CLUES: A Benchmark for Learning Classifiers using Natural Language Explanations

Rakesh R Menon* **Sayan Ghosh*** **Shashank Srivastava**
UNC Chapel Hill
{rrmenon, sayghosh, ssrivastava}@cs.unc.edu

Abstract

Supervised learning has traditionally focused on inductive learning by observing labeled examples of a task. In contrast, humans have the ability to learn new concepts from language. Here, we explore learning zero-shot classifiers for structured data[1] *purely* from language from natural language explanations as supervision. For this, we introduce CLUES, a benchmark for **C**lassifier **L**earning **U**sing natural language **E**xplanation**S**, consisting of a range of classification tasks over structured data along with natural language supervision in the form of explanations. CLUES consists of 36 real-world and 144 synthetic classification tasks. It contains crowdsourced explanations describing real-world tasks from multiple teachers and programmatically generated explanations for the synthetic tasks. We also introduce ExEnt, an entailment-based method for training classifiers from language explanations, which explicitly models the influence of individual explanations in making a prediction. ExEnt generalizes up to 18% better (relative) on novel tasks than a baseline that does not use explanations. We identify key challenges in learning from explanations, addressing which can lead to progress on CLUES in the future. Our code and datasets are available at: https://clues-benchmark.github.io.

1 Introduction

Humans have a remarkable ability to learn concepts through language (Chopra et al., 2019; Tomasello, 1999). For example, we can learn about *poisonous mushrooms* through an explanation like *'a mushroom is poisonous if it has pungent odor'*. Such

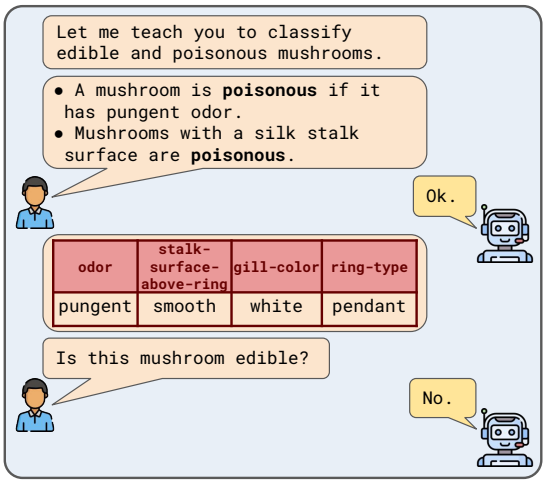


Figure 1: We explore learning classification tasks over structured data from natural language supervision in form of explanations. The explanations provide declarative supervision about the task, and are not example-specific. This is an example from the UCI Mushroom dataset, one of the 36 real-world datasets for which we collect multiple sets of explanations in CLUES.

an approach profoundly contrasts with the predominant paradigm of machine learning, where algorithms extract patterns by looking at scores of labeled examples of poisonous and edible mushrooms. However, it is unnatural to presume the availability of labeled examples for the heavy tail of naturally occurring concepts in the world.

This work studies how models trained to learn from natural language explanations can generalize to novel tasks without access to labeled examples. While prior works in this area (Srivastava et al., 2017, 2018; Hancock et al., 2018; Murty et al., 2020; Andreas et al., 2018; Wang* et al., 2020; Ye et al., 2020; Zhou et al., 2020) have explored explanations as a source of supervision, they evaluate models on a small number of tasks (2-3 relation extraction tasks in (Hancock et al., 2018; Wang* et al., 2020; Murty et al., 2020; Zhou et al., 2020), 7 email categorization tasks (Srivastava et al., 2017)). Owing to the paucity of large-scale benchmarks for

*Equal contribution

[1]By structured data, we refer to data that can be reasonably represented using tables. This is a highly flexible format for representing a lot of real-world data (e.g., spreadsheets, traditional classification datasets in CSV format, single-table databases, as well as structured text-rich data such as emails), with a large variety in possible table schemas.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6523 - 6546

learning from explanations over diverse tasks, we develop CLUES, a benchmark of classification tasks paired with natural language explanations. Over the last few decades, researchers and engineers alike have put immense effort into constructing structured and semi-structured knowledge bases (e.g., structured tables on Wikipedia, e-commerce sites, etc.). Developing models that can reason over structured data is imperative to improve the accessibility of machine learning models, enabling even non-experts to interact with such data. Hence, in this work, we specifically formulate our classification tasks over structured data.

Our benchmark is divided into CLUES-Real and CLUES-Synthetic consisting of tasks from real-world (UCI, Kaggle, and Wikipedia) and synthetic domains respectively. Explanations for CLUES-Real are crowdsourced to mimic the diversity and difficulty of human learning and pedagogy. For CLUES-Synthetic, we generate the explanations programmatically to explicitly test models' reasoning ability under a range of structural and linguistic modifications of explanations.

We train models with a mix of explanations and labeled examples, in a multi-task setup, over a set of *seen* classification tasks to induce generalization to *novel* tasks, where we do not have any labeled examples. Ye et al. (2021) refer to this problem setup as "cross-task generalization". Some recent methods on cross-task generalization from language use instructions/prompts (Mishra et al., 2022; Sanh et al., 2022; Wei et al., 2021) describing information about *'what is the task?'* to query large language models. In contrast, language explanations in CLUES provide the logic for performing the classification task, or intuitively *'how to solve the task?'*. For the running example of mushroom classification, an instruction/prompt might be *'can you classify a mushroom with pungent odor as poisonous or edible?'*. On the other hand, an example of an explanation in CLUES is *'a mushroom is poisonous if it has pungent odor'*.

We find that simply concatenating explanations to the input does not help pre-trained models, like RoBERTa (Liu et al., 2019), generalize to new tasks. Thus, we develop ExEnt, an entailment-based model for learning classifiers guided by explanations, which explicitly models the influence of individual explanations in deciding the label of an example. ExEnt shows a relative improvement of up to 18% over other baselines on unseen tasks.

To identify the challenges of learning from explanations, we perform extensive analysis over synthetic tasks. Our analysis explores how the structure of an explanation (simple clauses vs. nested clauses) and the presence of different linguistic components in explanation (conjunctions, disjunctions, and quantifiers) affect the generalization ability of models.

The rest of the paper is structured as follows: we describe our crowdsourced-benchmark creation pipeline in §3. In §4, we analyze our collected data. In §5, we describe our models, experiments, and results. We conclude with a brief discussion on the contributions and our findings, followed by a statement of ethics and broader impact. Our contributions are:

- We introduce CLUES, a benchmark for learning classifiers over structured data from language.
- We develop ExEnt, an entailment-based model for learning classifiers guided by explanations. ExEnt shows a relative improvement of up to 18% over other baselines on generalization to novel tasks.
- We explore the effect on the generalization ability of models learning from language by ablating the linguistic components and structure of explanations over our benchmark's synthetic tasks.

2 Related Work

Learning concepts from auxiliary information: Prior work has explored techniques to incorporate 'side-information' to guide models during training (Mann and McCallum, 2010; Ganchev et al., 2010). More recently, researchers have explored using language in limited data settings for learning tasks such as text classification (Srivastava et al., 2017, 2018; Hancock et al., 2018) and question answering (Wang* et al., 2020; Ye et al., 2020). However, we diverge from these works by exploring the generalization ability of classifiers learned by using language over novel tasks as opposed to gauging performance only on seen tasks.

Explanation-based Datasets: The role of explanations and how they can influence model behavior is a widely studied topic in machine learning (Wiegreffe and Marasović, 2021). Among language-based explanation studies, past work has primarily developed datasets that justify individual predictions made by a model (also called, local explanations) (Rajani et al., 2019; Camburu et al., 2018), *inter alia*. In contrast, our work focuses

on explanations that define concepts and capture a broad range of examples rather than individual examples. Our notion of explanations is shared with Andreas et al. (2018); Srivastava et al. (2017, 2018). We differ from these works as (1) our benchmark comprises a large set of classification tasks spanning diverse concepts for learning from explanations as opposed to working on a limited set of tasks in prior work and (2) our benchmark is domain agnostic in the source of classification tasks considered as long as we can represent the inputs of the task in a tabular (structured) format.

Few-shot & Zero-shot learning: Large pre-trained language models (LMs) (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020) have been shown to perform impressively well in few-shot settings (Brown et al., 2020; Lester et al., 2021). Reformulating natural language tasks with patterns has been shown to boost few-shot learning ability for small language models as well (Schick and Schütze, 2021; Tam et al., 2021). More recently, a few works have focused on evaluating the generalization of models to unseen tasks by using prompts and performing multi-task training (Mishra et al., 2022; Ye et al., 2021; Sanh et al., 2022; Min et al., 2021; Chen et al., 2022; Aghajanyan et al., 2021) While the training and evaluation setup is similar, our work is significantly different from these works as (1) the explanations in our work provide rationales for making a classification decision as opposed to explaining a task using prompts, (2) we explore classification over structured data as opposed to free-form text by designing a model that can leverage explanations.

3 Creating CLUES

In this section, we describe our benchmark creation process in detail. In CLUES, we frame classification tasks over structured data represented in tabular format. Based on the source of tables used to construct the classification tasks, we consider two splits of our benchmark, CLUES-Real (real-world datasets) and CLUES-Synthetic (synthetic datasets).

3.1 CLUES-Real

We first gather/create classification tasks from UCI, Kaggle, and Wikipedia tables, then collect explanations for each classification task.

3.1.1 Collecting classification datasets

Classification tasks from UCI and Kaggle. UCI ML repository[2] and Kaggle[3] host numerous datasets for machine learning tasks. For our benchmark, we pick out the tabular classification datasets. Then, we manually filter the available datasets to avoid ones with (a) many missing attributes and (b) complex attribute names that require extensive domain knowledge making them unsuitable for learning purely from language. CLUES-Real contains 18 classification tasks from UCI and 7 from Kaggle (the details of tasks are in Appendix B).

Mining tables from Wikipedia. Wikipedia is a rich, free source of information readily accessible on the web. Further, a lot of this information is stored in a structured format as tables. We explore creating additional classification tasks based on tables from Wikipedia, where each row in a table is assigned a category label. However, only a small fraction of the tables might be suitable to frame a classification task for our benchmark. Thus, we need to identify suitable tables by *mining* a large collection of tables from Wikipedia (we use Wikipedia dump available on April 2021). We formalize this mining-and-pruning process as a crowdsourcing task (on Amazon Mechanical Turk), where we present each turker with a batch of 200 tables and ask them to pick out suitable tables from that batch. For a table considered suitable by a turker, we further ask the turker to mention which column of the table should be considered as providing the classification labels. We identified 11 classification tasks corresponding to 9 Wikipedia tables after mining around 10K Wikipedia tables (the details of tasks are provided in Appendix B).

