QuExEnt: Improved Zero-Shot Classification from Explanations Through Quantifier Modeling and Curriculum Learning

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

Abstract

A hallmark of human intelligence is the ability to learn new concepts purely from language. While recent advances in training machine learning models via natural language explanations show promise, these approaches still fall short on modeling the the intricacies of natural language (such as quantifiers) or in mimicking human behavior in learning a suite a tasks with varying difficulty. In this work, we present QuExEnt, to learn better zero-shot classifiers from explanations by using three strategies - (1) model the semantics of quantifiers present in explanations (including exploiting ordinal strength relationships, such as 'always' > 'likely'), (2) aggregating information from multiple explanations using an attention-based mechanism, and (3) model training via curriculum learning from tasks with simple explanations to tasks with complex explanations. With these strategies, QuExEnt outperforms prior work showing an absolute gain of up to 7% on the recently proposed CLUES benchmark in generalizing to unseen classification tasks.

1 Introduction

Learning from language is a new paradigm of machine learning whereby machines are taught to perform tasks through language instructions/explanations (Arabshahi et al., 2020). Remarkably, this form of language supervision to train classification models has been shown to be effective in few-shot (Srivastava et al., 2017; Hancock et al., 2018), and zero-shot settings (Srivastava et al., 2018). More recently, Menon et al. (2022) introduce a benchmark, CLUES and a model, ExEnt to learn generalizable classifiers guided by explanations.

While recent approaches like ExEnt show promise in learning from explanations, they still fall short on modeling the intricacies of natural language or in mimicking human behaviour in learning a suite of tasks with varying difficulty.

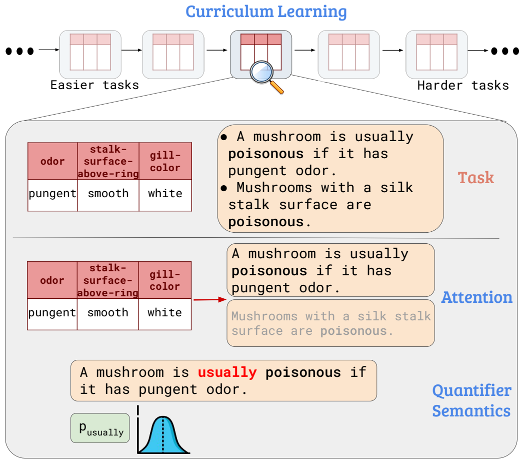

Curriculum Learning
Easier tasks
Harder tasks
odor
stalk-surface-above-ring
gill-color
pungent
smooth
white
• A mushroom is usually **poisonous** if it has pungent odor.
• Mushrooms with a silk stalk surface are **poisonous**.
Task
odor
stalk-surface-above-ring
gill-color
pungent
smooth
white
A mushroom is usually **poisonous** if it has pungent odor.
Mushrooms with a silk stalk surface are poisonous.
Attention
A mushroom is **usually** **poisonous** if it has pungent odor.
$p_{usually}$
Quantifier Semantics

Figure 1: We present improved techniques to learn classifiers from natural language explanations. Our proposed techniques makes use of Curriculum Learning to progressively learn easy to hard tasks, Attention to identify the most salient explanations for classifying an example, and modeling Quantifier Semantics to account for explanation confidences.

First, ExEnt fails to capture semantics of quantifiers present in the explanations. Quantifiers are an ubiquitous part of natural language and can dictate the vagueness and perceived confidence of relations expressed in a statement (Solt, 2009; Moxey and Sanford, 1986). Second, prior work does not completely mimic human learning, where humans learn 'simpler' concepts first and then gradually build towards 'harder' concepts (Newport, 1990). While curriculum learning (Bengio et al., 2009) has been shown to be a fruitful in numerous machine learning tasks (Platanios et al., 2019; Tay et al., 2019; Narvekar et al., 2017), it is yet to be explored in the context of learning from explanations. The deteriorating generalization performance of classifiers with increasing complexity of explanations in prior work (Menon et al., 2022) further motivates the need of curriculum learning in the context of training classifiers from explanations.

