DATA-EFFICIENT FINETUNING USING CROSS-TASK NEAREST NEIGHBORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models trained on massive prompted multitask datasets like T0 (Sanh et al., 2021) or FLAN (Wei et al., 2021) can generalize to tasks unseen during training. We show that training on a carefully chosen subset of instances can outperform training on all available data on a variety of datasets. We assume access to a small number (250-1000) of unlabeled target task instances, select their nearest neighbors from a pool of multitask data, and use the retrieved data to train target task specific models. Our method is more data-efficient than training a single multitask model, while still outperforming it by large margins. We evaluate across a diverse set of tasks not in the multitask pool we retrieve from, including those used to evaluate T0 and in addition, more complex tasks including legal and scientific document QA. We retrieve small subsets of P3 (the collection of prompted datasets from which T0's training data was sampled) and finetune T5 models that outperform the 3-billion parameter variant of T0 (T0-3B) by 8-30% on 11 out of 12 evaluation datasets while using at most 2% of the data used to train T0-3B. These models also provide a better initialization than T0-3B for few-shot finetuning on target-task data, as shown by a 3-23% relative improvement over few-shot finetuned T0-3B models on 8 datasets.

1 INTRODUCTION

Finetuning large models with data from a diverse set of tasks, augmented to include brief descriptions of the tasks (i.e., prompts) has been shown to help models generalize to unseen tasks (Wei et al., 2021; Sanh et al., 2021). This cross-task generalization capability is particularly helpful in cases where it is expensive to collect labeled target task training sets. Prior work trained single models with as much prompted data as possible — for example, Sanh et al. (2021) train a model on roughly 11 million instances (counting different prompt variations). The training datasets were selected without using any information about the target tasks with the goal of allowing models to generalize to new tasks from instructions alone, making the evaluation "zero-shot". However, it is unclear if all the training data is required for doing well on any given target task. Furthermore, given that neural network models have previously been shown to suffer from negative interference (where in training on more datasets results in worse performance on certain downstream tasks) in multitask setups (Aribandi et al., 2022) and benefit from pretraining on domain-relevant data (Gururangan et al., 2020; Phang et al., 2018), it is possible that training only on relevant prompted data could further improve task generalization.

Based on this hypothesis, we seek to find small subsets of relevant training data in the massive pool of multitask data that cause the models to generalize better to a given target task than the rest of the pool. Manually finding relevant training data in a massive pool of data is infeasible since it is not obvious which of the source tasks are relevant for a given target task, and within a source task dataset, not all instances may be equally useful for generalization to the target task (see Section 5.1). Hence we rely on a simple method to *automatically* select these subsets. Additionally, as only some samples within a given dataset may be relevant to a target task, we select per-instance rather than per-dataset, unlike prior work, which tries to identify useful datasets for transfer learning (Aribandi et al., 2022; Phang et al., 2018) and train on all data within the chosen datasets. We assume access to a small number of *unlabeled* target task instances and use these to retrieve *cross-task nearest neighbors* - labeled instances from the massive pool of data most similar to our unlabeled target task instances. The similarity is computed as the distance between the representations produced by

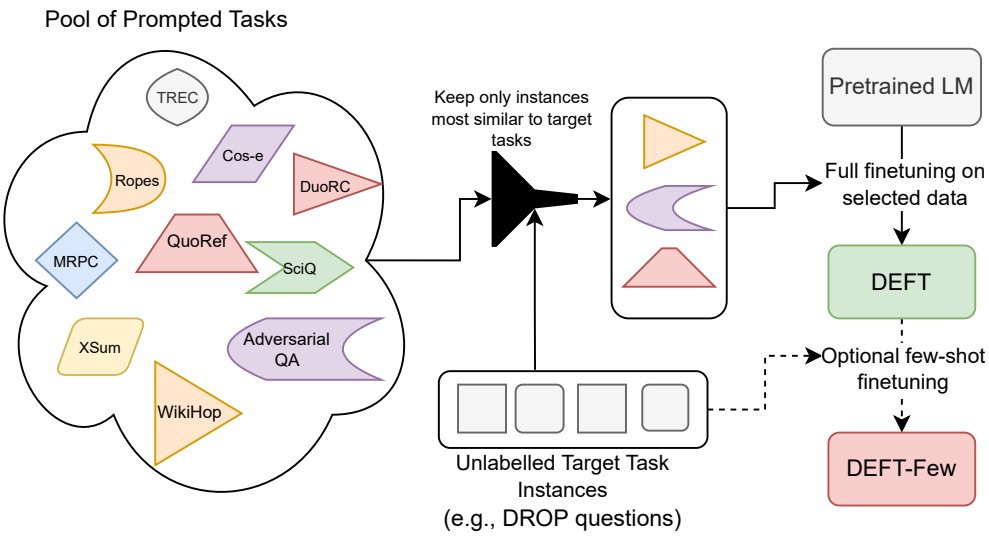

Figure 1: Overview of the DEFT method. Given some unlabeled target-task instances, we find the most similar instances in a large pool of multitask data. We train a model on these instances. If we have access to labeled data, we optionally few-shot finetune the DEFT model.