3.1.2 Explanation Collection Pipeline

Our explanation collection process consists of two stages – (1) teachers providing explanations after reviewing multiple labeled examples of the task, and (2) students verifying explanations and classifying new examples based on explanations for the tasks.

Collecting explanations: We use the Amazon Mechanical Turk (AMT) platform to collect explanations for CLUES-Real. In each HIT, we provide turkers with a few labeled examples of a dummy task (each corresponding to a row in a table) and a set of good and bad explanations for the task to

[2] https://archive.ics.uci.edu/ml/
[3] https://www.kaggle.com/datasets

teach them about the expected nature of explanations. Next, we test them on a 'qualification quiz' to gauge their understanding of good explanations.

Upon qualification, the turker advances to the explanation collection phase of the HIT. At this stage, the turker is provided with 15-16 labeled examples of a task in CLUES-Real and we ask them to write explanations describing the logic behind the classification for each class. Turkers are required to submit a minimum of two explanations (≥ 5 tokens each) for each task.

Further, teachers can test their understanding by taking a validation quiz, where they make predictions over new unlabeled examples from the task. Based on their informed classification accuracy, teachers can optionally refine their explanations.

Finally, when turkers are content with their performance, they 'freeze' the explanations and advance to the test-quiz where they are evaluated on a new set of unlabeled examples from the task (different from validation quiz).[4] We will refer to turkers who have provided responses at this stage as 'teachers' since they provide explanations to 'teach' models about different classification tasks.

Verification of explanations: After the explanation collection, we validate the utility of the sets of explanations for a task from each teacher by evaluating if they are useful they are for other humans in learning the task. For this, a second set of turkers[5] is provided access to the collected explanations from a teacher for a task, but no labeled examples. These turkers are then asked to predict the labels of test examples from the held-out test set, solely based on the provided explanations.

Additionally, we ask turkers in the verification stage to give a Likert rating (1-4 scale) on the usefulness of each explanation. Since the turkers in the verification stage perform the classification task using language explanations from a teacher, we refer to them as 'students' for our setup.

Thus, the tasks in CLUES-Real contain explanations from multiple teachers and multiple students corresponding to a teacher. This provides rich information about variance in teacher and student performance indicating how amenable different tasks are for learning via language. We provide insights into the performance of teachers and students of our setup in §4.

[4]For reference, we show snapshots of our annotation interface in Appendix §F.

[5]59 turkers participated in this stage.

CLUES-Real	
# Binary	26
# Multiclass	10
Avg. # Expls./task	9.6
Avg. # teachers	5.4
Avg. # Expls./teacher	2.3
# students/teacher	3
Max. # examples	65K
Min. # examples	5
Median. # examples	442
Avg. # features	5.6

CLUES-Synthetic	
# Task types	48
# Binary	94
# Multiclass	50
Avg. # Expls./task	1.7
# Examples/task	1000
# features/task	5

Table 1: Statistics of tasks in CLUES

3.2 CLUES-Synthetic

The complexity and fuzziness of real-world concepts and the inherent linguistic complexity of crowdsourced explanations can often shroud the aspects of the task that make it challenging for models to learn from explanations. To evaluate models in controlled settings where such aspects are not conflated, we create CLUES-Synthetic, a set of programmatically created classification tasks with varying complexity of explanations (in terms of structure and presence of quantifiers, conjunctions, etc.) and concept definitions.

We create tasks in CLUES-Synthetic by first selecting a table schema from a pre-defined set of schemas, then generating individual examples of the task by randomly choosing values (within a pre-defined range, obtained from schema) for each column of the table. Next, we assign labels to each example by using a set of 'rules' for each task. In this context, a 'rule' is a conditional statement (analogous to conditional explanations that we see for real-world tasks) used for labeling the examples. We use the following types of rules that differ in structure and complexity (c_i denotes i^{th} clause and l denotes a label):

- Simple: IF c_1 THEN l
- Conjunctive: IF c_1 AND c_2 THEN l
- Disjunctive: IF c_1 OR c_2 THEN l
- Nested disjunction over conjunction: IF c_1 OR (c_2 AND c_3) THEN l
- Nested conjunction over disjunction: IF c_1 AND (c_2 OR c_3) THEN l
- For each of the above, we include variants with negations (in clauses and/or labels): Some examples–IF c_1 THEN NOT l, IF c_1 OR NOT c_2 THEN l

We also consider other linguistic variations of rules by inserting quantifiers (such as 'always', 'likely'). The synthetic explanations are template-generated based on the structure of the rules used in creating

Vocabulary	1026
Avg. # tokens	15.53
Unique bigrams	3300

(a) Lexical statistics

Max. Score	106.67
Min. Score	3.12

(b) Flesch Reading Complexity Scores

Table 2: Explanations Statistics for CLUES

the task. For brevity, we defer additional details on the use of quantifiers, label assignment using rules, and creation of synthetic explanations to Appendix A. Overall we have 48 different task types (based on the number of classes and rule variants) using which we synthetically create 144 classification tasks (each containing 1000 labeled examples).

4 Dataset analysis

In this section, we describe the tasks and the collected explanations in CLUES.

Task Statistics: Table 1 shows the statistics of tasks in CLUES. The real-world tasks in our benchmark are from a wide range of domains, such as data corresponding to a simple game (e.g. tic-tac-toe), medical datasets (e.g. identifying liver patients), merit-classification of teachers and students, network-related datasets (eg. internet-firewall), among others. The synthetic tasks are created using table schemas denoting different domains, such as species of animals, species of birds, etc. (details in Appendix A).

As seen in Table 1, 5.4 explanation sets were collected for each classification task from human teachers on average. Further, each explanation set was verified by 3 students during the verification task. An aggregate of 133 teachers provide 318 explanations for tasks in CLUES-Real. All collected explanations were manually filtered and irrelevant explanations were removed.

Lexical analysis of explanations: Table 2a shows the statistics for explanation texts in our dataset.[6] We evaluate the average length of the explanation texts, vocabulary size and number of unique bigrams present in the explanations.

Explanation characteristics: Following Chopra et al. (2019), we categorize the explanations based on the different aspects of language (generics, quantifiers, conditional, and negation) present in these explanations. Table 3 shows the statistics of various categories in our dataset. Note that an explanation might belong to more than one category (for example, an example like *"if the number of hands equal to 2, then it is usually foo"*, will be categorized both as having both conditional and quantifiers.) We found that around 52% of the explanations for the real-world tasks had quantifiers (such as 'some', 'majority', 'most', etc.) in them. A full list of quantifiers present in the data is given in Appendix A.

CATEGORY	EXAMPLE	REAL	SYN
Generic	Being over 50 increases the risk of a stroke.	48 %	50 %
Quantifier	... *usually* means you won't have heart disease.	52 %	50 %
Conditional	*If* color code ... , *then* ...	15 %	100 %
Negations	... is *not* low.	16 %	50%

Table 3: Count of explanations in our dataset based on various aspects of language present in them

Reading complexity: We analyze the reading complexity of crowdsourced explanations by using *Flesch reading ease*[7]. Reading complexity values for our crowdsourced explanations vary from 3.12 (professional grade reading level) to 106.67 (easier than 3rd-grade reading level), with a median value of 65.73 (8th/9th-grade reading level).

Usefulness of the explanations: During the validation stage, we ask the turkers to provide a rating (on a Likert scale from 1 to 4) on the utility of the explanations for classification. The semantics of ratings are, 1 – 'not helpful', 2 – 'seems useful', 3 – 'helped in predicting for 1 sample', and 4 – 'mostly helpful in prediction'. The average rating for the explanations in CLUES-Real is 2.78, denoting most explanations were useful, even if they did not directly help predict labels in some cases. In Figure 2(a), we also provide a histogram of the Likert ratings provided by the students.

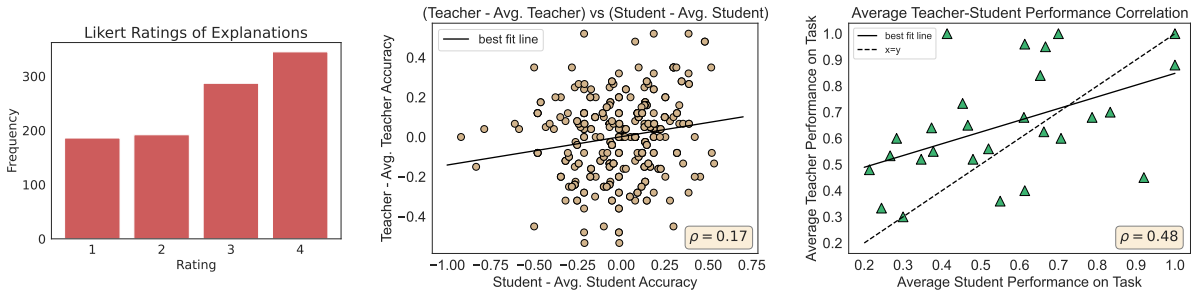
	VALIDATION	TEST
Teacher	69%	64%
Student	-	55%

Table 4: Teacher/student performance on CLUES-Real

Characteristics of teachers and students: Figure 2(b) shows the normalized teacher performance vs normalized student performance for teacher-student pairs in CLUES-Real. Normalized performance of an individual teacher (or, student) on a task is defined as the difference between the performances of the teacher (or, student) and an average teacher (or, student) for the same task. The positive correlation ($\rho = 0.17$) suggests that students tend

[6] Statistics in Table 2a was obtained using the spacy tokenizer.

[7] https://en.wikipedia.org/wiki/Flesch_Kincaid_readability_tests



(a) (b) (c)

Figure 2: (a) Histogram of count of explanations corresponding to different usefulness likert ratings. (b) Students typically perform well when taught tasks by good teachers. (c) Positive correlation in the average performance between a teacher and student for a task. (ρ denotes Pearson correlation coefficient in each of the plots)

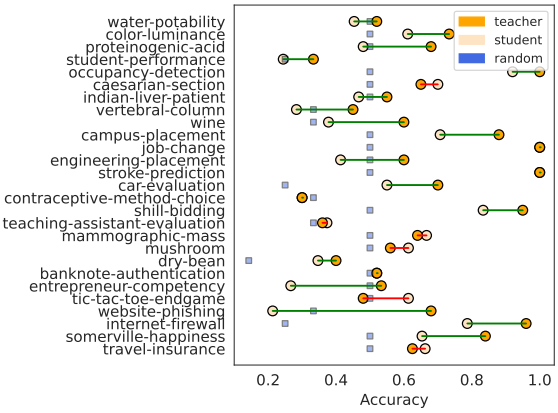


Figure 3: Average student vs average teacher performance for tasks in `CLUES-Real`. Red lines indicate cases where the student performance is more than the teacher performance. Green lines indicate cases where teachers perform better than students.

to perform well if taught by well-performing teachers. Positive correlation ($\rho = 0.48$) in Figure 2(c), indicates that task difficulty (captured by classification accuracy) is well-correlated for a teacher and student on average.