To address the first shortcoming, we present

[*] Equal contribution

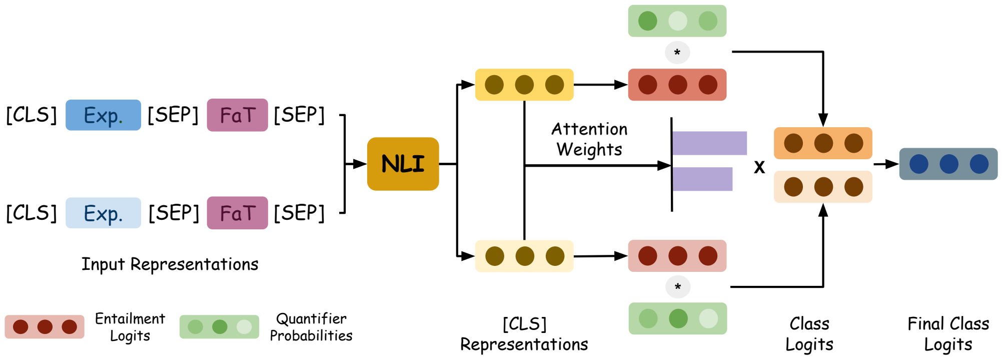


Figure 2: QuExEnt models quantifier semantics and uses attention over multiple explanations to aggregate class logits. As shown in the figure, our approach allows us to re-weight the logits from the NLI step, thus strengthening/weakening the contribution of an explanation towards assigning the label (mentioned in the explanation) to the input. $\circledast$ denotes the operations described in §3.1 for assignment of class logits using the outputs from the NLI step. Curriculum learning (not shown in the figure) entails training QuExEnt progressively on easy-to-hard tasks.

QuExEnt that extends ExEnt by modeling quantifier semantics explicitly. We explore learning quantifier semantics directly from labeled classification data and use weak supervision in the form of ordinal relations describing the relative strengths of quantifiers (e.g. 'always' > 'often'). Second, QuExEnt also uses an attention mechanism for aggregating the effect of different explanations corresponding to a task. Finally, we consider different axes of explanation complexities and empirically evaluate the utility of curriculum learning on three different curricula. Through experiments on the CLUES benchmark we demonstrate the effectiveness of our introduced strategies. Our model, QuExEnt outperforms prior work on generalising better to novel classification tasks.

2 Preliminaries

2.1 Setup

We employ a cross-task generalization setup (Mishra et al., 2022; Sanh et al., 2022, *inter alia*), and train models using multi-task training over a set of tasks $\mathcal{T}_{seen}$ and evaluate for zero-shot generalization on a new task, $t \in \mathcal{T}_{novel}$ ($\mathcal{T}_{novel} \cap \mathcal{T}_{seen} = \phi$). For experiments, we utilize the recently proposed CLUES benchmark (Menon et al., 2022) . In CLUES, the inputs are structured in nature, i.e., they are collection of attribute name-attribute value pairs. We encode each structured data example, x, as a text sequence of attribute-name and attribute-value pairs separated by [SEP] tokens (referred to as 'Features-as-Text' or 'FaT' in Menon et al. (2022)). Additional details on CLUES can be found in App. A.

2.2 Dataset

We use the synthetic and real-world classification datasets from CLUES Benchmark (Menon et al., 2022). On analysis, we observed that preconditions of the explanations were satisfied for only 30% of the samples on average in CLUES-Synthetic. In other words, using explanations we can only classify 30% of the samples and thus only this fraction of samples would be effective for learning quantifier semantics. Our initial experiments revealed a need for more samples where explanations would be applicable. Thus, we re-created the synthetic tasks, now with around 50-60% of the samples where explanations are applicable.

2.3 ExEnt

In the cross-task generalization setting, Menon et al. (2022) identified that a simple concatenation of explanations with the input was insufficient to endow pre-trained language models with the ability to generalize to novel tasks. Hence, they introduce ExEnt, a model which uses NLI[1] as an intermediate step. The operations in ExEnt can be broadly grouped into three steps: **(a) NLI step**: obtain scores from an entailment prediction model (RoBERTa+MNLI-finetuned) for the alignment between the input and each explanation available for a task; **(b) Entailment → Classification scores conversion**: convert the entailment scores for each input-explanation pair into classification scores based on the nature of the explanation; and **(c) Aggregation**: aggregate classification scores from each input-explanation pair using mean to obtain an overall score for classification. The aggregated

