CLUES Appendix: Few-Shot Learning Evaluation in NLU

Subhabrata Mukherjee, Xiaodong Liu, Guoqing Zheng, Saghar Hosseini, Hao Cheng
Greg Yang, Christopher Meek, Ahmed Hassan Awadallah, Jianfeng Gao
Microsoft Research
{submukhe, xiaodl, zheng, sahoss, chehao}@microsoft.com
{gregyang, meek, hassanam, jfgao}@microsoft.com

A Appendix

A.1 Additional Considerations for Benchmark Construction

Tasks Considered But Not Selected While the Winograd challenge is a task in GLUE, we do not include it here because the dataset is small and noisy. Other tasks in GLUE or SuperGLUE not mentioned above all have large overlap with the included tasks (e.g., all the textual entailment tasks correlate strongly with MNLI performance). We also considered adversarially constructed version of tasks such as ANLI [3]. Ultimately we did not include ANLI because 1) there is already a large gap between human and machine performance on MNLI and 2) like many adversarially constructed datasets, the construction of ANLI is biased against the model using for its construction (RoBERTa), violating Selection Principle 4.

We also considered creating more difficult versions of tasks via adversarially perturbing context/prompt or by selecting hard questions w.r.t. a reference model (e.g. BERT or RoBERTa). However, we ultimately did not adopt these approaches for the following two reasons: 1) We observed that the perturbed examples from such adversarial methods are unnatural and typically not readable by humans. 2) Both adverserial perturbation and selection require a reference model, which violates Design Principle 3.

A.2 Human Evaluation Platform

In this section, we provide the snapshots of our human evaluation platform for some of the tasks in CLUES. Figure 1 shows an example of the sentence classification task (MNLI). The short task description included the fist sentence that annotators see when they open the WebApp and judges can click on the help hyperlink to open the small windows which contain the definition of the corresponding label and one example. In Figure 1, the annotator has clicked on both the Neutral and the Entailment help links.

Figure 2 shows an example of the named entity recognition task (CoNLL03). In this WebApp, the annotator sees a question at the top and a short instruction about how to highlight and submit their answer on the right side of the page.

Figure 3 shows an example of the named entity recognition task (SQuADv2). In this WebApp, the annotator sees a question at the top and a short instruction about how to highlight and submit their answer on the right side of the page.

A.3 Human Performance Analysis in Training Step

Table 1 shows the difference in performance between groups of annotators on the 20-shots examples used for training the judges. We can observe that the annotators who had worked on 30-shot setting

35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks.

