CLUES: A Benchmark for Learning Classifiers using Natural Language Explanations

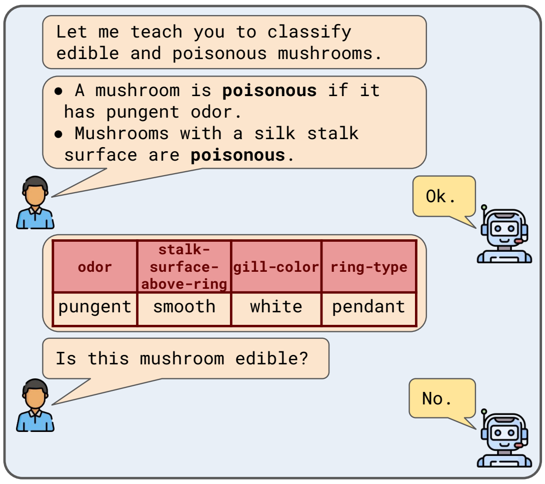
Rakesh R Menon[*] **Sayan Ghosh**[*] **Shashank Srivastava**
UNC Chapel Hill
{rrmenon, sayghosh, ssrivastava}@cs.unc.edu

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 training classifiers for structured data[1] *purely* from language. 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 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 develop ExEnt, an entailment-based model that learns classifiers using explanations. ExEnt generalizes up to 18% better (relative) on novel tasks than a baseline that does not use explanations. 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 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.


Let me teach you to classify edible and poisonous mushrooms.
• A mushroom is **poisonous** if it has pungent odor.
• Mushrooms with a silk stalk surface are **poisonous**.
Ok.
Is this mushroom edible?
No.

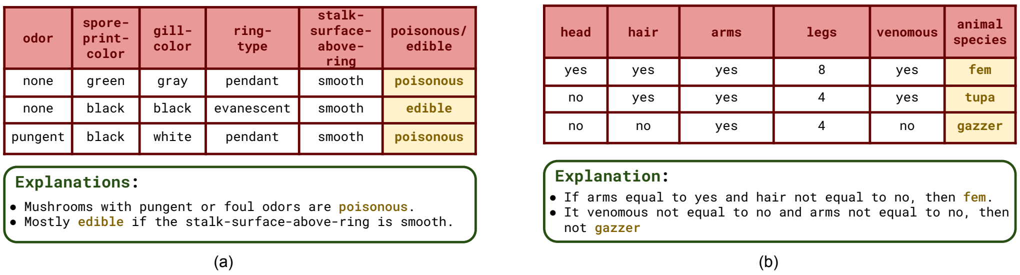
odor	stalk-surface-above-ring	gill-color	ring-type
pungent	smooth	white	pendant

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 datasets for which we collect explanations in CLUES.

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, *inter alia*) 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); Murty et al. (2020), 7 email categorization tasks (Srivastava et al., 2017)). Owing to the paucity of large-scale benchmarks for learning from explanations over diverse tasks, we develop CLUES, a benchmark of classification tasks paired with natural language explanations. 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 test

[*] Equal contribution

[1] By structured data, we refer to data that can be represented using tables (e.g., spreadsheets, traditional classification datasets in CSV format, single-table databases).



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

models' reasoning ability under a range of structural and linguistic modifications of explanations.

In addition to creating CLUES, 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. However, we notice that simply concatenating explanations to the input does not help pre-trained models, like RoBERTa (Liu et al., 2019), generalize. Hence, 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 novel tasks.

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.

2 Creating CLUES

In CLUES, we frame classification tasks over structured data represented in tabular format. We consider two splits of our benchmark, CLUES-Real (real-world datasets) and CLUES-Synthetic (synthetic datasets).

2.1 CLUES-Real

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

Collecting datasets: We choose 18 tabular classification tasks from UCI ML repository[2] and 7 from Kaggle[3]. Additionally, we mine suitable tables from Wikipedia that can be posed as classification tasks. We formalize the mining process as a crowdsourcing task (details in Appendix B.2). We identified 11 classification tasks corresponding to 9 Wikipedia tables after mining around 10K Wikipedia tables. The details of tasks in CLUES-Real are provided in Appendix B.

Explanation Collection Pipeline:

1. 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 task and ask them to provide a set of explanations corresponding to the task. The turkers participating in this task have been vetted by a qualification task to test their understanding of 'good' and 'bad' explanations (see Appendix H for templates). After providing explanations, the turkers predict classes for a set of unlabeled samples by using their explanations.[4] The turkers in this stage are henceforth referred as 'teachers' in our setup since they provide explanations to 'teach' models about different classification tasks.
2. EXPLANATION VERIFICATION: We validate the utility of the explanations for a task from each teacher by evaluating if they are useful for other humans in learning the task. We perform explanation verification on AMT as well, where we ask a new set of turkers to label a set of unlabeled samples from a task using the explanations provided by individual teachers in the 'explanation collection' phase. Additionally, we ask the turkers 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 henceforth.

[2] https://archive.ics.uci.edu/ml/

[3] https://www.kaggle.com/datasets

[4] Details of our collection template along with information on payment is provided in Appendices D and H.