[1] Natural Language Inference

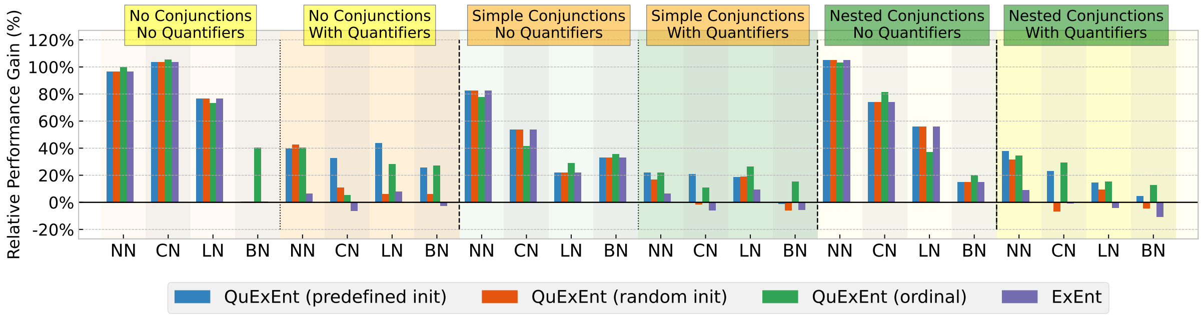


Figure 3: Performance of `QuExEnt` as compared to the baseline models. We see that `QuExEnt` outperforms `ExEnt` (Menon et al., 2022) on the synthetic datasets that contain explanations with quantifiers.

scores are then converted to probabilities using softmax, and the model is trained using cross-entropy loss. For further details, we refer the reader to Menon et al. (2022).

3 `QuExEnt`

3.1 Modeling Quantifier Semantics

Prior work in cognitive science (Chopra et al., 2019; Steinert-Threlkeld, 2021) and machine learning (Srivastava et al., 2018; Menon et al., 2022) show that people use quantifiers often in learning or teaching tasks to express varying strength of relations in a statement. To the best of our knowledge, prior work has not explored learning quantifier semantics in a data-driven way. Here, we devise methods to explicitly model the differential semantics of quantifiers present in explanations to guide classifier training.

To formalize our approach to modeling quantifier semantics, consider a task t with the set of class labels L and set of explanations E. Given the Feature-as-Text (FaT) representation of a structured data example $x \in t$ and an explanation $e_j \in E$, our model takes FaT(x) and e_j as input and passes it through a pretrained RoBERTa+MNLI model, similar to previous work (Menon et al., 2022). For each example-explanation pair, the NLI model outputs entailment, neutral, and contradiction scores (denoted as s_e^j, s_n^j, and s_c^j respectively). In the next step, we incorporate quantifier semantics to assign logits to the set of class labels, L, using the outputs of the NLI model. In this work, we model the semantics of a quantifier by a probability value signifying the strength of the quantifier, i.e., the confidence of the quantifier in conveying the beliefs expressed in the explanation. The assignment of class logits is done as follows. If:

- **Explanation e_j mentions a label l_{exp}**: An illustrative example is 'If head equal to 1, then dax', where 'dax' is the label mentioned (l_{exp}). Let p_{quant} denote the probability of the quantifier mentioned in the explanation[2] and $\mathbb{P}(l)$ denote the probability of any label $l \in L$. Then,

$$\log(\mathbb{P}(l_{exp})) \propto p_{quant} \times s_e^j + (1 - p_{quant}) \times s_c^j + s_n^j/|L| \tag{1}$$

$$\forall\, l \in L \setminus \{l_{exp}\},$$

$$\log(\mathbb{P}(l)) \propto p_{quant} \times s_c^j + (1 - p_{quant}) \times s_e^j + s_n^j/|L| \tag{2}$$

Note: If quantifiers are absent in the explanations, we assume p_{quant} is 1.

- **Explanation e_j mentions negation of a label 'l_{exp}' (NOT l_{exp})**: An illustrative example is 'If head equal to 1, then not dax", where 'dax' is the label mentioned (l_{exp}). The roles of s_c^j and s_e^j as described in the previous equations are reversed.