On visualizing the difference between an average student and an average teacher performance for each task in `CLUES-Real`, we find that an average teacher performs better than the average student on most tasks. However, for the 'tic-tac-toe' task in `CLUES-Real`, we find that the student accuracy was around 13% higher than average teacher performance. We hypothesize that this task can be solved by commonsense reasoning without relying on the provided explanations, resulting in students performing better than teachers. We quantify the average performance of teachers and students on `CLUES-Real` in Table 4.[8] We find that students perform lower than teachers on average as expected since a teacher has more expertise in the task. Moreover, it is challenging to teach a task perfectly using explanations in a non-interactive setting where a student cannot seek clarifications.

Additional data analysis and details of HIT compensation can be found in Appendix C and D.

5 Experiment Setup and Models

In this section, we describe our training and evaluation setup, our models, and experimental findings.

5.1 Training and Evaluation Setup

Our goal is to learn a model that, at inference, can perform classification over an input x to obtain the class label y, given the set of explanations E for the classification task. Figure 4 shows our setup, where we train our model using multi-task training over a set of tasks $\mathcal{T}_{seen}$ and evaluate generalization to a new task, $t \in \mathcal{T}_{novel}$. The task split we use for our experiments can be found in Appendix E.1. We select our best model for zero-shot evaluation based on the validation scores on the seen tasks. Since we do not make use of any data from the novel tasks to select our best model, we maintain the *true zero-shot* setting (Perez et al., 2021).

We encode each structured data example, x, as a text sequence, by linearizing it as a sequence of attribute-name and attribute-value pairs, separated by `[SEP]` tokens. To explain, the leftmost attribute-name and attribute-value pair of structured input example in Figure 1 is represented as `'odor | pungent'`. The linearization allows us to make use of pre-trained language models for the classification task. Our linearization technique

[8] Note that teacher scores in the tables and figures do not include 9 Wikipedia Tasks for which the authors formed the explanations. These 9 datasets had extremely few samples (~5), so this procedure was adopted. The list of crowdsourced tasks can be found in Table 7.

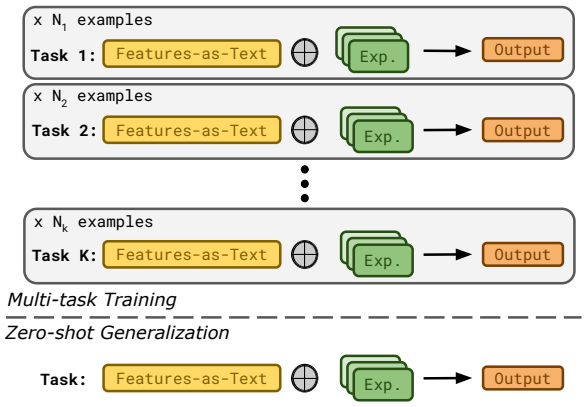


Figure 4: **Benchmark setup**: The model is trained on a set of classification tasks using explanations. At inference, the model is evaluated zero-shot on novel tasks using *only* explanations for the novel tasks.

is similar to the one used in Yin et al. (2020) with the exception that we do not use the column type. We will refer to the linearized format of structured inputs by 'Features-as-Text' or 'FaT'.

5.2 Baseline models

For our baselines, we make use of a pre-trained RoBERTa model (Liu et al., 2019). However, RoBERTa with the standard-fine-tuning approach cannot allow a generalization test as the number of output classes varies for each task. Furthermore, we cannot train individual class heads at inference since we test *zero-shot*. Hence, we make the following modifications to make RoBERTa amenable for zero-shot generalization tests: a pre-trained RoBERTa model takes the linearized structured data (FaT) as input and outputs a representation for this context (in the `[CLS]` token). Next, we run another forward pass using RoBERTa to obtain a representation of the labels based on their text (e.g., 'poisonous' or 'edible' for our example in Figure 1). Finally, we compute the probability distribution over labels by doing a dot-product of the representations of the input and the labels. We train this model using cross-entropy loss. In our experiments, we refer to this model as RoBERTa w/o Exp since the model does not use any explanations.

We also experiment with a RoBERTa w/ Exp. model where a RoBERTa model takes as input a concatenated sequence of all the explanations for the task along with FaT. The rest of the training setup remains the same as RoBERTa w/o Exp.

We find that a simple concatenation of explanations is not helpful for zero-shot generalization to novel tasks (results in Figure 6). Next, we describe `ExEnt` which explicitly models the role of each explanation in predicting the label for an example.

5.3 `ExEnt`

To model the influence of an explanation towards deciding a class label, we draw analogies with the entailment of an explanation towards the structured input. Here, given a structured input (*premise*) and an explanation (*hypothesis*), we need to decide whether the explanation strengthens the belief about a specific label (*entailment*), weakens belief about a specific label (*contradiction*) or provides no information about a label (*neutral*).

Figure 5 shows the overview of our explanation-guided classification model, `ExEnt`; given a structured input and explanation of a task, let l_{exp} denote the label mentioned in the explanation, and L denote the set of labels of the task. The entailment model assigns logits p_e, p_c and p_n to the hypothesis being entailed, contradicted or neutral respectively w.r.t. the premise. Based on the label assignment referred to by an explanation, we assign logits to class labels as follows:

- **If explanation mentions to assign a label** : Assign p_e to l_{exp}, p_c is divided equally among labels in $L \setminus \{l_{exp}\}$, and p_n is divided equally among labels in L.
- **If explanation mentions to <u>not</u> assign a label** : This occurs if a negation is associated with l_{exp}. Assign p_c to l_{exp}, p_e is divided equally among labels in $L \setminus \{l_{exp}\}$, and p_n is divided equally among labels in L.

We obtain logit scores over labels of the task corresponding to each explanation as described above. We compute the final label logits by aggregating (using mean) over the label logits corresponding to each explanation of the task. The final label logits are converted to a probability distribution over labels, and we train `ExEnt` using cross-entropy loss.

In experiments, we consider a pre-trained RoBERTa model fine-tuned on MNLI (Williams et al., 2017) corpus as our base entailment model.[9] Further, in order to perform the assignment of logits using an explanation, we maintain meta-information for each explanation to (1) determine if the explanation mentions to 'assign' a label or 'not assign' a label, and (2) track l_{exp} (label mentioned in explanation). For `CLUES-Synthetic`, we parse the templated explanations to obtain the

[9]Weights link: https://huggingface.co/textattack/roberta-base-MNLI

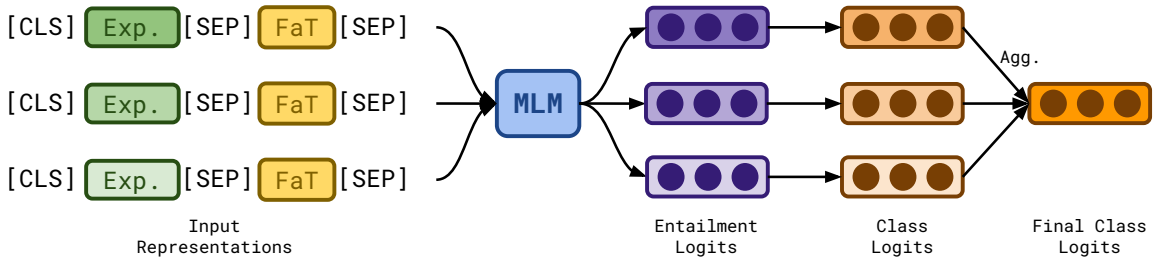


Figure 5: ExEnt takes in concatenated pairs of individual task explanations and features of an example as input and uses a masked language model (MLM) to compute an entailment score for every explanation-feature pair of a task. Next, we map the entailment scores to class logits and finally apply an aggregation function over all the logits to obtain a final class prediction for the example.

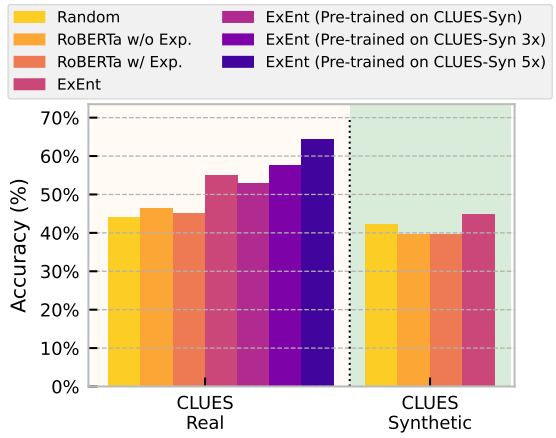


Figure 6: Zero-shot generalization performance of models on novel tasks of CLUES.

meta-information, while for the explanations in CLUES-Real, the authors manually annotate this meta-information. Additional training details and hyperparameters are provided in Appendix E.

5.4 Zero-Shot Generalization Performance

We evaluate ExEnt and the baselines on zero-shot generalization to novel tasks in our benchmark as described in §5.1. We train separate models for CLUES-Real and CLUES-Synthetic. Figure 6 shows the generalization performance of all models. On CLUES, we find that ExEnt outperforms the baselines suggesting that performing entailment as an intermediate step helps aggregate information from multiple explanations better. On CLUES-Real, ExEnt gets an 18% relative improvement over the baselines while having an 11% relative improvement on CLUES-Synthetic

To evaluate the utility of our synthetic tasks in enabling transfer learning to real-world tasks, we fine-tune a ExEnt model pre-trained on synthetic tasks. We experiment with three pre-training task sets - CLUES-Synthetic, CLUES-Synthetic (3x) and CLUES-Synthetic (5x) consisting of 144, 432, and 720 tasks. These larger synthetic task sets are created by sampling tasks from each of the 48 different synthetic tasks types similar to how CLUES-Synthetic was created (see §3.2 for reference). We find that pre-training on synthetic tasks boosts the performance of ExEnt on the novel tasks of CLUES-Real by up to 39% (relative) over the RoBERTa w/o Exp. model.

Human Performance To situate the performance of the automated models, we performed human evaluation for tasks in test split of CLUES-Real using AMT. For this, we sampled at most 50 examples [10] from the test split of tasks in CLUES-Real and each example was 'labeled' by 2 turkers using the explanations of the 'best teacher' (the teacher whose students got the best performance during 'explanation verification' stage; see §3.1.2 for reference). The average human accuracy for this was about 70%. However, the performance numbers of humans and models are not directly comparable as the model looks at all the explanations for the task, whereas the humans observe a small number of explanations. Humans also see multiple examples of the task during the evaluation, which they can use to fine-tune their understanding of a concept. The automated models don't have a mechanism to leverage such data.

