΔH°(b) ΔE for the system would be less than ΔH°(c) ΔE for the system would the same as ΔH°(d) It is impossible to determine without knowing the magnitude of work done.	Calculate the work done by or on a gas	Calculate how vapor pressure will change as the pressure, volume, temperature, or amount are varied Know the difference between systems and surroundings Use the ideal gas law (PV=nRT) to solve problems
Determine the longest wavelength of light capable of removing an electron from a sample of potassium metal, if the binding energy for an electron in K is 1.76 × 103 kJ/mol. (a) 147 nm (b) 68.0 nm (c) 113 nm (d) 885 nm (e) 387 nm	Know how the photoelectric effect can be used to assess binding energy Use the relationship between the frequency and wavelength and velocity (speed) of a wave to calculate any one (frequency, wavelength or velocity) given the other two	Know how the photoelectric effect can be used to assess binding energy Use the relationship between the frequency and wavelength and velocity (speed) of a wave to calculate any one (frequency, wavelength or velocity) given the other two Explain how electronic structure gives rise to periodic trends (i.e. recognizing isoelectronic species)