Following this step, we average the class logits from each example-explanation pair to aggregate the decisions. Finally, we apply a softmax over the resulting class scores to obtain a distribution over class labels and train the model to minimize the cross-entropy loss, $\mathcal{L}_{CE}$.

Approaches to learn quantifier semantics We experiment with the following approaches to learn the quantifier probabilities:

- **Finetuning pre-defined probability values**: We initialize the quantifier probability values (p_{quant}) then fine-tuning them while training `QuExEnt`. These initial probabilities can be specified from domain knowledge or by an expert. In this work, we adopt the pre-defined quantifier values from LNQ (Srivastava et al., 2018).
- **Learning probability values for the quantifiers from scratch**: We start from random initialization and then learn the probability values of each

[2] We assume that explanations contain a single quantifier. This assumption also holds true with the explanations found in `CLUES`.

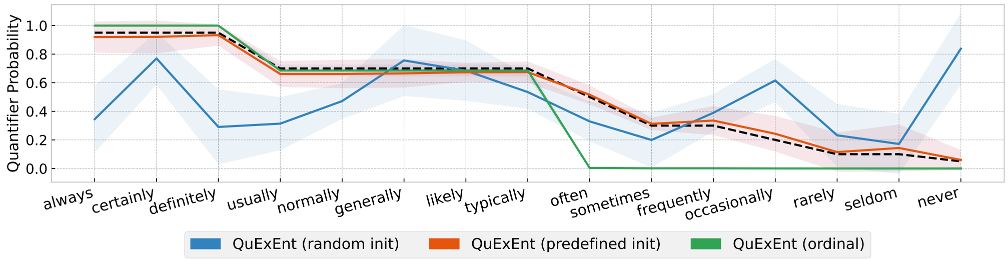


Figure 4: Quantifier probability values learned by the different approaches mentioned in §3.1. The solid line and the shaded region denote the mean and standard deviation respectively of the learned probabilities for a given approach across 48 synthetic task complexities in `CLUES-Synthetic`. The dotted-line denotes (1) probability values used by Menon et al. (2022) to create synthetic tasks of `CLUES-Synthetic` and (2) the quantifier probability initialization values for `QuExEnt` (predefined init).

quantifier while training the classifier. Technically, we learn a real-valued number corresponding to each quantifier and then map it to range [0,1] to model probability associated with the corresponding quantifier.

- **Weak supervision in form of ordinal ranking**: We explore a weaker form of supervision by specifying ordinal ranks between quantifiers based on their relative strengths (‘always’ > ‘likely’). We start from random initialization of the probability values and leverage ordinal relations in the form of a ranking loss. Following Pavlakos et al. (2018) we define our ranking loss for a pair of quantifiers q_i and q_j $(i \neq j)$ as :

$$\mathcal{L}_{i,j} = \begin{cases} \log(1 + \exp(p_{q_i} - p_{q_j})), & \mathbf{p^*_{q_i}} > \mathbf{p^*_{q_j}} \\ (p_{q_i} - p_{q_j})^2, & \mathbf{p^*_{q_i}} = \mathbf{p^*_{q_j}} \end{cases}$$

where, $\mathbf{p^*_q}$ refers to the subjective probability value of a quantifier, q. Further, we define

$$\mathcal{L}_{rank} = \sum_{(q_i,q_j) \in Q} \mathcal{L}_{i,j} \tag{3}$$

where, Q denotes the full set of quantifiers present in the explanations of `CLUES` (§A.1). The final loss is a weighted sum of classification loss ($\mathcal{L}_{CE}$) and ranking loss ($\mathcal{L}_{rank}$).

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{rank} \tag{4}$$

where, λ denotes the weight of ranking loss.

Performance of `QuExEnt` on synthetic tasks We experiment on the expanded set of synthetic tasks in `CLUES-Synthetic` (i.e., 80 train and 20 test datasets per synthetic complexity) to judge the effectiveness of modeling quantifier semantics explicitly. Figure 3 shows the results of different variants of `QuExEnt` and baselines across the different synthetic task types. We find that explicit modeling of quantifier semantics is generally helpful and outperforms prior work in most task types containing quantifiers. Note that `QuExEnt` and `ExEnt` perform same on task types not containing quantifiers as p_{quant} is 1 in such cases making `QuExEnt` functionally same as `ExEnt`. The generalization ability of the models decrease with the increasing complexity of explanations due to changes in structure of explanations or presence of negations.We leave modeling of negations to future work.