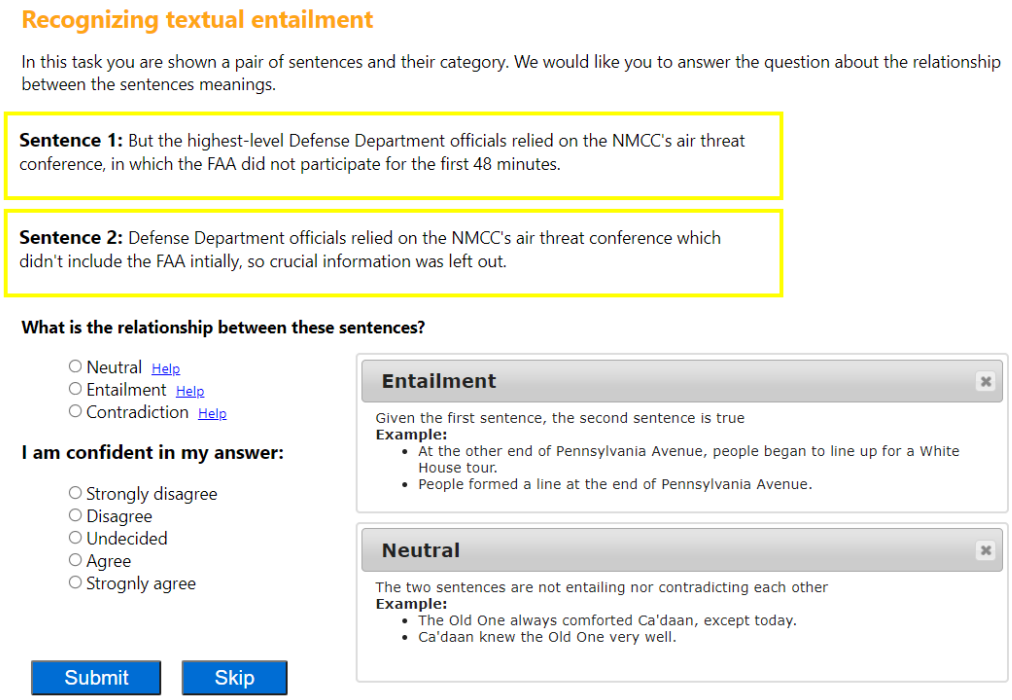

Figure 1: Human evaluation platform for MNLI task.

are out-performing the annotators who worked on 20-shots setting on average on training tasks while under performing on test tasks. One hypothesis is that the extra 10 examples in 30-shot setting have confused the annotators and thus affected their performance on test task. Therefore, we analyse the performance of annotators on those examples in Table 2. From Table 2 and Table 1, we can observe that the performance of the annotators significantly declined on the 10 examples except for WikiANN task. Therefore, these examples can create more confusion and self-doubt which could compromise the performance of the annotators.

Table 1: Human performance on 20 shots train set. We report *S1* score and its variance across 3 annotators for each setting.

	Sentence Classification		**Named Entity Recognition**		**Machine Reading Comprehension**	
Train	SST-2	MNLI	CoNLL03	WikiANN	SQuADv2	ReCoRD
20	83.33 ± 6.2	90.0 ± 4.1	86.1 ± 18	86.1 ± 0.8	86 ± 10.6	95 ± 4.1
30	91.7 ± 4.7	98.3 ± 2.4	84.7 ± 17.3	88.7 ± 0.5	89.9 ± 2.0	100 ± 0.0

Table 2: Human performance on 10 examples of the 30 shots training set that do not appear in the 20-shots training set. We report *S1* score and its variance across 3 annotators for each setting.

Sentence Classification		**Named Entity Recognition**		**Machine Reading Comprehension**	
SST-2	MNLI	CoNLL03	WikiANN	SQuADv2	ReCoRD
83.3 ± 9.4	93.3 ± 4.7	80.4 ± 21.5	92.6 ± 5.3	80.0 ± 8.1	93.3 ± 4.7

A.4 Variance comparison

Figure 4 compares the variance in the few-shot performance of standard fine-tuning between the prompt-based fine-tuning. We observe a wide variance in the few-shot performance of standard