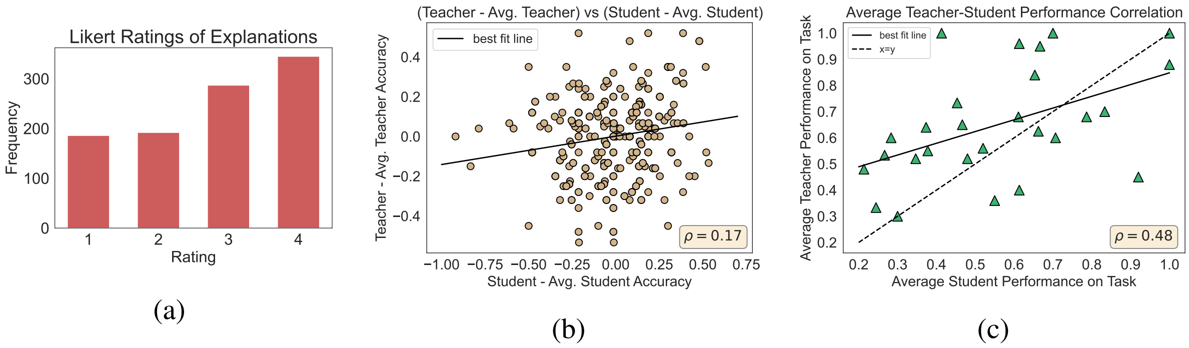


Figure 3: (a) Histogram of count of explanations corresponding to different usefulness likert ratings. (b) Students typically perform well when taught tasks by good teachers. (c) Positive correlation in the average performance between a teacher and student for a task. (ρ denotes Pearson correlation coefficient in each of the plots)

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.

2.2 CLUES-Synthetic

To delineate challenges in learning from explanations under controlled settings, 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 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): for example, IF c_1 THEN NOT l

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

3 Dataset analysis

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

Task Statistics: Table 1 shows the statistics of tasks in CLUES. Task details are provided in Appendices A and B. An aggregate of 133 teachers provide 318 explanations for tasks in CLUES-Real. All collected explanations were manually filtered and irrelevant explanations were removed. Further, each explanation set corresponding to a teacher in CLUES-Real was verified by 3 students.

Lexical analysis of explanations: Using the spacy tokenizer, we find the vocabulary size of explanations in CLUES as 1026 tokens resulting in 15.53 tokens on average per explanation. The median reading complexity of the explanations is 65.73 (8th/9th-grade reading level)[5].

Usefulness of the explanations: During explanation verification, turkers provide a rating (on a Likert scale from 1 to 4) on the utility of the explanations for classification (1 - 'not helpful' ; 4 - 'denotes mostly helpful in prediction'). The average rating for the explanations in CLUES-Real is 2.78, denoting most explanations were useful.

Characteristics of teachers and students: Figure 3(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 (ρ = 0.17) suggests that students tend

[5] computed using Flesch reading test

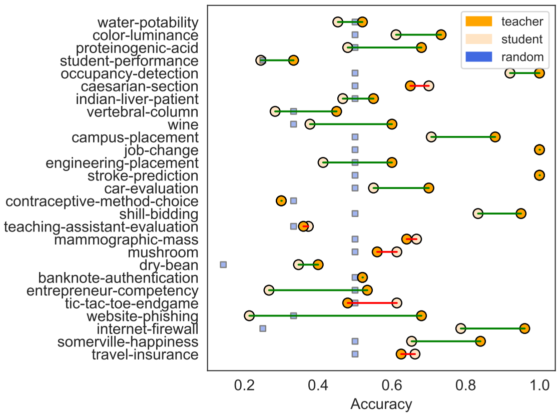


Figure 4: Average student vs average teacher performance for tasks in CLUES-Real. Red lines indicate cases where the student performance is more than the teacher performance. Green lines indicate cases where teachers perform better than students.

to perform well if taught by well-performing teachers. Figure 4 shows average teacher and avergage student performance for each task in CLUES-Real. We find that an average teacher performs better than the average student on most tasks, barring a few tasks such as 'tic-tac-toe', which could be solved using commonsense without relying on explanations. Please refer to Appendix C for additional data analysis and insights on CLUES.

4 Experiment Setup and Models

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

4.1 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. 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. Following linearization techniques for structured data Yin et al. (2020), we adapt a similar encoding process wherein 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. We will refer to the linearized format of structured inputs by 'Features-as-Text' or 'FaT'.

4.2 Baseline models

We consider the following baselines:

- RoBERTa w/o Exp. : This is a RoBERTa-based (Liu et al., 2019) baseline that does not utilize the explanations to train a classifier. The pre-trained RoBERTa model takes the linearized structured data (FaT) as input and outputs a representation for this context (using the [CLS] token). Next, we run another forward pass using RoBERTa to obtain a representation of the labels based on their text. 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.
- RoBERTa w/ Exp.: This follows the same training setup as RoBERTa w/o Exp. However, in addition to FaT, this model also takes a concatenation of explanations as the input to RoBERTa.

4.3 ExEnt

We empirically notice that simply concatenating explanations to the input does not help pre-trained models, like RoBERTa generalize. In order to better model the influence of an explanation towards deciding a class label, we draw analogies with the entailment of an explanation (*hypothesis*) towards the structured input (*premise*). 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 provides entailment (p_e), contradiction (p_c) and neutral (p_n) logits which are then assigned to the class labels as:

- **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 not 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.

Additional training details and hyperparameters

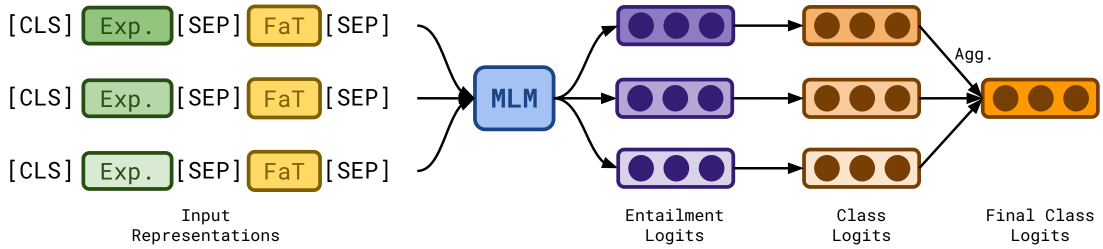


Figure 5: ExEnt takes in concatenated pairs of individual task explanations and features of an example as input and uses a masked language model (MLM) to compute an entailment score for every pair. Next, we map the entailment scores to class logits and finally aggregate over all the logits to obtain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

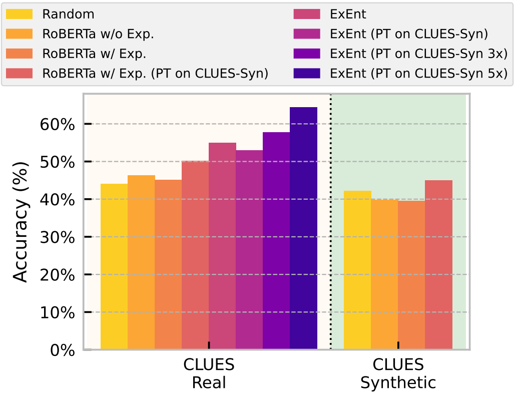


Figure 6: Zero-shot generalization performance of models on novel tasks of CLUES. PT = PRE-TRAINED

are provided in Appendix E.