3.2 Improving aggregation across explanations using attention

By design, using mean to aggregate class logits resulting from different explanations corresponding to an input considers all explanations to be equally important for classifying the input. In order to model the varying importance of each explanation towards deciding the class label, we use an attention mechanism for the aggregation step (see §2.3).

We obtain the attention weights by using a feed-forward network over the `[CLS]` representations obtained from the intermediate NLI model. The attention weights are then normalized using softmax. The final aggregated class logits for the label l is $\sum_{j=1}^{m} a_j z_j$, where a_j is the attention weight for each explanation e_j, and z_j denotes the classification logits resulting from e_j.

We train two variants of `QuExEnt` (scratch), one using mean and the other using attention to aggregate over explanations. We find that `QuExEnt` with attention has better generalization (relative improvement of 10%) than `QuExEnt` using mean for aggregation over explanations. We refer the reader to Figure 7(a) in the Appendix.

3.3 Curriculum learning

We define ‘complexity’ of an explanation under three axes - (1) type of classification task (binary vs multiclass), (2) presence of negations, and (3)

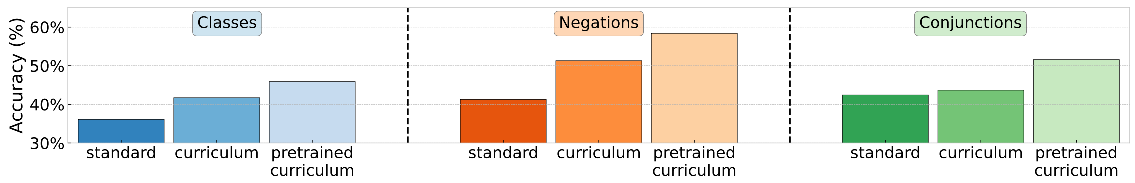


Figure 5: Averaged generalization accuracy on novel classification tasks of the ‘most difficult’ task complexities across three curricula: classes, negations, conjunctions. While effective in all three curricula, curriculum learning shows maximum gains when handling negations.

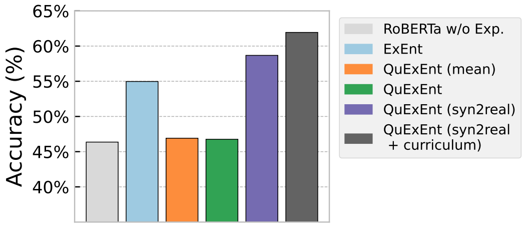


Figure 6: Classification accuracy on novel real-world classification tasks in `CLUES-Real`.

structure of the explanation (presence of conjunction/disjunctions or nested clauses). Drawing motivation from success of curriculum learning (Bengio et al., 2009) in training on an array of increasingly complex tasks, we explore its utility when learning a classifier from explanations. We empirically evaluate if pre-training on a classification task with ‘easier’ explanations gives any advantage when learning a task with ‘hard’ explanations on the following curriculums:

- Binary to multiclass : We first train classifiers on binary classification tasks and then finetune them on multiclass classification tasks.
- No-negations to negations : We begin with tasks that have no negations in their explanations. Gradually, we add tasks with explanations containing negations.
- Simple explanations to explanations containing conjunctions/disjunctions or nested clauses: We train first on tasks without any conjunctions/disjunctions in their explanations. Following this, we train on tasks having explanations that contain one conjunction/disjunction and then on tasks with explanations that contain nested clauses.

Figure 5 shows the results of curriculum learning on the synthetic tasks of `CLUES`. We find that curriculum learning is effective in all the three curricula when we pre-train the quantifier semantics on the simplest binary task with quantifiers and keep the semantics fixed for the remaining curriculum. Notably, we find curriculum learning to be most effective in handling negations.