the encoder of a pretrained seq2seq model. We then finetune target task specific models on these neighbors alone, without using any target task specific labeled data. We hope that the similarity between the cross-task neighbors and our target task data will enable greater generalization to our target task, with dissimilar examples that may cause interference removed from the training mixture.

We run experiments with T5 (Raffel et al., 2020) models, and use Public Pool of Prompts (P3) (Sanh et al., 2021) as the pool of prompted multitask data from which to retrieve cross-task nearest neighbors. We evaluate on 9 datasets originally used to evaluate T0 (a collection of natural language understanding and commonsense tasks), as well as 3 additional datasets with more varied domains (legal, NLP, math domains). Our findings are as follows:

- For all the target tasks, we find that their cross-task nearest neighbors in P3 are much more relevant as training data than the rest of the pool—training T5 models, sometimes even variants smaller than T0-3B, on these subsets yields models with performance 3-20% better than T0-3B evaluated zero-shot across 8 datasets.
- For some target tasks on which T0-3B performs close to random chance, T5 models of the same size trained using cross-task nearest neighbors perform significantly above chance, confirming our hypothesis that massive multitask prompted training could lead to negative interference between tasks.
- When target task labeled data is available for few-shot finetuning, we find that T5 models trained with cross-task nearest neighbors provide better initialization for parameter-efficient finetuning methods than T0-3B, performing 3-23% better than T0-3B with few-shot finetuning across 8 out of 9 datasets.
- An analysis of what relevant data gets retrieved shows that most of the tasks in the massive pool of multitask data are not retrieved for any target tasks, confirming our hypothesis that only a small subset of data within the pool is relevant to any given target task.

These findings suggest that prior work may have underestimated the cross-task generalization capabilities of large multitask models and that training these models on all the available multitask prompted data results in negative interference, even in relatively large (3 billion parameter) models. Furthermore, they show that training large models on all available multitask data is not optimal, with higher performance possible through better-curated and much smaller training sets.

2 RELATED WORK

Multi-task Transfer Models Training on large multi-task mixtures is a common trend within NLP, with most existing approaches first training a pretrained language model on a large collection of tasks, and then evaluating these models in either zero or few-shot settings on a collection of held-out datasets (Wei et al., 2021; Sanh et al., 2021; Khashabi et al., 2020; Mishra et al., 2021; Aribandi et al., 2022). Most approaches do not customise their task selection to downstream tasks and assume no knowledge of the target tasks ahead of time, instead focusing on building a single model most applicable to any arbitrary evaluation task. In contrast, we show that if assume access to unlabeled target task instances, we can make much better use of the multitask data, selecting only instances useful to a given task.

Retrieval-based methods for NLP Adding retrieval components to language models has been shown (Khandelwal et al., 2019; Guu et al., 2020; Lewis et al., 2020) to augment their generalization capabilities by externalizing memorization. In contrast to prior work in this direction that mostly focused on language modeling as the end task, we evaluate on a variety of language understanding tasks. The work from Shi et al. (2022) used retrieval-based methods for classification tasks by heuristically mapping the label space of the end-tasks to that of the predicted next words of the nearest neighbors from a language model. We instead finetune the models on the nearest neighbors. Beyond language modeling, Lin et al. (2022) similarly use nearest neighbors to retrieve relevant cross-task data to improve zero-shot performance, but make use of an additional re-ranker model and multitask trained model to retrieve useful instances. In contrast, we use non-multitask trained models for retrieval and training, and show this outperforms using generalist multitask models in both zero and few-shot settings.

Parameter-efficient fine-tuning methods In contrast to work that focused on finetuning fewer parameters in large models to adapt them to new tasks (Houlsby et al., 2019; Hu et al., 2021; Liu et al., 2022), our proposal is a *data*-efficient pretraining method for obtaining task-specific models without using any target task labels. Our method is complementary to parameter-efficient methods, and they can both be used in conjunction, as shown in section 4.3.