4.4 Zero-Shot Generalization Performance

Figure 6 shows the generalization performance of all models on CLUES. 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 §2.2 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. Additional experiments on synthetic tasks also reveal that ExEnt struggles with handling negations, conjunctions and disjunctions when learning from explanations (details in Appendix G).

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[6] 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 §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.

5 Conclusion

We introduce CLUES, a benchmark with diverse classification tasks over structured data using natural language explanations to test the ability of models to learn novel classification tasks purely from language. Additionally, we introduce ExEnt, an entailment-based model to learn classifiers guided by explanations. Our results indicate that explicitly modeling the role of each explanation through entailment can enable learning classifiers for new tasks from explanations. Future work can explore open challenges such as modeling quantifiers and negations present in an explanation. CLUES is agnostic in the domain of tasks allowing the research community to contribute more tasks in the future.

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

References

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.

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.

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

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*.

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.

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.

Michael Tomasello. 1999. The cultural origins of human cognition.

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.

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.

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 5, 6, 7 8, 9.

A.2 List of quantifiers

The full list of quantifiers along with their associated probability values are shown in Table 2.

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 2: 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’
- $!>$ → ‘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’ $\}$

Algorithm 1 Label Assignment

```
Given: Task $\mathcal{T}$ with rule set $R$ and label set $L$
Votes ← 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 2
        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
```

- Structure of explanation: $\mathbb{C}$ = {'simple', 'conjunction/disjunction', 'nested' }
- Presence of quantifier: $\mathbb{Q}$ = {'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.

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 4, we list the keywords that we use to refer to these tasks along with the URLs to the datasets/tables.

B.1 Selecting tasks from UCI and Kaggle

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.

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` 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.

B.2 Mining tables from Wikipedia

Wikipedia is a rich, free source of information readily accessible on the web with a lot of this information stored in a structured format in form of 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.

C Additional Analysis on Teacher-Student Performance

Figure 3(a) shows the histogram of the Likert ratings (scale 1-4) provided by the students in the explanation verification stage. 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. Figure 3(c) shows average teacher and average student performance on tasks in `CLUES-Real`. Positive correlation (ρ = 0.48) in this figure indicates that task difficulty (captured by classification accuracy) is well-correlated for a teacher and student on average.

For the crowdsourced datasets, we show the number of explanations collected per task in Figure 9(a). The number of explanations is largely around an average value of 11 explanations per task.

Figure 9(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 9(c).

We also illustrate the differences between teacher and student on our tasks in §3. Here we present two additional plots showing the performance of (1) best teacher vs their students for each task (Figure 7) and (2) worst teacher vs their students for each task (Figure 8). 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.

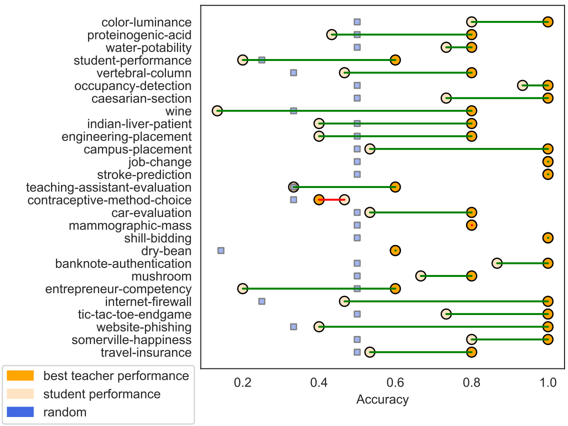


Figure 7: Best teacher vs average of their students for tasks in CLUES-Real. Red lines indicate cases where the student performance is more than the teacher performance. Green lines indicate cases where teachers perform better than students.

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 3, we summarize the payment structure provided to the turkers on the AMT platform for each of the stages (described in detail in §2) – (1) Wikipedia mining on tables scraped from Wikipedia, (2) Explanation collection for tables obtained from UCI, Kaggle and Wikipedia, and (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).

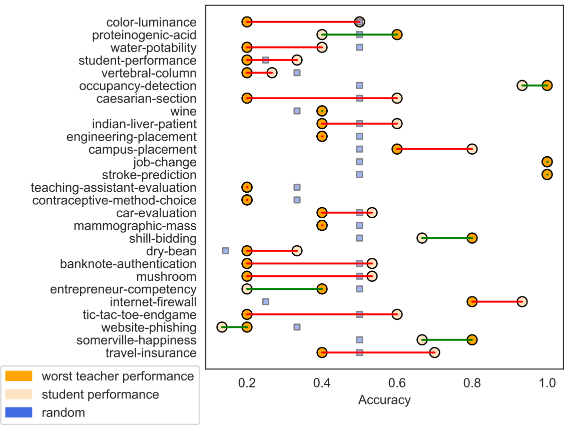


Figure 8: Worst teacher vs average of their students for tasks in CLUES-Real. Red lines indicate cases where the student performance is more than the teacher performance. Green lines indicate cases where teachers perform better than students.