6 Key Challenges

To identify key challenges in learning from explanations, we perform experiments ablating the linguistic components and structure of explanations. For a robust analysis, we generate more tasks for each task type in CLUES-Synthetic, making 100 tasks for each of the 48 different task-types in CLUES-Synthetic (axes of variation include 4 negation types, 3 conjunction/disjunction types, 2

[10] Many tasks (such as tasks created from Wikipedia tables) have less than 50 examples in their test split.)

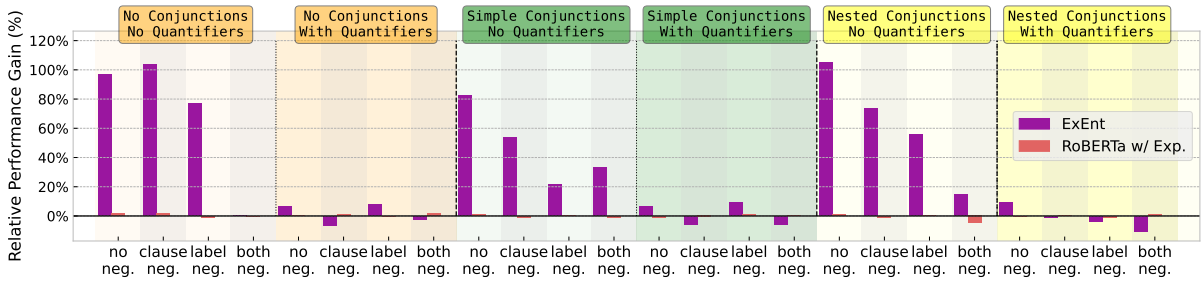


Figure 7: Ablation analysis on the effect of structural and linguistic variations of explanations on generalization ability of models. All bars indicate the relative performance gain over the RoBERTa w/o Exp. baseline.

quantifier types, and number of labels; details in Appendix A.5).

We evaluate the generalization performance of ExEnt to novel tasks on each of the different types separately by training separate models for each task type. Figure 7 shows the relative gain in generalization performance of models learned using explanations compared to the performance of baseline RoBERTa w/o Exp.[11] Our results indicate that learning from explanations containing quantifiers is highly challenging. In the presence of quantifiers, models guided by explanations perform on par with the baseline RoBERTa w/o Exp model. Negations also pose a challenge, as indicated by the decline in relative gains of models guided by explanation compared to the RoBERTa w/o Exp model. Structurally complex explanations (containing conjunctions/disjunctions of clauses) are also hard to learn from compared to simple conditional statements. These challenges provide a fertile ground for future research and improvements.

7 Conclusion

We have introduced CLUES, a benchmark with diverse classification tasks over structured data along with natural language explanations to learn them. CLUES is agnostic in the domain of tasks allowing the research community to contribute more tasks in the future. We also present ExEnt, an entailment-based model to learn classifiers guided by explanations. Our results are promising and indicate that explicitly modeling the role of each explanation through entailment can enable learning classifiers for new tasks from explanations alone. Future work can explore the open challenges in learning from explanations, such as modeling the influence of quantifiers and negations present in an explanation.

Our empirical analyses here aggregates explanations for a task from multiple teachers. Future work can explore learning from explanations from individual teachers, as well as cross-teacher variance. Alternatively, rather than treat explanations from different teachers homogeneously, future work can model trustworthiness of a crowd of teachers from their provided explanations.

Ethics and Broader Impact

All tables in CLUES-Real were collected from free public resources (with required attributions) and tables in CLUES-Synthetic were created by us programmatically. We do not collect any personal information from the turkers who participated in our crowdsourced tasks. The dataset has been released without mentioning any personal details of turkers available automatically in AMT (such as turker IDs). The turkers were compensated fairly and the payment per task is equivalent to an hourly compensation that is greater than minimum wage (based on the median time taken by turkers). We provide details of the reward structure for the crowdsourcing tasks in Appendix D. For the Wikipedia mining task in this work, we limited the locale of eligible turkers to US, UK, New Zealand and Australia. For other crowdsourcing tasks, we limited the locale of eligible turkers to US. Further, to ensure good-faith turkers, we required that the approval rate of the turkers be above 98%. Our screening process has selection biases that likely over-samples turkers from demographics that are over-represented on AMT (ethnically white, college-educated, lower-to-medium income and young) (Hitlin, 2016), and this is likely to affect the type of language usage in the collected explanations.

The broader impact of this research in the longer term could make developing predictive technologies more accessible to ordinary users, rather than data-scientists and experts alone.

[11] Accuracies have been averaged over the multi-class and binary datasets since the trends remain the same across both.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Andreas, Dan Klein, and Sergey Levine. 2018. Learning with latent language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2166–2179, New Orleans, Louisiana. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via Language Model In-context Tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (to appear)*.

Sahil Chopra, Michael Henry Tessler, and Noah D Goodman. 2019. The first crank of the cultural ratchet: Learning and transmitting concepts through language. In *CogSci*, pages 226–232.

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

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1884–1895, Melbourne, Australia. Association for Computational Linguistics.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Paul Hitlin. 2016. 4. turkers in this canvassing: Young, well-educated and frequent users. *Pew Research Center*, 437.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open source scientific tools for Python.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Gideon S. Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11(32):955–984.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (to appear)*.

Shikhar Murty, Pang Wei Koh, and Percy Liang. 2020. ExpBERT: Representation engineering with natural language explanations. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 2106–2113, Online. Association for Computational Linguistics.

Federico Soriano Palacios. 2021. Stroke Prediction Dataset (Retrieved September, 2021).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Proceedings of NeurIPS*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Utkarsh Sharma and Naman Manchanda. 2020. Predicting and improving entrepreneurial competency in university students using machine learning algorithms. In *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 305–309.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2017. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1527–1536, Copenhagen, Denmark. Association for Computational Linguistics.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2018. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 306–316, Melbourne, Australia. Association for Computational Linguistics.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael Tomasello. 1999. *The Cultural Origins of Human Cognition*. Harvard University Press.

Ziqi Wang*, Yujia Qin*, Wenxuan Zhou, Jun Yan, Qinyuan Ye, Leonardo Neves, Zhiyuan Liu, and Xiang Ren. 2020. Learning from explanations with neural execution tree. In *International Conference on Learning Representations*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable nlp. In *Proceedings of NeurIPS*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Qinyuan Ye, Xiao Huang, Elizabeth Boschee, and Xiang Ren. 2020. Teaching machine comprehension with compositional explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1599–1615, Online. Association for Computational Linguistics.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Wangchunshu Zhou, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. In *Advances in Neural Information Processing Systems*, volume 33, pages 6803–6814. Curran Associates, Inc.

Appendix

A Additional details on creating `CLUES-Synthetic`

In this section we discuss in detail about the various table schemas followed by the details of quantifiers and label assignment for creating synthetic tasks.

A.1 Tables schemas

We define five different table schemas, each corresponding to a different domain. For all the attributes in a schema we define a fixed domain from which values for that attribute can be sampled.

- **Species of bird**: The classification task here is to classify a bird into a particular species based on various attributes (column names in table). We define several artificial species of birds using commonly used nonce words in psychological studies (Chopra et al., 2019) such as “dax", “wug", etc.
- **Species of animal**: The classification task here is to classify an animal into a particular species based on various attributes (column names in table). Artificial species of animals are again defined using commonly used nonce words in psychological studies such as “dax", “wug", etc.
- **Rainfall prediction**: This is a binary classification task where the objective is to predict whether it will rain tomorrow based on attributes such as “location", “minimum temperature", “humidity", “atmospheric pressure" etc.
- **Rank in league**: This is a multi-label classification task where given attributes such “win percentage", “power rating", “field goal rating" of a basketball club, the objective is to predict its position in the league out of 1, 2, 3, 4, "Not qualified".
- **Bond relevance**: This is a multi-label classification task where given attributes such “user age", “user knowledge", “user income", the objective is to predict the relevance of a bond out of 5 classes (1 to 5).

In each of the above schemas, the attributes can be either of types categorical or numeral. For each of the above schemas we also define range of admissible values for each attribute. Detailed description of schemas are provided in Tables 8, 9, 10 11, 12.

A.2 List of quantifiers

The full list of quantifiers along with their associated probability values are shown in Table 5.

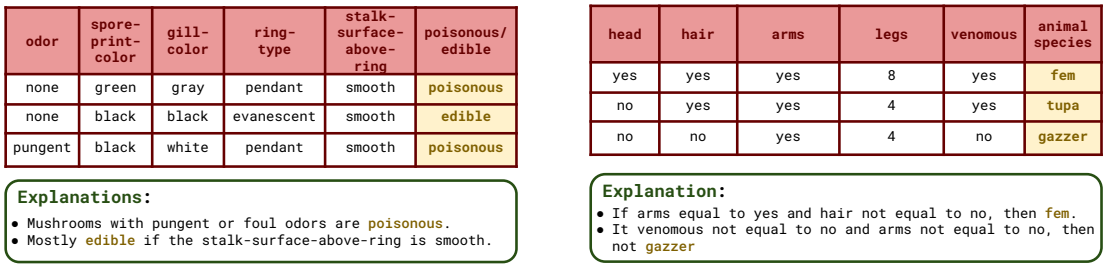
QUANTIFIERS	PROBABILITY
"always", "certainly", "definitely"	0.95
"usually", "normally", "generally", "likely", "typically"	0.70
"often"	0.50
"sometimes", "frequently",	0.30
"occasionally"	0.20
"rarely", "seldom"	0.10
"never"	0.05

Table 5: Probability values used for quantifiers in `CLUES-Synthetic`. We choose these values based on Srivastava et al. (2018).

A.3 Creating synthetic explanations

We use a template-based approach to convert the set to rules into language explanations. We convert every operator in the clauses into their corresponding language format as:

- == → ‘equal to’
- > →‘greater than’
- >= →‘greater than or equal to’
- < →‘lesser than’
- <= →‘lesser than or equal to’
- != → "not equal to’



odor	spore-print-color	gill-color	ring-type	stalk-surface-above-ring	poisonous/edible
none	green	gray	pendant	smooth	poisonous
none	black	black	evanescent	smooth	edible
pungent	black	white	pendant	smooth	poisonous

Explanations:
- Mushrooms with pungent or foul odors are poisonous.
- Mostly edible if the stalk-surface-above-ring is smooth.