4 Performance on real-world tasks

In the previous sections, we established the effectiveness of our proposed strategies on a large number of synthetic tasks from `CLUES`. Here, we empirically evaluate `QuExEnt` on the 36 real-world classification tasks from `CLUES` using the aforementioned strategies. Figure 6 shows the generalization performance of `QuExEnt` and the baselines on `CLUES-Real`. We find that directly trying to train `QuExEnt` fails to surpass the baselines (even with attention to aggregate over explanations) as the comparatively low number of explanations in `CLUES-Real` hinders the model from learning quantifier semantics and classification from explanations jointly. To alleviate this issue, we pretrain on `CLUES-Synthetic` and then fine-tune the learned model on `CLUES-Real`. We find that pretraining on synthetic tasks (`QuExEnt` (syn2real)) gives a relative gain of 6.7% in generalization accuracy over `ExEnt`. Next, we evaluate the utility of curriculum learning on real tasks. We start with a pre-trained `QuExEnt` on synthetic tasks and then fine-tune it first on binary tasks of `CLUES-Real` followed by the multiclass tasks of `CLUES-Real`. We find that curriculum learning (`QuExEnt` (syn2real + curriculum)) results in the best generalization, performing significantly better than `ExEnt` (relative gain of 12.7%; $p < 0.005$, paired t-test) on `CLUES-Real`.

5 Conclusion

We present three effective and generalizable strategies to learn classifiers from language explanations. Our strategies focus on modeling quantifier semantics and mimicking human behaviour through curriculum learning setting. Our model, `QuExEnt` trained under this improved setup outperforms prior work showing better generalizing on tasks of the `CLUES` benchmark. Future work can explore other open challenges such as explicit modeling of negations, conjunctions and disjunctions for learing from explanations.

References

Forough Arabshahi, Kathryn Mazaitis, Toby Jia-Jun Li, Brad A Myers, and Tom Mitchell. 2020. Conversational Learning.

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

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

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.

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

Rakesh R. Menon, Sayan Ghosh, and Shashank Srivastava. 2022. CLUES: Benchmark on Learning Classifiers using Natural Language Explanations. *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (to appear)*.

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)*.

Linda M. Moxey and Anthony J. Sanford. 1986. Quantifiers and Focus. *Journal of Semantics*, 5(3):189–206.

Sanmit Narvekar, Jivko Sinapov, and Peter Stone. 2017. Autonomous task sequencing for customized curriculum design in reinforcement learning. In *IJCAI*, pages 2536–2542.

Elissa L Newport. 1990. Maturational constraints on language learning. *Cognitive science*, 14(1):11–28.

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.

Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. 2018. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7307–7316.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*.

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*.

Stephanie Solt. 2009. The semantics of adjectives of quantity.

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.

Shane Steinert-Threlkeld. 2021. Quantifiers in natural language: Efficient communication and degrees of semantic universals. *Entropy*, 23(10):1335.

Yi Tay, Shuohang Wang, Luu Anh Tuan, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. *arXiv preprint arXiv:1905.10847*.

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.

Appendix

A Details on CLUES

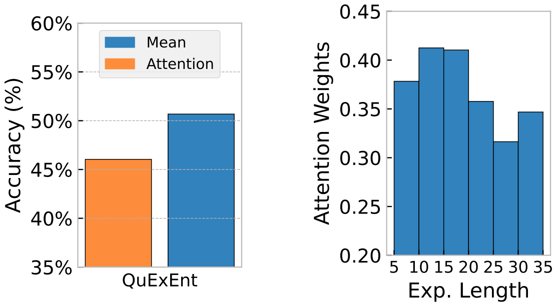
CLUES (Menon et al., 2022) is a recently proposed benchmark of classification tasks paired with natural language explanations. The benchmark consists of 36 real-world classification tasks (CLUES-Real) as well 144 synthetic classification tasks (CLUES-Synthetic). The explanations for the real-world tasks are obtained through crowdsourcing while they are programmatically generated for CLUES-Synthetic. In this work, we follow the train and test splits for CLUES-Real from Menon et al. (2022). Additionally, 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, similar to (Menon et al., 2022), we use the entire set of examples for zero-shot testing.

A.1 List of quantifiers

The full list of quantifiers along with their associated probability values are shown in Table 1.

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 1: Probability values used for quantifiers in CLUES. These values are based on Srivastava et al. (2018).



(a)

(b)

Figure 7: (a) Generalization accuracy on CLUES-Synthetic ablating the use of attention to combine results from multiple explanations in QuExEnt. (b) Mean attention scores of explanations from QuExEnt vs explanation length (# of tokens).