Instance attribution Our approach works by identifying the most relevant training examples for a given data point, which is called *instance attribution*. Prior work (Koh & Liang, 2017; Yeh et al., 2018; Pruthi et al., 2020; Han & Tsvetkov, 2022) used instance attribution methods to interpret predictions of neural network models. These methods generally relied on the gradients of the model to identify the effect specific data points, either in pretraining or finetuning stage, have on the model's predictions. Our method for identifying cross-task neighbors is simpler because we do not use gradients and we do not even rely on the labels of the data. Results from Pezeshkpour et al. (2021) show that instance attribution based on similarity between the model's representations are comparable to gradient-based approaches in terms of finding the most important training data points.

3 DATA EFFICIENT FINETUNING ACROSS MULTIPLE TASKS

Given a large collection of labeled prompted data (i.e., data converted into a text-to-text form, with task instructions included in the plain-text input, e.g., P3), our core hypothesis is that some tasks in this massive pool of data are more similar to a given target task than others. Given a target task, we assume we have access to a small amount of *unlabeled* target task data, which is often much easier and cheaper to collect than labeled data (see section 5.2). Our aim is to find a relevant subset of data from our pool given a single target task, ideally allowing us to train a model using this subset that outperforms a similar model trained on the entire pool of data.

Manually identifying the relevant subsets of these datasets is not feasible since task boundaries are usually not clearly defined in NLP, and it is hard to interpret what skills get transferred when a model is trained on one dataset and evaluated on other. Hence, we use similarity between the pretrained model's representations to compute relevance. We encode all instances in the large pool of multitask data with a pretrained language model (e.g., T5), and build a search index over the resulting representations. Given small amounts of unlabeled target task data, we retrieve relevant multitask subsets from the index, which we call **cross-task nearest neighbors** of the target tasks.

We then build task-specific models by finetuning the pretrained models on the cross-task neighbors. We refer to this approach as **Data-Efficient FineTuning**, or **DEFT** for short.

We evaluate our approach both in cases where no labeled data is available, and when a few (20-70) annotated labels are available. In the former case, we simply use the unlabeled data for retrieval and evaluate the resulting DEFT model 'zero-shot' on the target task. In the latter case, we first train a DEFT model and then perform parameter-efficient few-shot tuning using IA3 (Liu et al., 2022) to make use of the labeled data.

Retrieving cross-task nearest neighbors To retrieve the most similar instances to a given set of target task instances, we first build an index over the massive pool of multi-task data for efficient retrieval, encoding samples using a pretrained encoder. Then, given a set of query instances Q, we retrieve our subset of similar data by computing a union of the k-nearest neighbors to all $q \in Q$. Note that there may be an overlap between the sets of nearest neighbors retrieved for different queries, and hence $|R| \leq |Q| * k$, where R is the retrieved subset. Empirically, we find $|R|$ tends to be 5-50$\times$ smaller than $|Q| * k$ due to this overlap.

Data-Efficient Training Given a retrieved set of data R, we can then simply train a pretrained language model on this mixture of data using a cross-entropy loss, as all data is in a unified text-to-text prompted format. This training is similar to the multitask prompted training of T0 (Sanh et al., 2021). We refer to models trained on R as DEFT models. If we have no labeled data available, we directly evaluate these models on our target tasks.

Parameter-efficient Few-shot Finetuning For the case where few annotated labels are available, we make use of parameter-efficient few-shot finetuning. For this, we take our multi-task trained DEFT checkpoints and finetune them using IA3 (Liu et al., 2022) on task-specific few-shot data. Concretely, given a trained transformer model, we introduce three new vectors l_{k}, l_{v}, and l_{ff} into the attention and feed-forward mechanisms of each layer as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q(l_{\text{k}} \odot K^T)}{\sqrt{d_k}}\right)(l_{\text{v}} \odot V) \tag{1}$$

$$\text{Feed-forward}(x) = (l_{\text{ff}} \odot f(W_1 x))W_2 \tag{2}$$

We initialize these vectors with all ones and only update them during the few-shot finetuning. This provides an efficient method of further training our DEFT models on task-specific data and has been shown to outperform full finetuning in the few-shot setting (Liu et al., 2022).