(a)

head	hair	arms	legs	venomous	animal species
yes	yes	yes	8	yes	fem
no	yes	yes	4	yes	tupa
no	no	yes	4	no	gazzer

Explanation:
- If arms equal to yes and hair not equal to no, then fem.
- It venomous not equal to no and arms not equal to no, then not gazzer

(b)

Figure 8: Example of tasks from CLUES. The left and right tables are sample tables and explanations drawn from CLUES-Real and CLUES-Synthetic respectively.

- $!>$ → 'not greater than'
- $!<$ → 'not lesser than'

For example if we have a rule '`IF number of hands == 2 THEN foo`', we convert it into a language explanation as 'If number of hands equal to 2, then foo'. In the presence of quantifiers, we add 'it is `[INSERT QUANTIFIER]`' before the label. For example if the rule was associated with a quantifier 'usually', the language explanation would be 'If number of hands equal to 2, then it is usually foo'.

A.4 Label Assignment using Rules

In Algorithm 1, we detail the procedure for obtaining label assignments for our synthetic tasks. Given that our rules are in an "`IF ... THEN ..`" format, we split each rule into an antecedent and a consequent based on the position of `THEN`. Note that our voting-based approach to choose the final label for an example helps to tackle (1) negation on a label for multiclass tasks and (2) choose the most suited label in case antecedents from multiple rules are satisfied by an example.

A.5 Different synthetic task types

We create our synthetic tasks by varying along the following axes:

- Number of labels: $\mathbb{L} = \{$'binary', 'multiclass' $\}$
- Structure of explanation: $\mathbb{C} = \{$'simple', 'conjunction/disjunction', 'nested' $\}$
- Quantifier presence: $\mathbb{Q} = \{$'not present', 'present'$\}$
- Negation: $\mathbb{N} = \{$'no negation', 'negation only in clause', 'negation only on label', 'negation in clause or on label'$\}$

The set of task types is defined as $\mathbb{L} \times \mathbb{C} \times \mathbb{Q} \times \mathbb{N}$, enumerating to 48 different types.

Algorithm 1 Label Assignment

Given: Task $\mathcal{T}$ with rule set R and label set L
Votes $\leftarrow$ Zeros($|L|$)
for rule $r \in R$ **do**
  r_a : Antecedent of r
  r_c : Consequent of r
  $l_r \leftarrow$ Label mentioned in r_c
  $t \leftarrow$ Truth Value of r_a
  if any quantifier in r **then**
    p_{quant} : Prob. of quantifier from Table 5
    Alter l_r to any label in $L \setminus l_r$ with probability
    $1 - p_{quant}$
  end if
  if $t =$ True **then**
    Votes[l_r] += 1
  else
    for label $l \in L \setminus l_r$ **do**
      Votes[l] += 1
    end for
  end if
  $l_{assigned} \leftarrow$ argmax(Votes)
end for

A.6 Large synthetic task collections for ablation experiment

In section §6 we describe an ablation experiment, for which we create collections of 100 tasks corresponding to each synthetic task type. Here, the task type of a collection denotes the maximum complexity of explanations in that collection. For example, for the collection 'multiclass classification with nested clauses and negation only in clause', all the 100 tasks might not have negations or nested clauses in their explanations. This collection might contain explanations with no negations or non-nested clauses. However, it will not contain explanations that have nested clauses and negations in both clause and label.

B Real-World Tasks from UCI, Kaggle and Wikipedia

For our benchmark, we made use of 18 datasets in UCI, 7 datasets in Kaggle, and 9 tables in Wikipedia. In Table 7, we list the keywords that

we use to refer to these tasks along with the URLs to the datasets/tables.

B.1 Feature Selection for Real-World Datasets

During pilot studies for collection of explanations for `CLUES-Real`, we identified that annotators found it difficult to provide explanations for classifications tasks with more than 5 to 6 columns. Appropriately, we reduced the number of columns in most datasets of `CLUES-Real` (apart from some Wikipedia tables) to 5 by choosing the top features that had maximum mutual information with the labels in the training dataset. The mutual information between the features and the label was computed using the scikit-learn package with a random state of 624.

C Additional Analysis on Teacher-Student Performance

For the crowdsourced datasets, we show the number of explanations collected per task in Figure 11(a). The number of explanations is largely around an average value of 11 explanations per task.

Figure 11(b) shows the relation between explanation quality (quantified by likert scores) and rank of the explanation. Rank denotes the order in which a teacher provided that explanation during our crowdsourced explanation collection phase. We find a positive correlation between quality and rank of explanation showing that teachers generally submit most useful explanations (as perceived by them) to teach a task. Finally, we do not observe any correlation between explanation length and ratings as indicated by Figure 11(c).

We also illustrate the differences between teacher and student on our tasks in §4. Here we present two additional plots showing the performance of (1) best teacher vs their students for each task (Figure 9) and (2) worst teacher vs their students for each task (Figure 10). We find that even though the best teachers often attain near-perfect accuracies for the tasks, their students perform significantly worse than them in many tasks. The explanations from the worst teachers did not help students in getting significantly better than random performance for majority of the tasks, even though the student did outperform the worst teacher.

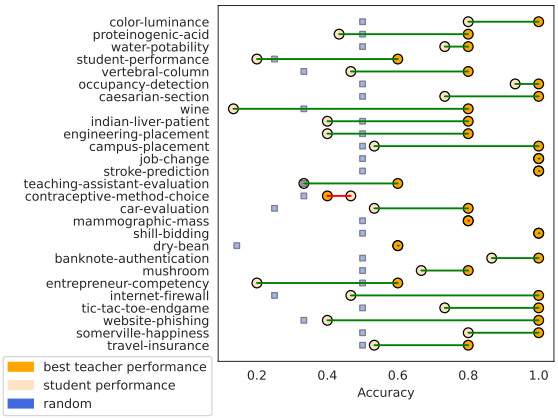


Figure 9: Best teacher vs average of their students for tasks in `CLUES-Real`. Red lines indicate cases where the student performance is more than the teacher performance. Green lines indicate cases where teachers perform better than students.

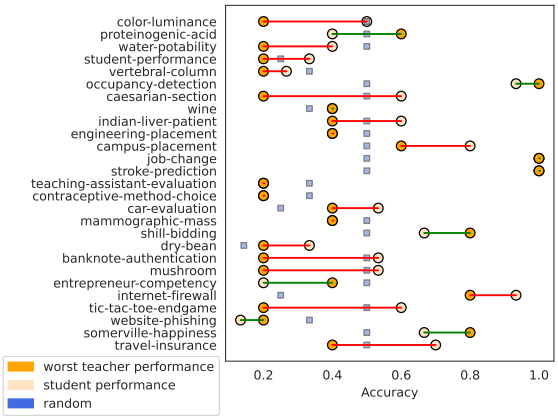
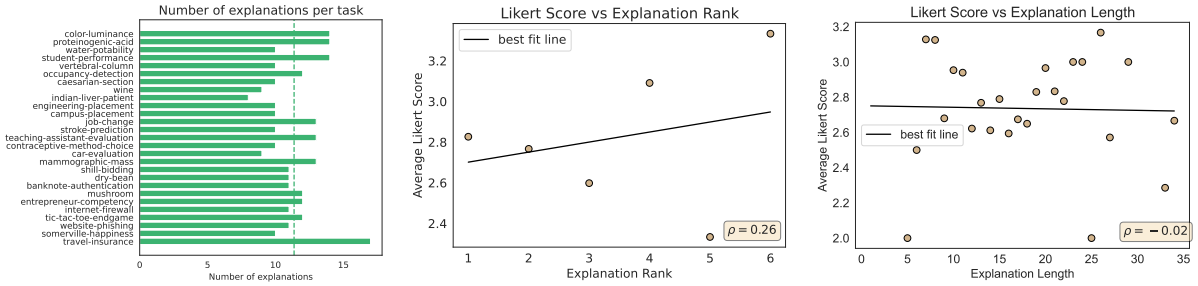


Figure 10: Worst teacher vs average of their students for tasks in `CLUES-Real`. Red lines indicate cases where the student performance is more than the teacher performance. Green lines indicate cases where teachers perform better than students.

D Reward Structure for Crowd-sourcing Tasks

Our work involves multiple stages of crowdsourcing to collect high-quality explanations for the classification tasks. We pick turkers in the US for explanation collection and verification tasks (US,UK,NZ, and GB for Wikipedia mining Task) with a 98% HIT approval rate and a minimum of 1000 HITs approved. In Table 6, we summarize the payment structure provided to the turkers on the AMT platform for each of the stages (described in detail in §3) – (1) Wikipedia mining on tables scraped from Wikipedia, (2) Explanation collection for tables obtained from UCI, Kaggle and Wikipedia, and



(a) (b) (c)

Figure 11: (a) On Average we obtain over 10 explanations per task in CLUES-Real for tasks that are crowdsourced (b) Weak positive correlation indicating later explanations were given higher likert scores by students. Likert ratings were averaged for each rank. (c) Near-zero correlation indicating that likert ratings given by students were almost independent of explanation length. Likert ratings were averaged for each length. (ρ denotes Pearson correlation coefficient in each of the plots)

(3) Explanation validation for collected explanations. For all the three crowdsourcing tasks, the turkers were compensated fairly and the payment per task is equivalent to an hourly compensation that is greater than minimum wage (based on the median time taken by turkers).

STAGE	$/HIT	BONUS
Wikipedia Mining	$3	$3-$4 [12]
Explanation Collection	$5.5	-
Explanation Validation	$1.2	-

Table 6: Payment structure for AMT Tasks

E Training details

In this section we proved details about implementation of various models, hyperparameter details, and details about hardware and software used along with an estimate of time taken to train the models. Code and dataset for our paper will be made public upon first publication.

E.1 Details of seen and novel tasks for CLUES-Real and CLUES-Synthetic

For CLUES-Real, we chose the tasks from Wikipedia that have very examples to be part of novel task set. Among the tasks from Kaggle and UCI, we kept tasks with higher number of samples as part of seen tasks (training tasks). Seen tasks (20) for CLUES-Real are:

- website-phishing
- internet-firewall
- mushroom
- dry-bean
- wine
- caesarian-section
- occupancy-detection
- vertebral-column
- student-performance
- shill-bidding
- mammographic-mass
- teaching-assistant-evaluation
- somerville-happiness
- stroke-prediction
- job-change
- campus-placement
- engineering-placement
- water-potability
- color-luminance
- proteinogenic-acid

Novel tasks (16) for CLUES-Real are:

- banknote-authentication
- tic-tac-toe-endgame
- car-evaluation
- contraceptive-method-choice
- indian-liver-patient
- travel-insurance
- entrepreneur-competency
- award-nomination-result
- coin-face-value
- coin-metal
- driving-championship-points
- election-outcome
- hotel-rating
- manifold-orientability
- soccer-club-region
- soccer-league-type

We train on 70% of the labeled examples of the seen tasks and perform zero-shot generaliza-

[12] ¢50 per table submitted

tion test over the 20% examples of each task in CLUES-Real. For the extremely small Wikipedia tasks (for which we do not crowdsource explanations), we use all examples for zero-shot testing.