STAGE	$/HIT	BONUS
Wikipedia Mining	$3	$3-$4 [7]
Explanation Collection	$5.5	-
Explanation Validation	$1.2	-

Table 3: Payment structure for AMT Tasks

E Training details

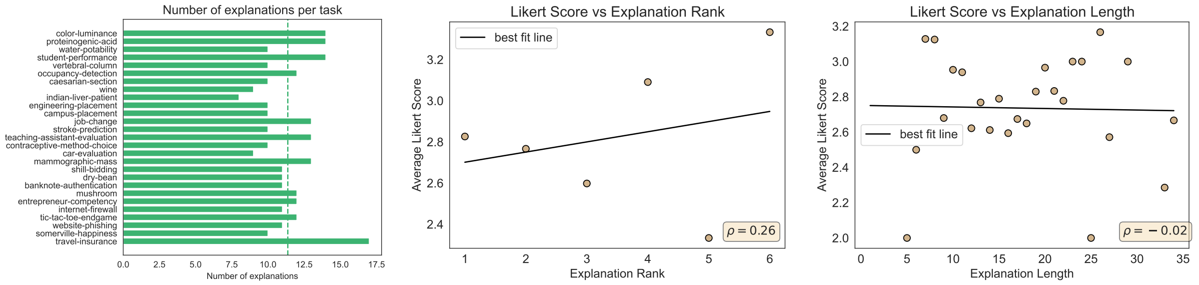
In this section we proved details about implementation of various models, hyperparameter details, and details about hardware and software used along with an estimate of time taken to train the models. Code and dataset for our paper will be made public upon first publication.

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`

[7] ¢50 per table submitted



(a) (b) (c)

Figure 9: (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)

- `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 generalization 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 the entire set of 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`: We consider a pre-trained RoBERTa model fine-tuned on MNLI (Williams et al., 2017) corpus as our base entailment model.[8] The number of parameters in `ExEnt` is same as

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

the pre-trained RoBERTa mdoel finetuned on MNLI corpus. 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 meta-information, while for the explanations in CLUES-Real, the authors manually annotate this meta-information.

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 pretrained 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 a sequential order) in the seen split.

E.4 ExEnt Implementation Details

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 meta-information, while for the explanations in CLUES-Real, the authors manually annotate this meta-information.

E.5 Hardware and software specifications

All the models are coded using Pytorch 1.4.0[10] (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.6 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 Effect of scrambling attribute names in input to ExEnt

We performed an additional experiment on our synthetic data to evaluate if ExEnt understands (1) the relationship between attribute names and attribute values and (2) identify the correspondence between attribute names in the explanations with the attribute name-value pair in the FaT representation of

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

[10] https://pytorch.org/

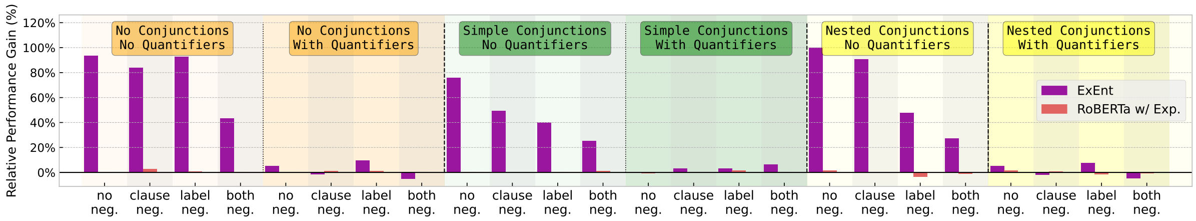


Figure 10: Ablation analysis on the effect of structural and linguistic variations of explanations on generalization ability of models. All bars indicate the relative performance gain over the RoBERTa w/o Exp. baseline.

the structured input. In this experiment, we scrambled (randomly permuted the column names) the structured input when performing inference over the tasks in CLUES-Synthetic. So after the scrambling operation, the attribute name-value pairs will be incoherent. We considered 5 random seeds (42 to 46) for this scrambling operation. The mean generalization performance (accuracy) using scrambled inputs for CLUES-Synthetic is 41.62% (with standard deviation as 0.9%). This is comparable with the random baseline on CLUES-Synthetic (42.19%) as expectadly, ExEnt fails to identify meaningful correspondences between the explanation and input when the inputs are incoherent.

G Key Challenges

We identify the open challenges in learning from explanations by 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 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 10 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 exciting future research.

H Annotation interfaces

We present the different annotation templates and interfaces used for our explanation collection and verification stages in Figures 11,12,13,14 and Figure 15 respectively.