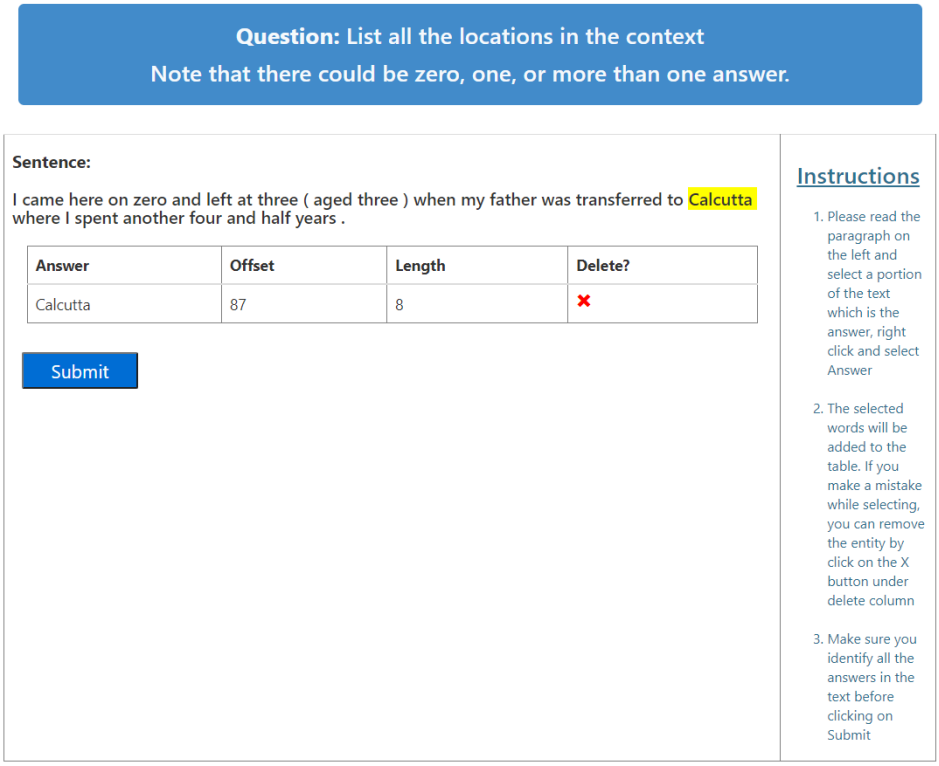

Figure 2: Human evaluation platform for CoNNL03 task.

fine-tuning that is exacerbated by the model size, although the impact is less on prompt-based fine-tuning.

A.5 Code and Hyper-parameters

For classic fine-tuning, we adopt and extend MT-DNN [2] codebase[1] while retaining most of the existing hyper-parameters in the package.

In the absence of any validation set for hyper-parameter tuning, we run each model for a fixed numbers of epochs and use the same batch-size and learning rate as follows. We use a batch-size of 32 for all the models except DeBERTa that uses 16 due to memory constraints. We use a learning rate of $5e-5$ for all the models and a maximum sequence length of 512. We use 20 epochs for few-shot fine-tuning and report the results from the last epoch. For fully supervised fine-tuning, we run each model for 5 epochs and report results from the last one.

Similarly, for prompt fine-tuning, we adapt and built upon LM-BFF [1] codebase[2] with the following hyper-parameters. For few-shot settings, we train all models for 20 epochs with learning rate 10^{-5} and batch-size 8. For fully supervised prompt-tuning, we train each model for 30000 steps and report the performance of last model checkpoint. All experiments are trained with the default prompt pattern and verbalizer with demonstrations randomly selected from the corresponding training sets.

A.6 Broader Impact

This benchmark is likely to increase the progress of NLU models and drive the development of general-purpose language systems especially for domains with limited resources. While it is not only expensive to acquire large amounts of labeled data for every task and language, in many cases, we cannot perform large-scale labeling due to access constraints from privacy and compliance concerns.

[1] https://github.com/microsoft/MT-DNN
[2] https://github.com/princeton-nlp/LM-BFF