For CLUES-Synthetic, we have 96 tasks as seen (training) tasks and 48 as novel tasks. Task in CLUES-Synthetic that belong to the following schemas are part of the seen tasks:

- Species of Animal
- Species of Bird
- Rainfall prediction

Tasks belonging to 'Bond relevance classification' and 'League Rank Classification' were part of novel tasks for CLUES-Synthetic. We train on 700 labeled examples of each seen task and perform zero-shot generalization test over 200 examples of each novel task in CLUES-Synthetic.

E.2 Model parameters

- RoBERTa w/o Exp.: The number of parameters is same as the pretrained RoBERTa-base model available on HuggingFace library.
- RoBERTa w/ Exp.: The number of parameters is same as the pretrained RoBERTa-base model available on HuggingFace library.
- ExEnt: The number of parameters is same as the pre-trained RoBERTa mdoel finetuned on MNLI (Williams et al., 2017) corpus. We obtain the pretrained checkpoint from HuggingFace.[13]

E.3 Hyper-parameter settings

For all the transformer based models we use the implementation of HuggingFace library (Wolf et al., 2020). All the model based hyper-parameters are thus kept default to the settings in the HuggingFace library. We use the publicly available checkpoints to initialise the pre-trained models. For RoBERTa based baselines we use 'roberta-base' checkpoint available on HuggingFace. For our intermediate entailment model in ExEnt, we finetune a pretrained checkpoint of RoBERTa trained on MNLI corpus ('textattack/roberta-base-MNLI')

When training on CLUES-Synthetic, we use a maximum of 64 tokens for our baseline RoBERTa w/o Exp. and ExEnt. For the RoBERTa w/ Exp. model we increase this limit to 128 tokens as it takes concatenation of all explanations for a task. When training on CLUES-Real, we use 256 tokens as limit for RoBERTa w/ Exp. using explanations as the real-world tasks have roughly two times more explanations on average than synthetic tasks.

We used the AdamW (Loshchilov and Hutter, 2019) optimizer commonly used to fine-tune pre-trained Masked Language Models (MLM) models. For fine-tuning the pre-trained models on our benchmark tasks, we experimented with learning rates $\{1e-5, 2e-5\}$ and chose $1e-5$ based on performance on the performance on the validation set of seen tasks. Batch sizes was kept as 2 with gradient accumulation factor of 8. The random seed for all experiments was 42. We train all the models for 20 epochs. Each epoch comprises of 100 batches, and in each batch the models look at one of the tasks in the seen split.

E.4 Hardware and software specifications

All the models are coded using Pytorch 1.4.0[14] (Paszke et al., 2019) and related libraries like numpy (Harris et al., 2020), scipy (Jones et al., 2001–) etc. We run all experiments on one of the following two systems - (1) GeForce RTX 2080 GPU of size 12 GB, 256 GB RAM and 40 CPU cores (2) Tesla V100-SXM2 GPU of size 16GB, 250 GB RAM and 40 CPU cores.

E.5 Training times

- Training on CLUES-Real: The baseline RoBERTa w/o Exp model typically takes 3 seconds on average for training on 1 batch of examples. In 1 batch, the model goes through 16 examples from the tasks in seen split. RoBERTa w/ Exp. takes around 5 seconds to train on 1 batch. ExEnt takes longer time than baselines owing to the multiple forward passes. For training on 1 batch of CLUES-Real, ExEnt took 12 seconds on average.
- Training on CLUES-Synthetic: All the models take comparatively much lesser time for training on our synthetic tasks owing to lesser number of explanations on average for a task. For training on 1 batch, all models took 1 seconds or less to train on 1 batch of examples from CLUES-Synthetic.

F Annotation interfaces

We present the different annotation templates and interfaces used for our explanation collection and verification stages in Figures 12,13,14,15 and Figure 16 respectively.

[13] Weights link: https://huggingface.co/textattack/roberta-base-MNLI

[14] https://pytorch.org/

DATASET	SOURCE	URL	CROWD-SOURCED
car-evaluation	UCI	https://archive.ics.uci.edu/ml/datasets/Car+Evaluation	YES
indian-liver-patient	UCI	https://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29	YES
bank-note-authentication	UCI	http://archive.ics.uci.edu/ml/datasets/banknote+authentication	YES
contraceptive-method-choice	UCI	http://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice	YES
mushroom	UCI	http://archive.ics.uci.edu/ml/datasets/Mushroom	YES
mammographic-mass	UCI	https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass	YES
wine	UCI	http://archive.ics.uci.edu/ml/datasets/Wine	YES
teaching-assistant-evaluation	UCI	https://archive.ics.uci.edu/ml/datasets/Teaching+Assistant+Evaluation	YES
shill-bidding	UCI	https://archive.ics.uci.edu/ml/datasets/Shill+Bidding+Dataset	YES
website-phishing	UCI	https://archive.ics.uci.edu/ml/datasets/Website+Phishing	YES
tic-tac-toe-endgame	UCI	https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame	YES
somerville-happiness	UCI	https://archive.ics.uci.edu/ml/datasets/Somerville+Happiness+Survey	YES
occupancy-detection	UCI	https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+	YES
vertebral-column	UCI	https://archive.ics.uci.edu/ml/datasets/Vertebral+Column	YES
caesarian-section	UCI	https://archive.ics.uci.edu/ml/datasets/Caesarian+Section+Classification+Dataset	YES
student-performance	UCI	https://archive.ics.uci.edu/ml/datasets/Student+Performance+on+an+entrance+examination	YES
dry-bean	UCI	https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset	YES
internet-firewall	UCI	https://archive.ics.uci.edu/ml/datasets/Internet+Firewall+Data	YES
campus-placement	Kaggle	https://www.kaggle.com/benroshan/factors-affecting-campus-placement	YES
job-change	Kaggle	https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists?select=aug_train.csv	YES
water-potability	Kaggle	https://www.kaggle.com/adityakadiwal/water-potability	YES
stroke-prediction	Kaggle	https://www.kaggle.com/fedesoriano/stroke-prediction-dataset	YES
engineering-placement	Kaggle	https://www.kaggle.com/tejashvi14/engineering-placements-prediction	YES
travel-insurance	Kaggle	https://www.kaggle.com/tejashvi14/travel-insurance-prediction-data	YES
entrepreneur-competency	Kaggle	https://www.kaggle.com/namanmanchanda/entrepreneurial-competency-in-university-students	YES
soccer-league-type	Wikipedia	https://en.wikipedia.org/wiki/Oklahoma	NO
soccer-club-region	Wikipedia	https://en.wikipedia.org/wiki/Oklahoma	NO
hotel-rating	Wikipedia	https://en.wikipedia.org/wiki/Disneyland_Paris	NO
coin-face-value	Wikipedia	https://en.wikipedia.org/wiki/Coins_of_the_United_States_dollar	NO
coin-metal	Wikipedia	https://en.wikipedia.org/wiki/Coins_of_the_United_States_dollar	NO
election-outcome	Wikipedia	https://en.wikipedia.org/wiki/Kuomintang	NO
driving-championship-points	Wikipedia	https://en.wikipedia.org/wiki/Judd_(engine)	NO
manifold-orientability	Wikipedia	https://en.wikipedia.org/wiki/Homology_(mathematics)	NO
award-nomination-result	Wikipedia	https://en.wikipedia.org/wiki/When_Harry_Met_Sally...	NO
color-luminance	Wikipedia	https://en.wikipedia.org/wiki/Hue	YES
proteinogenic-acid	Wikipedia	https://en.wikipedia.org/wiki/Miller%E2%80%93Urey_experiment	YES

Table 7: List of datasets and URLs that make up CLUES-Real.

```
    "description": "This dataset is used to predict the type of birds based on the
        given attributes. Each row provides the relevant attributes of a bird.",
        "column_names":{
            "size" : ["categorical", ["large", "medium", "small"]],
            "size (number)" : ["number", [10, 100]],
            "color" : ["categorical", ["red", "blue", "green", "brown", "pink", "
                orange", "black", "white"]],
            "head" : ["categorical", ["yes", "no"]],
            "length" : ["categorical", ["tall", "medium", "short"]],
            "length (number)" : ["number", [10,100]],
            "tail" : ["categorical", ["yes", "no"]],
            "number of faces" : ["number", [1,3]],
            "arms" : ["categorical", ["yes", "no"]],
            "legs" : ["categorical", [2, 4, 6, 8]],
            "hair" : ["categorical", ["yes", "no"]],
            "wings" : ["categorical", ["yes", "no"]],
            "feathers" : ["categorical", ["yes", "no"]],
            "airborne" : ["categorical", ["yes", "no"]],
            "toothed" : ["categorical", ["yes", "no"]],
            "backbone" : ["categorical", ["yes", "no"]],
            "venomous" : ["categorical", ["yes", "no"]],
            "domestic" : ["categorical", ["yes", "no"]],
            "region": ["categorical", ["asia", "europe", "americas", "africas", "
                antartica", "oceania"]]
        },
        "targets": {
            "bird species": ["wug", "blicket", "dax", "toma", "pimwit", "zav", "
                speff", "tulver", "gazzer", "fem", "fendle", "tupa"]
        }
}
```

Table 8: Synthetic table schema 1: Species of Birds

```
{
    "description": "This dataset is used to predict the type of an aquatic animal
        based on the given attributes. Each row provides the relevant attributes of
        an animal.",
        "column_names":{
            "size" : ["categorical", ["large", "medium", "small"]],
            "size (number)" : ["number", [10, 100]],
            "color" : ["categorical", ["red", "blue", "green", "brown", "pink", "
                orange", "black", "white"]],
            "head" : ["categorical", ["yes", "no"]],
            "length" : ["categorical", ["tall", "medium", "short"]],
            "length (number)" : ["number", [10,100]],
            "tail" : ["categorical", ["yes", "no"]],
            "number of faces" : ["number", [1,3]],
            "arms" : ["categorical", ["yes", "no"]],
            "legs" : ["categorical", ["yes", "no"]],
            "hair" : ["categorical", ["yes", "no"]],
            "fins" : ["categorical", ["yes", "no"]],
            "toothed" : ["categorical", ["yes", "no"]],
            "venomous" : ["categorical", ["yes", "no"]],
            "domestic" : ["categorical", ["yes", "no"]],
            "region": ["categorical", ["atlantic", "pacific", "indian", "arctic"]]
        },
        "targets": {
            "animal species": ["wug", "blicket", "dax", "toma", "pimwit", "zav", "
                speff", "tulver", "gazzer", "fem", "fendle", "tupa"]
        }
}
```