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

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 4: 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 5: 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 6: 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"]
        }
}
```

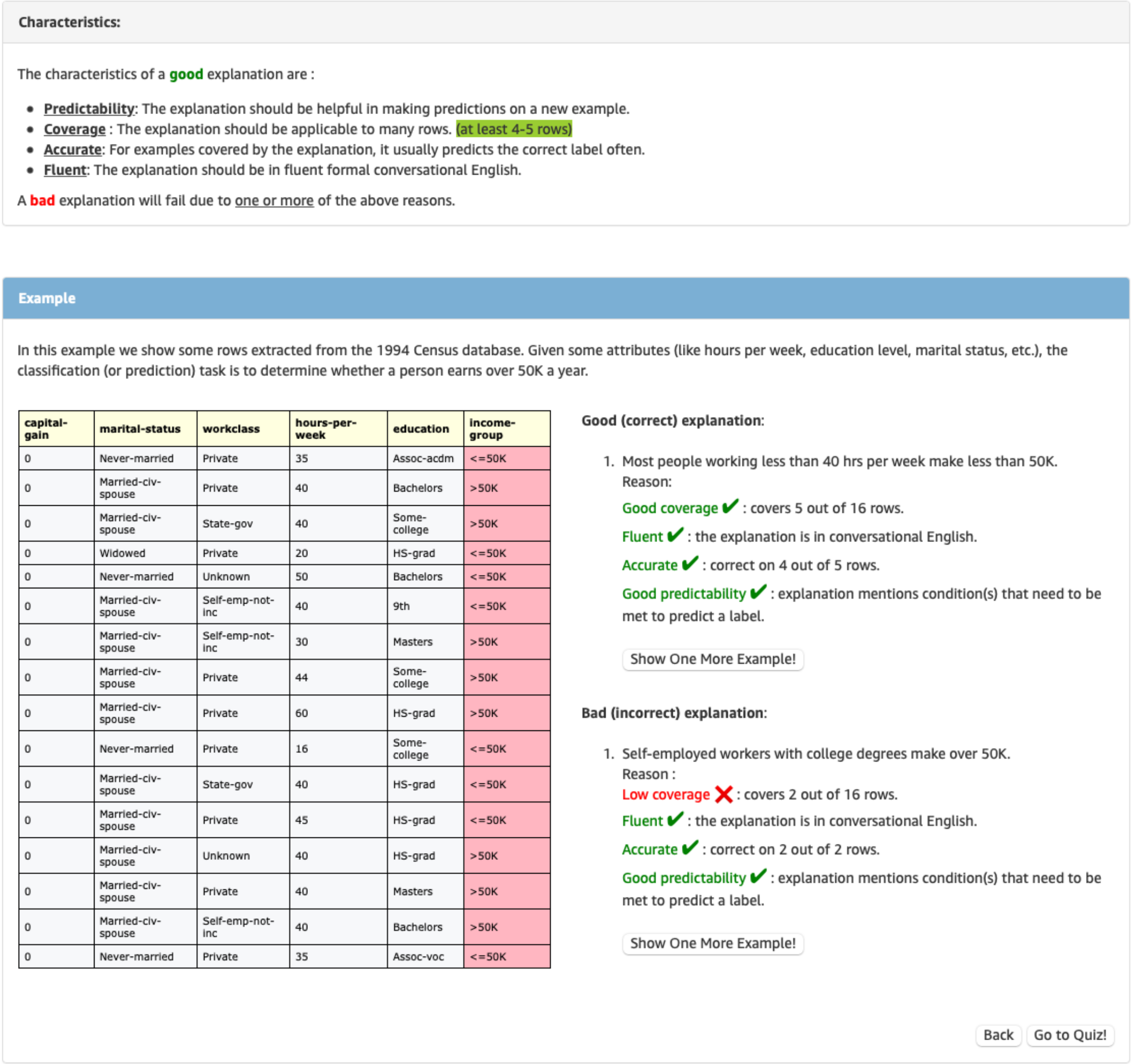
Table 7: Synthetic table schema 3: Rainfall Prediction

```
{
    "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 8: 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 9: Synthetic table schema 5: Bond Relevance Classification



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.

Example

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

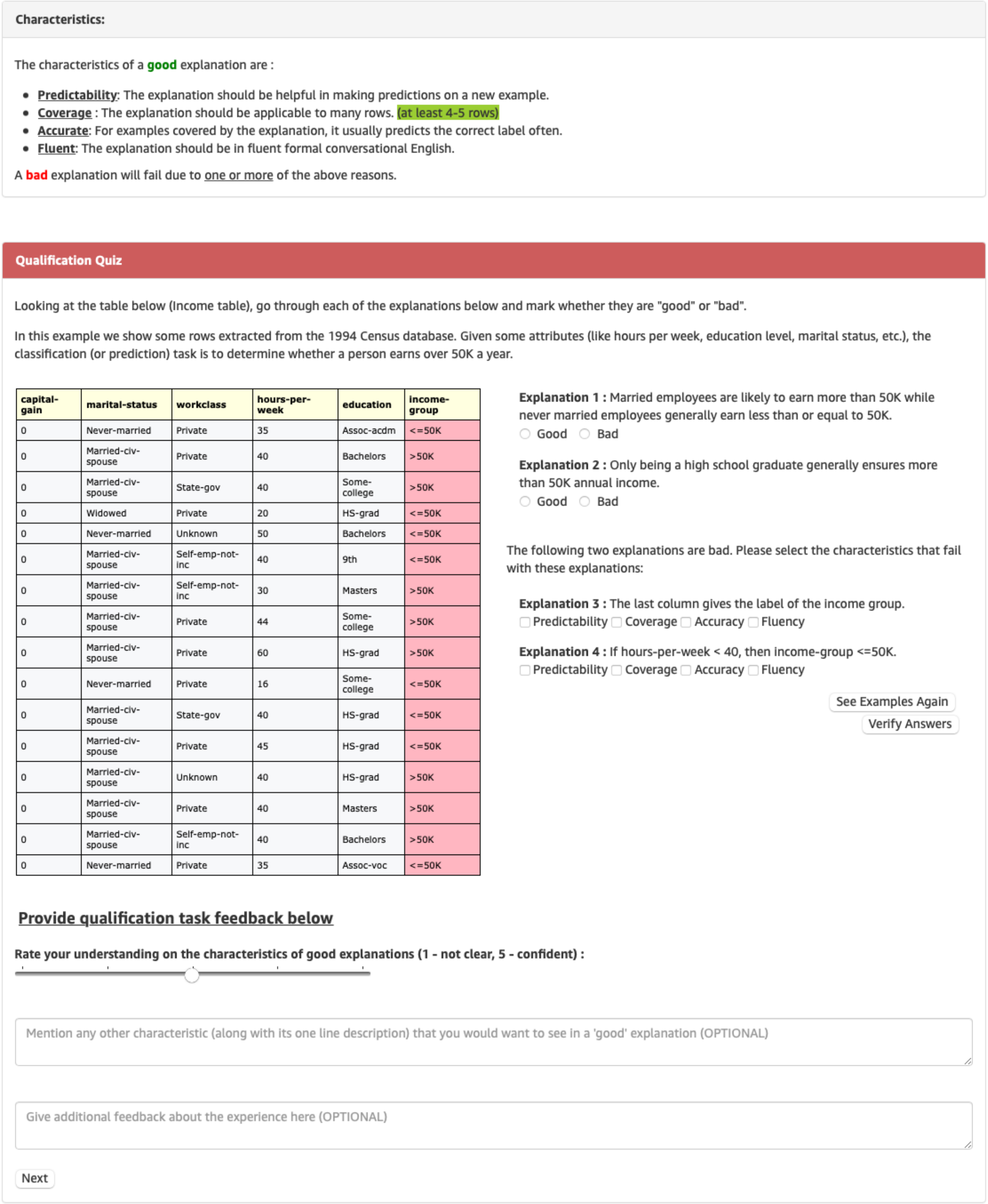
Good (correct) explanation:

1. Most people working less than 40 hrs per week make less than 50K.
   Reason:
   Good coverage ✔ : covers 5 out of 16 rows.
   Fluent ✔ : the explanation is in conversational English.
   Accurate ✔ : correct on 4 out of 5 rows.
   Good predictability ✔ : explanation mentions condition(s) that need to be met to predict a label.