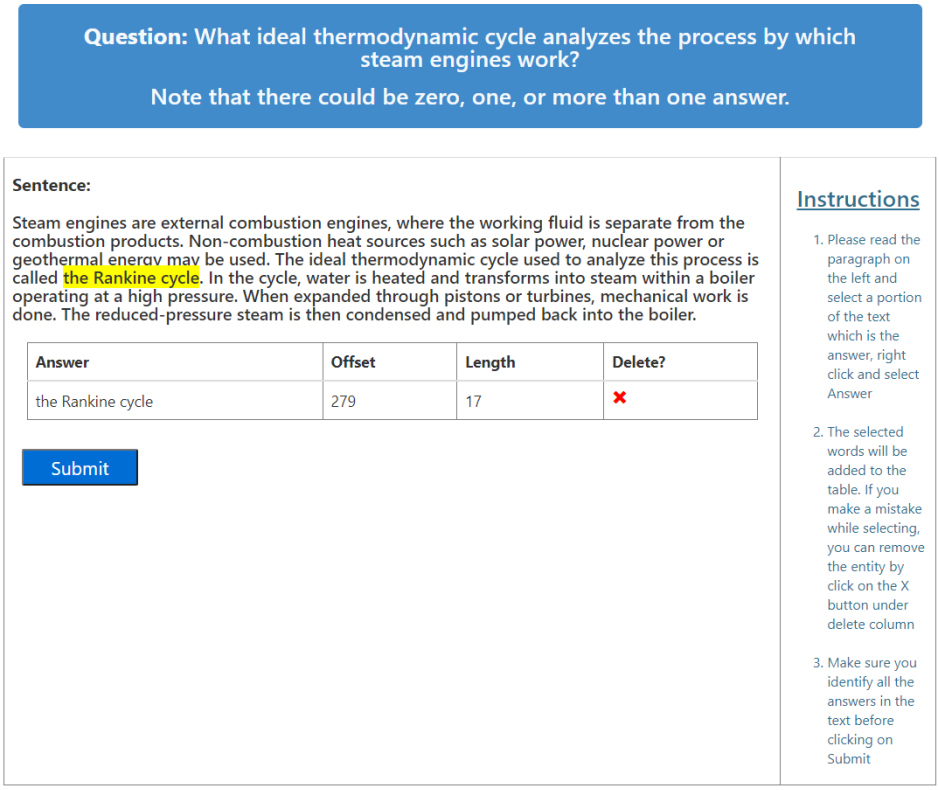

Figure 3: Human evaluation platform for SQuADv2 task.

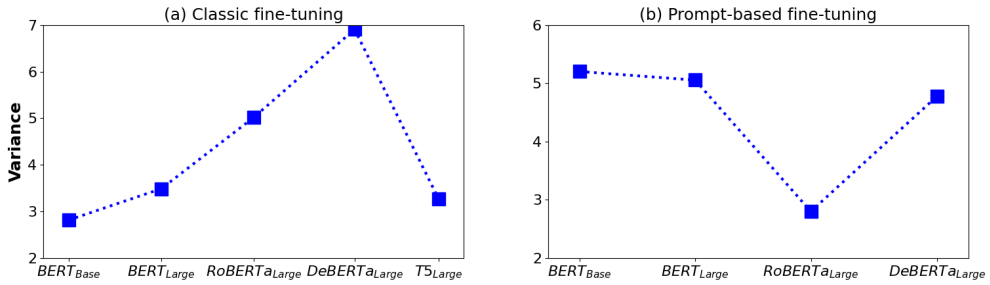

Figure 4: Variance in few-shot model performance with size. (Left) Standard fine-tuning averaged over all tasks and shots; (Right) Prompt-based tuning averaged over all shots in SST-2 and MNLI.

The latter concerns are amplified when dealing with sensitive user data for various personalization and recommendation tasks. Our work provides a benchmark and design principles to evaluate the progress of NLU models and systems in such low-resource, specifically, few-shot settings.

Limitations. Our benchmark primarily serves as an evaluation framework for few-shot learning of large pre-trained models. Therefore, these limitations primarily apply to the candidate models. Few-shot models may suffer from associated societal implications of automation ranging from job losses for workers who provide annotations as a service as well as for other industries relying on human labor. Additionally, they suffer from similar concerns as with the use of NLU models by malicious agents for propagating bias, misinformation and indulging in other nefarious activities.

However, many of these concerns can also be alleviated with few-shot learning to develop better detection models and mitigation strategies with only a few representative examples of such intents.

References

[1] Tianyu Gao, Adam Fisch, and Danqi Chen. "Making Pre-trained Language Models Better Few-shot Learners". In: *Association for Computational Linguistics (ACL)*. 2021.

[2] Xiaodong Liu, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, et al. "The Microsoft Toolkit of Multi-Task Deep Neural Networks for Natural Language Understanding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2020, pp. 118–126.

[3] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. "Adversarial NLI: A New Benchmark for Natural Language Understanding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.