Table 9: Synthetic table schema 2: Species of Animal

```
{
   "description": "This dataset is used to predict if it will rain tomorrow or not
       based on the given attributes. Each row provides the relevant attributes of a
        day.",
        "column_names":{
            "location" : ["categorical", ["sphinx", "doshtown", "kookaberra", "
                shtick union", "dysyen"]],
            "mintemp": ["number", [1,15]],
            "maxtemp": ["number", [17,35]],
            "rainfall today": ["categorical", [0, 0.2, 0.4, 0.6, 0.8, 1]],
            "hours of sunshine": ["categorical", [0, 4, 8, 12]],
            "humidity": ["number", [0,100]],
            "wind direction": ["categorical", ["N", "S", "E", "W", "NW", "NE", "SE",
                 "SW"]],
            "wind speed": ["number", [10,85]],
            "atmospheric pressure": ["number", [950,1050]]
        },
        "targets": {
            "rain tomorrow": ["yes", "no"]
        }
}
```

Table 10: Synthetic table schema 3: Rainfall Prediction

```
{
    "description": "This dataset is used to predict the final league position of a
        team based on the given attributes. Each row provides the relevant
        attributes of a team.",
        "column_names":{
            "win percentage":["number", [0,100]],
            "adjusted offensive efficiency": ["number", [0,100]],
            "adjusted defensive efficiency": ["number", [0,100]],
            "power rating": ["categorical", [1,2,3,4,5]],
            "turnover percentage": ["number", [0,100]],
            "field goal rating": ["categorical", [1,2,3,4,5]],
            "free throw rating": ["categorical", [1,2,3,4,5]],
            "two point shoot percentage": ["number", [0,100]],
            "three point shoot percentage": ["number", [0,100]]
        },
        "targets": {
            "final position": ["1", "2", "3", "4", "Not Qualified"]
        }
}
```

Table 11: Synthetic table schema 4: League Ranking Classification

```
{
   "description": "This dataset is used to predict the relevance (higher the better)
        of a bond to a user based on the given attributes. Each row provides the
       relevant attributes of a user.",
        "column_names":{
            "user age":["number", [15,65]],
            "user knowledge": ["categorical", [1,2,3,4,5]],
            "user gender": ["categorical", ["male", "female"]],
            "user loyalty": ["categorical", [1,2,3,4,5]],
            "user income": ["number", [1000,10000]],
            "user marital status": ["categorical", ["yes", "no"]],
            "user dependents": ["number", [0,3]]
        },
        "targets": {
            "relevance score": ["1", "2", "3", "4", "5"]
        }
}
```

Table 12: Synthetic table schema 5: Bond Relevance Classification

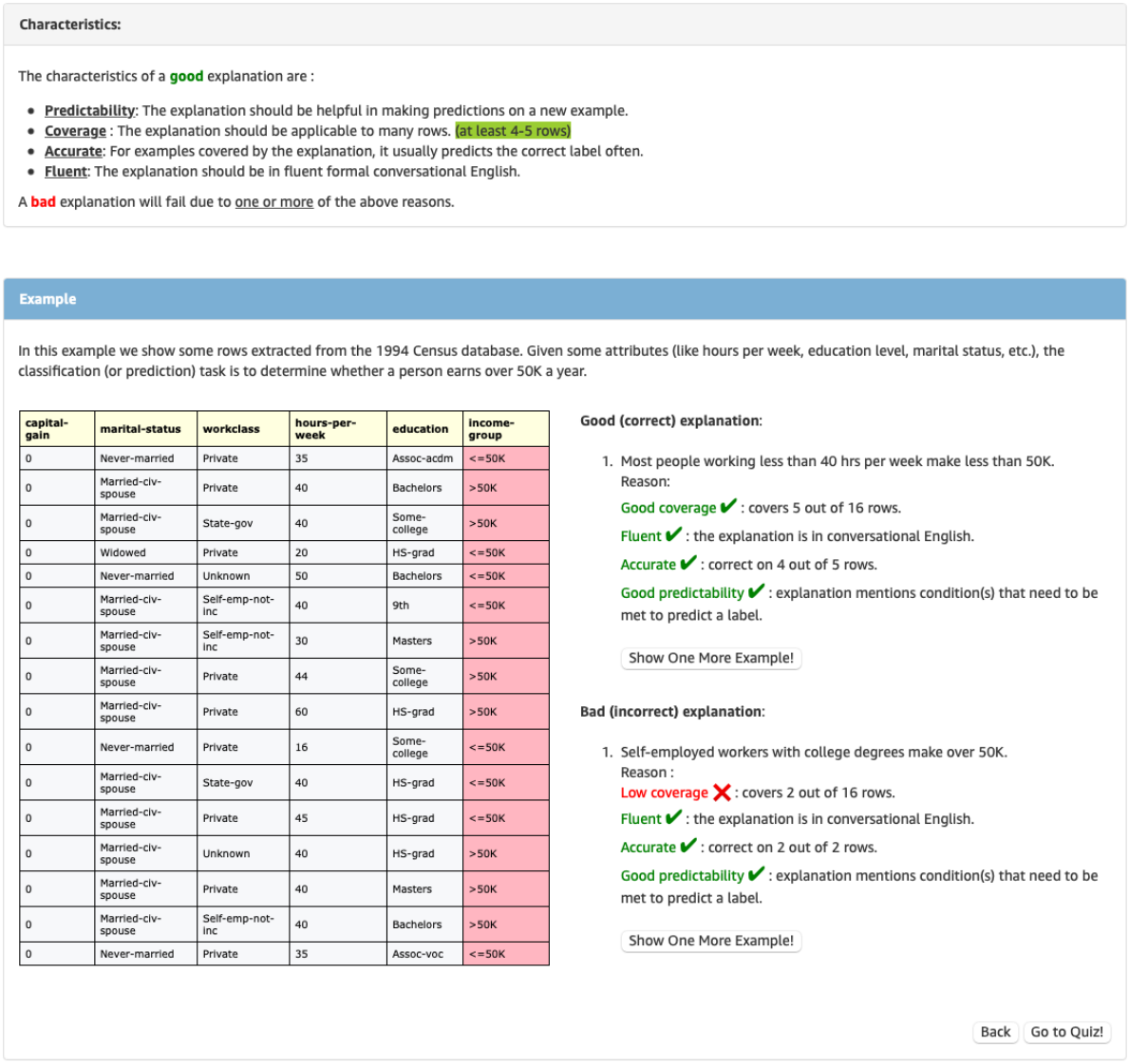


Figure 12: Explanation Collection: Annotation Task Examples page.

Characteristics:

The characteristics of a **good** explanation are :

- **Predictability**: The explanation should be helpful in making predictions on a new example.
- **Coverage** : The explanation should be applicable to many rows. (at least 4-5 rows)
- **Accurate**: For examples covered by the explanation, it usually predicts the correct label often.
- **Fluent**: The explanation should be in fluent formal conversational English.

A **bad** explanation will fail due to one or more of the above reasons.

Qualification Quiz

Looking at the table below (Income table), go through each of the explanations below and mark whether they are "good" or "bad".

In this example we show some rows extracted from the 1994 Census database. Given some attributes (like hours per week, education level, marital status, etc.), the classification (or prediction) task is to determine whether a person earns over 50K a year.

capital-gain	marital-status	workclass	hours-per-week	education	income-group
0	Never-married	Private	35	Assoc-acdm	<=50K
0	Married-civ-spouse	Private	40	Bachelors	>50K
0	Married-civ-spouse	State-gov	40	Some-college	>50K
0	Widowed	Private	20	HS-grad	<=50K
0	Never-married	Unknown	50	Bachelors	<=50K
0	Married-civ-spouse	Self-emp-not-inc	40	9th	<=50K
0	Married-civ-spouse	Self-emp-not-inc	30	Masters	>50K
0	Married-civ-spouse	Private	44	Some-college	>50K
0	Married-civ-spouse	Private	60	HS-grad	>50K
0	Never-married	Private	16	Some-college	<=50K
0	Married-civ-spouse	State-gov	40	HS-grad	<=50K
0	Married-civ-spouse	Private	45	HS-grad	<=50K
0	Married-civ-spouse	Unknown	40	HS-grad	>50K
0	Married-civ-spouse	Private	40	Masters	>50K
0	Married-civ-spouse	Self-emp-not-inc	40	Bachelors	>50K
0	Never-married	Private	35	Assoc-voc	<=50K

Explanation 1 : Married employees are likely to earn more than 50K while never married employees generally earn less than or equal to 50K.
Good Bad

Explanation 2 : Only being a high school graduate generally ensures more than 50K annual income.
Good Bad

The following two explanations are bad. Please select the characteristics that fail with these explanations:

Explanation 3 : The last column gives the label of the income group.
Predictability Coverage Accuracy Fluency

Explanation 4 : If hours-per-week < 40, then income-group <=50K.
Predictability Coverage Accuracy Fluency

See Examples Again
Verify Answers

Provide qualification task feedback below

Rate your understanding on the characteristics of good explanations (1 - not clear, 5 - confident) :

Mention any other characteristic (along with its one line description) that you would want to see in a 'good' explanation (OPTIONAL)

Give additional feedback about the experience here (OPTIONAL)

Next

Figure 13: Explanation Collection: Qualification Task page.