Bad (incorrect) explanation:

1. Self-employed workers with college degrees make over 50K.
   Reason :
   Low coverage ✗ : covers 2 out of 16 rows.
   Fluent ✔ : the explanation is in conversational English.
   Accurate ✔ : correct on 2 out of 2 rows.
   Good predictability ✔ : explanation mentions condition(s) that need to be met to predict a label.

Figure 11: 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

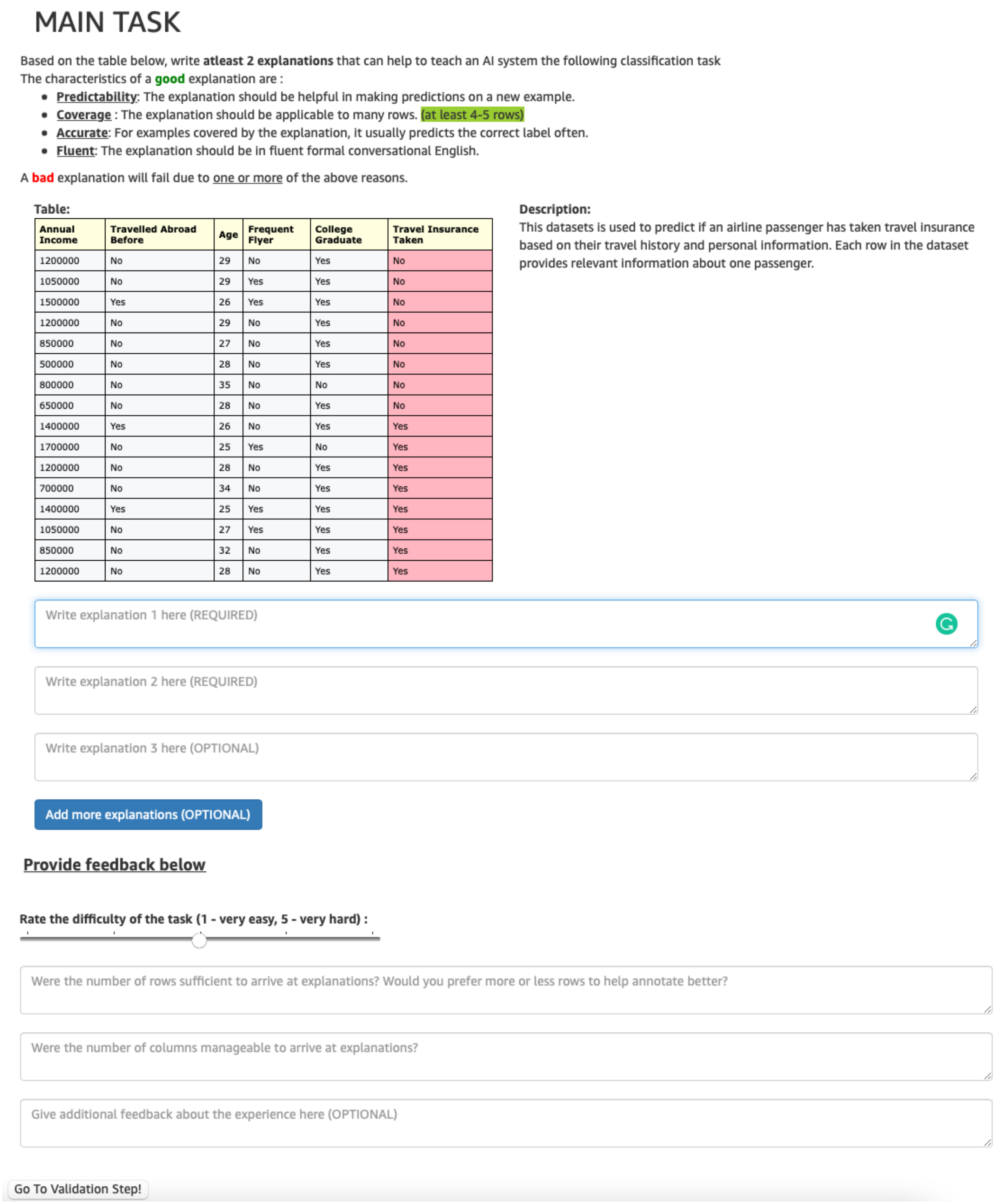
Provide qualification task feedback below

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)

Figure 12: Explanation Collection: Qualification Task page.



MAIN TASK

Based on the table below, write **atleast 2 explanations** that can help to teach an AI system the following classification task
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.

Table:

Annual Income	Travelled Abroad Before	Age	Frequent Flyer	College Graduate	Travel Insurance Taken
1200000	No	29	No	Yes	No
1050000	No	29	Yes	Yes	No
1500000	Yes	26	Yes	Yes	No
1200000	No	29	No	Yes	No
850000	No	27	No	Yes	No
500000	No	28	No	Yes	No
800000	No	35	No	No	No
650000	No	28	No	Yes	No
1400000	Yes	26	No	Yes	Yes
1700000	No	25	Yes	No	Yes
1200000	No	28	No	Yes	Yes
700000	No	34	No	Yes	Yes
1400000	Yes	25	Yes	Yes	Yes
1050000	No	27	Yes	Yes	Yes
850000	No	32	No	Yes	Yes
1200000	No	28	No	Yes	Yes

Description:
This datasets is used to predict if an airline passenger has taken travel insurance based on their travel history and personal information. Each row in the dataset provides relevant information about one passenger.