B Utility of attention for aggregating across explanations

In this section we discuss the utility of using attention instead of mean for aggregating across explanations. Further, we discuss the attention weights observed for different types of synthetic explanations.

Performance on CLUES-Synthetic: Figure 7(a) shows the generalization performance on CLUES-Synthetic for two variants of QuExEnt, one using mean and the other using attention for aggregation. We find that using attention for aggregation across explanations results in significantly better generalization accuracy (50.68% vs 46.04% ; $p < 0.1$, paired t-test). While technically simple, we see that this modification allows the model to behave in conceptually sophisticated ways.

Attention weight analysis: Figure 7(b) shows a histogram of average attention weights from QuExEnt for different explanation lengths. We find that longer explanations (typically explanations containing nesting of conjunctions and disjunctions) get lower attention weights on average. This seems reasonable and intuitive, since more complex explanations are likely harder for the model to interpret correctly, and hence relying overly on them may be riskier. Further, we find that explanations containing a quantifier receive higher attention on average than explanations without quantifier (0.44 vs 0.35), further highlighting the value of modeling quantifiers in explanations. Explanations containing 'definitely' and 'frequently' received higher attention than explanations containing other quantifiers. Somewhat surprisingly, we found that the

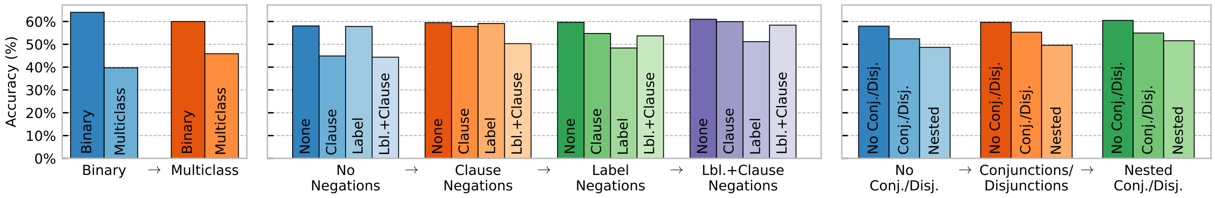


Figure 8: Progression of generalization accuracies on task complexities as we move forward in the curriculum for all three curricula (from left to right: Classes, Negations and Conjunctions curriculum). The text on each bar indicates the evaluation complexity, while the x-axis indicates the complexity that the model has been currently trained on in the curriculum.

average attention weights were comparable for explanations with and without negation.

C Forgetting in Curriculum Learning

In Figure 8, we show the trajectories of generalization performance as we increase the complexity along three independent axes in the three curricula. Briefly, our results indicate that in learning tasks with more classes, generalization increases on multiclass classification tasks at the expense of a slight performance decrease on the more straightforward binary tasks. In the curriculum focused on negations, `QuExEnt` underperforms on tasks with explanations that have 'label negations' after training on the relevant training datasets for that complexity. However, on further analysis, we observe that this trend is more pronounced when 'label negations' are paired with multiclass classification tasks. By contrast, `QuExEnt` improves through training on the relevant training datasets of binary classification tasks with 'label negations' in concepts. Lastly, training progressively on more structurally complex tasks resulting from conjunctions/disjunctions in explanations shows improvements during evaluation across all conjunction types without forgetting how to solve simpler tasks.

D Training details

In this section we proved details about implementation such as hyperparameter details, and details about hardware and software used along with an estimate of time taken to train the models.

D.1 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`.

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 a learning rate of $1e-5$. 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.

For `QuExEnt` (ordinal), to weight the ranking loss we use $\lambda = 10$, chosen using validation performance.

D.2 Hardware and software specifications

All the models are coded using Pytorch 1.4.0[3] (Paszke et al., 2019) and related libraries like numpy (Harris et al., 2020), scipy (Jones et al., 2001–) etc. We run all experiments on a Tesla V100-SXM2 GPU of size 16GB, 250 GB RAM and 40 CPU cores.

D.3 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.
- Training on `CLUES-Synthetic`: All the models take comparatively much lesser time for training on our synthetic tasks owing to lesser num-

[3] https://pytorch.org/

ber 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`.