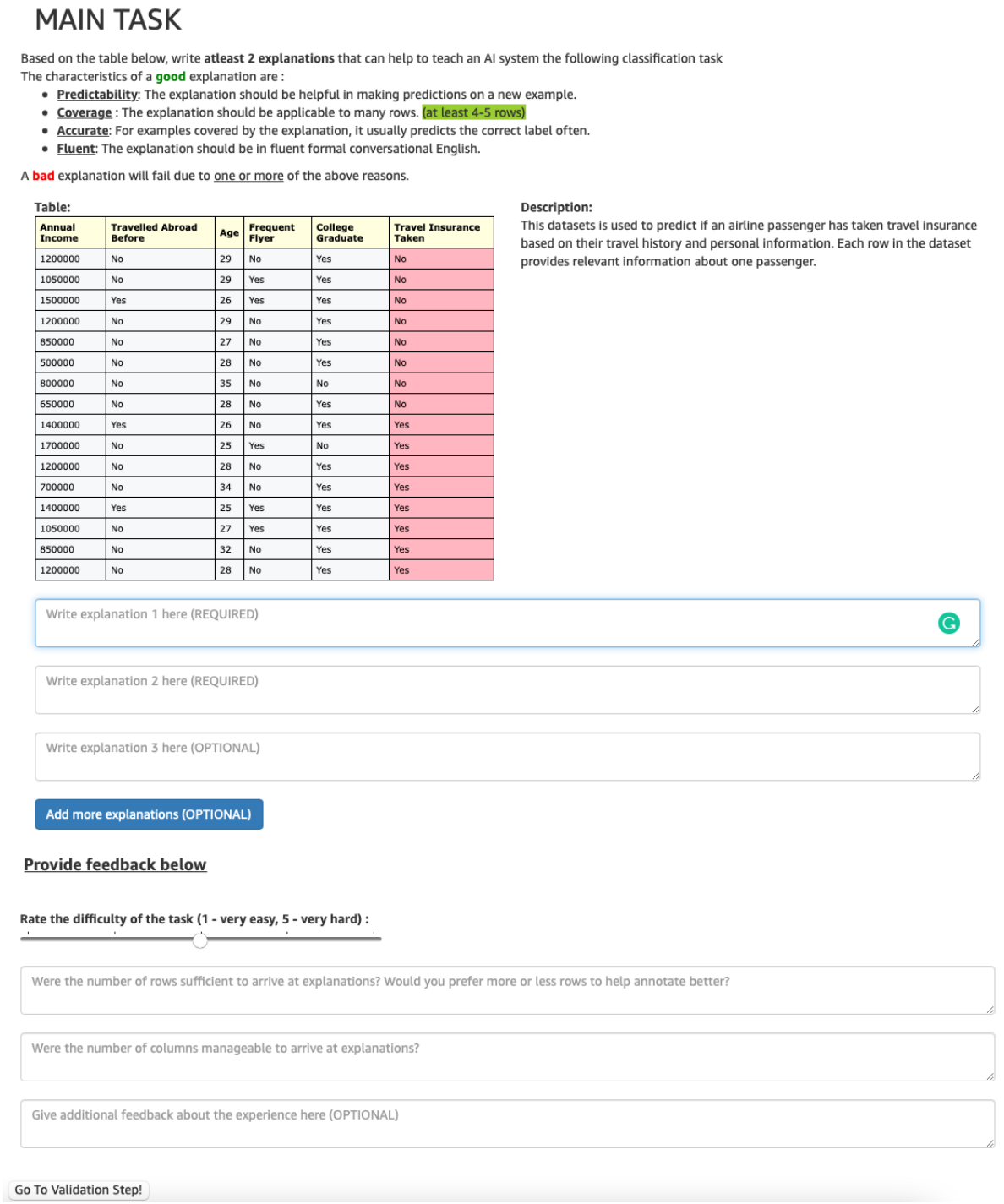


Figure 14: Explanation Collection: Main Task page.

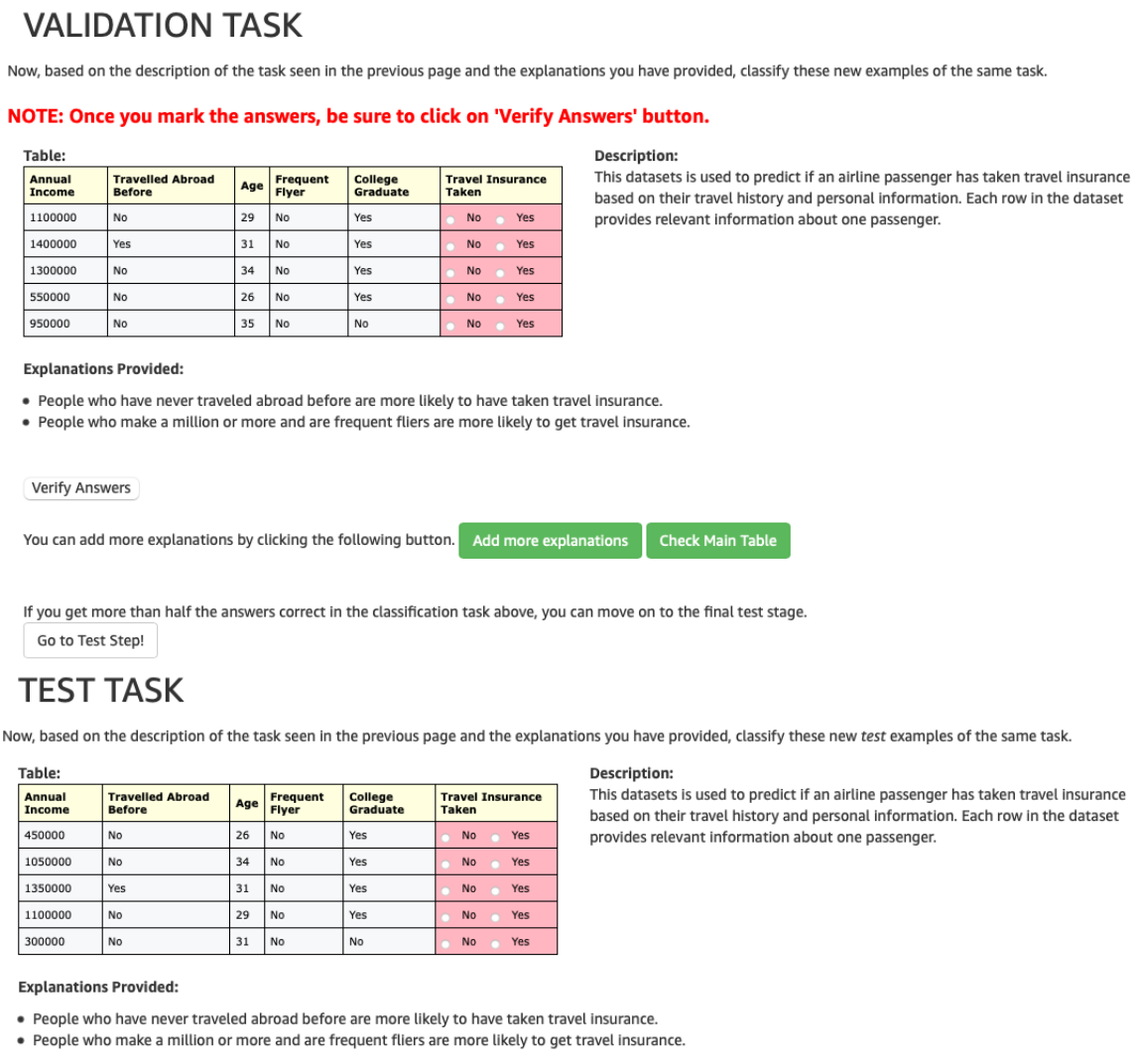


Figure 15: Explanation Collection: Validation and Test page.

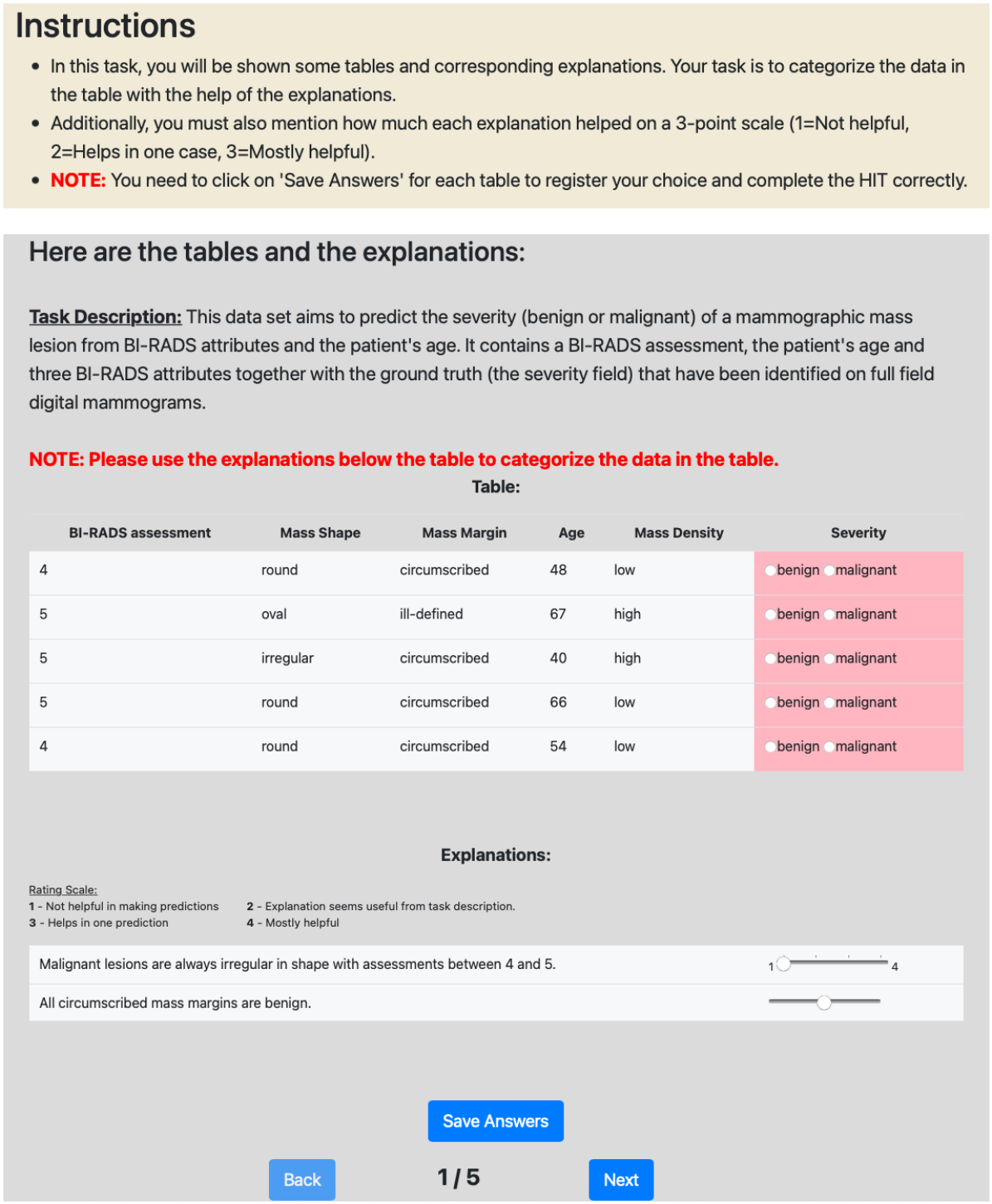


Figure 16: Explanation Verification page.