Write explanation 1 here (REQUIRED)

Write explanation 2 here (REQUIRED)

Write explanation 3 here (OPTIONAL)

Add more explanations (OPTIONAL)

Provide feedback below

Rate the difficulty of the task (1 - very easy, 5 - very hard) :

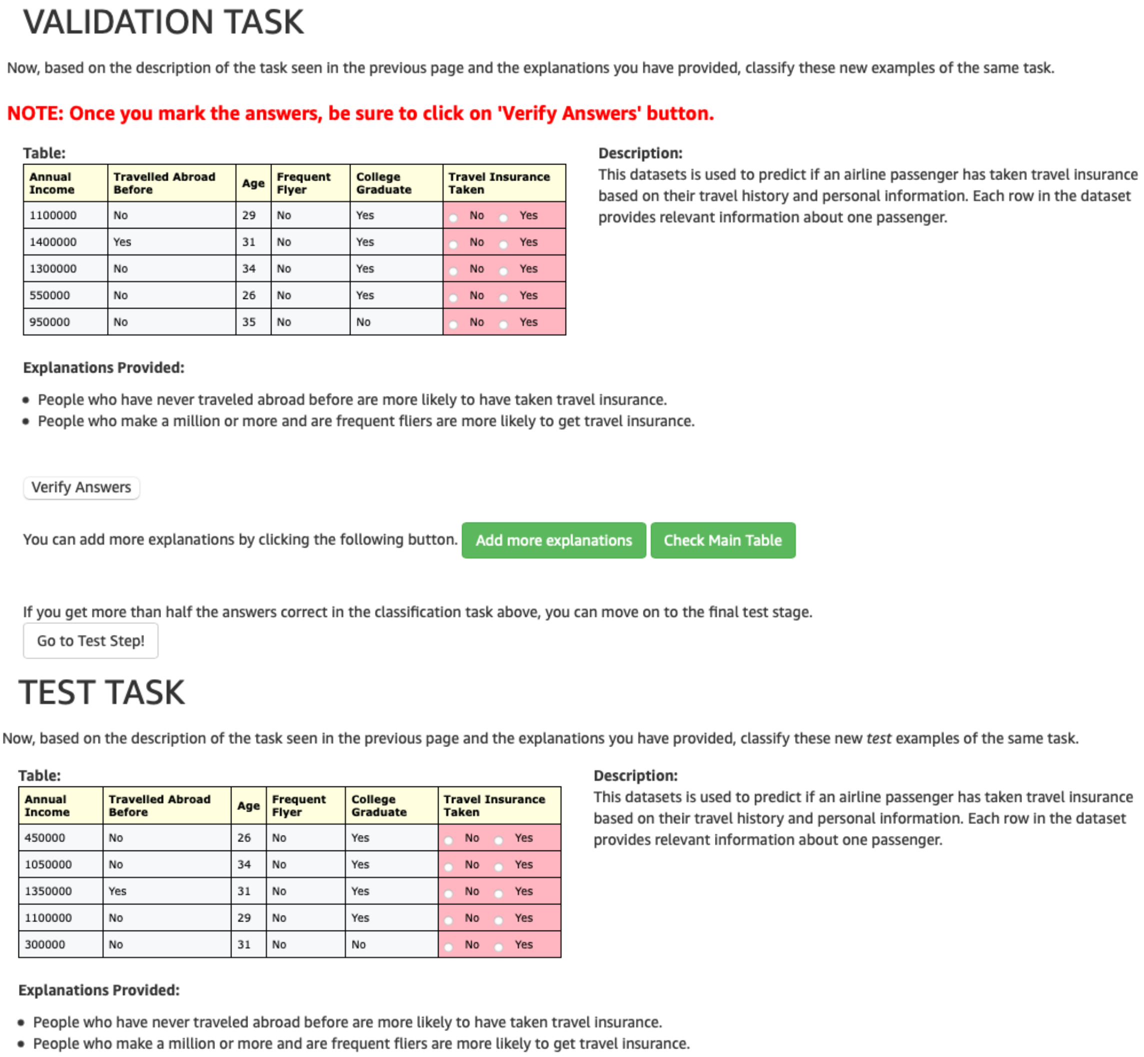
Were the number of rows sufficient to arrive at explanations? Would you prefer more or less rows to help annotate better?

Were the number of columns manageable to arrive at explanations?

Give additional feedback about the experience here (OPTIONAL)

Go To Validation Step!

Figure 13: Explanation Collection: Main Task page.



VALIDATION TASK

Now, based on the description of the task seen in the previous page and the explanations you have provided, classify these new examples of the same task.

NOTE: Once you mark the answers, be sure to click on 'Verify Answers' button.

Table:

Annual Income	Travelled Abroad Before	Age	Frequent Flyer	College Graduate	Travel Insurance Taken
1100000	No	29	No	Yes	○ No ○ Yes
1400000	Yes	31	No	Yes	○ No ○ Yes
1300000	No	34	No	Yes	○ No ○ Yes
550000	No	26	No	Yes	○ No ○ Yes
950000	No	35	No	No	○ No ○ Yes

Description:
This datasets is used to predict if an airline passenger has taken travel insurance based on their travel history and personal information. Each row in the dataset provides relevant information about one passenger.

Explanations Provided:

- People who have never traveled abroad before are more likely to have taken travel insurance.
- People who make a million or more and are frequent fliers are more likely to get travel insurance.

You can add more explanations by clicking the following button.

If you get more than half the answers correct in the classification task above, you can move on to the final test stage.

TEST TASK

Now, based on the description of the task seen in the previous page and the explanations you have provided, classify these new *test* examples of the same task.

Table:

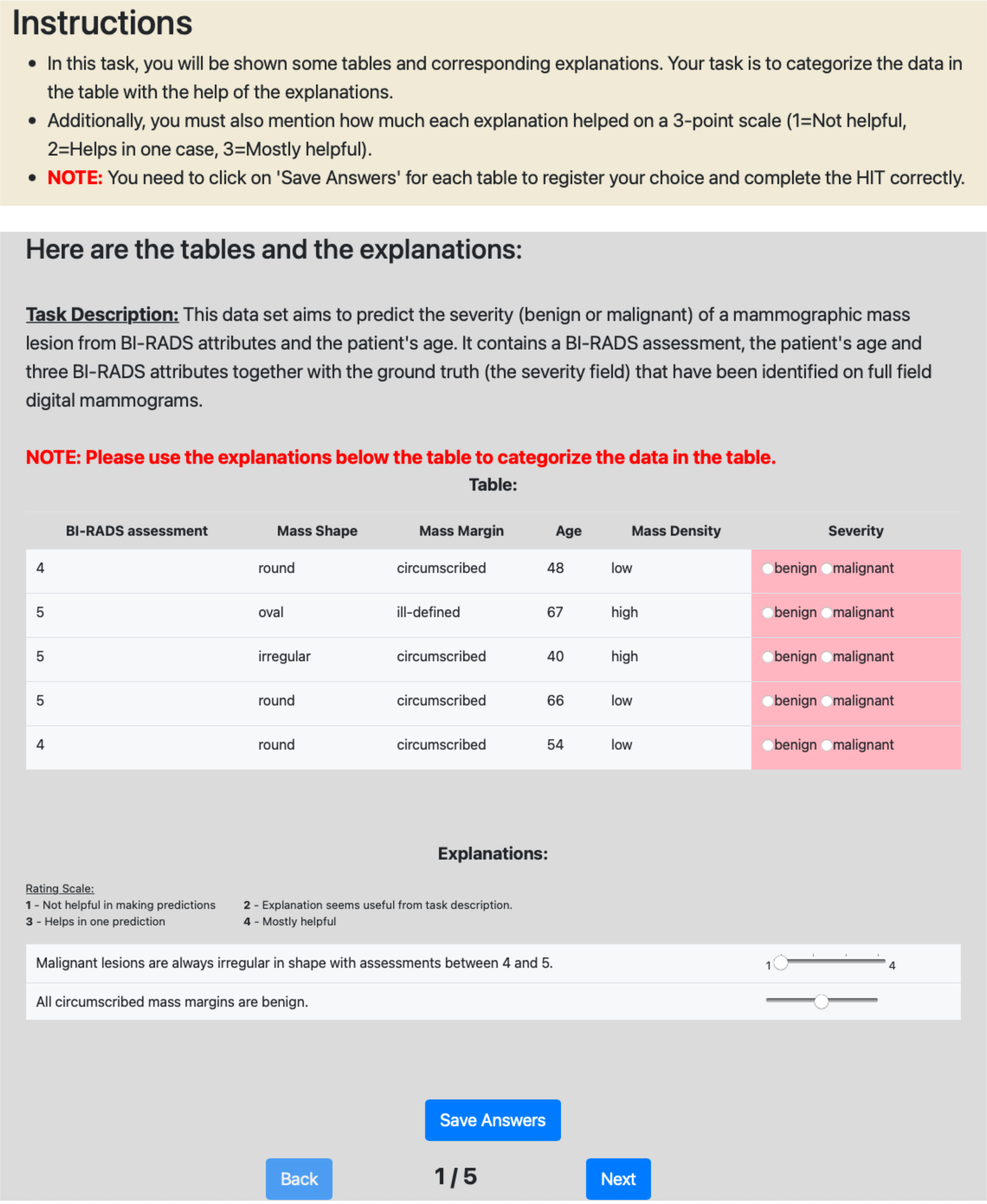
Annual Income	Travelled Abroad Before	Age	Frequent Flyer	College Graduate	Travel Insurance Taken
450000	No	26	No	Yes	○ No ○ Yes
1050000	No	34	No	Yes	○ No ○ Yes
1350000	Yes	31	No	Yes	○ No ○ Yes
1100000	No	29	No	Yes	○ No ○ Yes
300000	No	31	No	No	○ No ○ Yes

Description:
This datasets is used to predict if an airline passenger has taken travel insurance based on their travel history and personal information. Each row in the dataset provides relevant information about one passenger.

Explanations Provided:

- People who have never traveled abroad before are more likely to have taken travel insurance.
- People who make a million or more and are frequent fliers are more likely to get travel insurance.

Figure 14: Explanation Collection: Validation and Test page.



Instructions

- In this task, you will be shown some tables and corresponding explanations. Your task is to categorize the data in the table with the help of the explanations.
- Additionally, you must also mention how much each explanation helped on a 3-point scale (1=Not helpful, 2=Helps in one case, 3=Mostly helpful).
- **NOTE:** You need to click on 'Save Answers' for each table to register your choice and complete the HIT correctly.

Here are the tables and the explanations:

Task Description: This data set aims to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains a BI-RADS assessment, the patient's age and three BI-RADS attributes together with the ground truth (the severity field) that have been identified on full field digital mammograms.

NOTE: Please use the explanations below the table to categorize the data in the table.

Table:

BI-RADS assessment	Mass Shape	Mass Margin	Age	Mass Density	Severity
4	round	circumscribed	48	low	○benign ○malignant
5	oval	ill-defined	67	high	○benign ○malignant
5	irregular	circumscribed	40	high	○benign ○malignant
5	round	circumscribed	66	low	○benign ○malignant
4	round	circumscribed	54	low	○benign ○malignant

Explanations:

Rating Scale:
1 - Not helpful in making predictions
2 - Explanation seems useful from task description.
3 - Helps in one prediction
4 - Mostly helpful

Malignant lesions are always irregular in shape with assessments between 4 and 5.	1 ○ 4
All circumscribed mass margins are benign.	○

Figure 15: Explanation Verification page.