4 EXPERIMENTS

4.1 SETUP & HYPERPARAMETERS

Indexing P3 We construct an index of P3 data using FAISS (Johnson et al., 2019), a library for efficient similarity search over dense vectors. We use a Hierarchical Navigable Small World index (Malkov & Yashunin, 2020) to approximate the k-nearest neighbor search. To keep the size of the index manageable, we use Product Quantization Jegou et al. (2010) and reduce the dimensionality of the encoded representations using an Optimized Product Quantization transform Ge et al. (2013). We encode our instances using the T5 v1.1 model with extra language model pretraining introduced by Lester et al. (2021). For all experiments, we match the size of the encoder used to index data and the size of downstream models trained on this data (e.g., if we train a T5-XL sized model, we use T5-XL to index and retrieve the data). For models compared against T0-3B (i.e., the models shown in Table 1 and Table 2), we use the subset of P3 used to train T0 as our pool of multitask data. For other models, we instead use the pool used to train T0++, but remove the datasets we evaluate on from the mixture.

DEFT Training Following T0, we start with the T5 v1.1 model with extra language model pretraining. Unless otherwise stated, we use the 'XL' variant with 3 billion parameters across our experiments. When training on cross-task nearest neighbors, we train for 5 epochs with a batch size of 8 using the Adam optimizer (Kingma & Ba, 2015) and a learning rate of 5e-5. We use a linear warmup schedule for the first 10% of the total training steps and linear decay for the rest of training.

Few-shot training We follow the settings suggested by Liu et al. (2022): training for 1000 steps with a batch size of 8. We use the Adafactor optimizer with a maximum learning rate of 3e-3 and a linear decay schedule with 60 warmup steps. We only update the IA3 vectors during training.

Evaluation Datasets We evaluate on a set of 9 datasets used to evaluate T0 (ANLI R1/2, CB, HellaSwag, Story Cloze, WinoGrande, WSC, COPA, WiC), which include natural language inference and commonsense reasoning datasets. In addition to the T0 evaluation datasets, we also evaluate on three additional datasets in diverse domains: CaseHold (Chalkidis et al., 2022; Zheng et al., 2021), a legal QA dataset, DROP (Dua et al., 2019), a QA dataset that requires discrete operations, and a subtask of Qasper (Dasigi et al., 2021), a QA dataset over entire NLP papers. Qasper has two subtasks—selecting paragraphs in the paper that provide evidence for answering the questions, and generating the answers. We focus on the former because it was shown to more difficult of the two, and convert it into a binary classification task where the inputs are combinations of questions and single paragraphs. We refer to this subtask as *QasperEvidence* henceforth, and evaluate model performance in terms of document-level F1 as described in Dasigi et al. (2021). For evaluation and few-shot training, we convert all datasets to a prompted text-to-text format[1] either using the prompt templates from P3 for the T0 evaluation datasets or an original prompt for the other datasets. For CaseHold, DROP, and QasperEvidence we randomly split out 1000 examples from the existing validation sets to use for retrieval, and use the remaining data for evaluation. For all other datasets, we simply retrieve using up to 1000 randomly chosen examples from the training splits. We provide a list of these prompts and further dataset details in the supplementary materials.

Model Evaluation Following Sanh et al. (2021) and Brown et al. (2020), we calculate accuracy on all datasets except DROP using *rank classification*, where we pick the answer with lowest loss across possible answer choices given the instance input as the model prediction. As DROP is an QA dataset that requires selecting spans or generating numbers, and does not have answer choices, we simply generate the prediction using greedy decoding.

Baselines For zero-shot evaluation, we primarily compare against 3 baselines: 1) *T0-3B*, trained on about 10% of the P3 data[2], 2) *random* a model trained on a random selection of P3 data the same size as the subsets selected by DEFT, and 3) T5-XL not finetuned any further. For few-shot settings, we compare T0-3B with additional few-shot training with DEFT checkpoints trained on subsets chosen using (a) 1000 unlabeled instances and (b) the instances used in the few-shot training without labels. This means (b) uses no additional data, labeled or unlabeled, compared to T0-3B with few-shot finetuning.

4.2 Data-Efficient Fine-Tuning vs. Massive Multitask Training

We first assume we have access *only* to unlabeled task-specific data, and so cannot train on any target task labeled data. We sample 1000 unlabeled instances[3] per dataset and retrieve the 500 nearest neighbors of each instance. We then train dataset-specific models on each of the retrieved sets. As seen in Table 1, our DEFT-T5 models consistently outperform T0-3B and other baselines, with a median relative improvement of 13% over T0-3B. We also see that T5-base sized models also improve over baselines in Table 1 - the DEFT-T5-base models have a median relative improvement of 8% over the random baseline. All DEFT models are trained on subsets of P3 consisting of 0.1-0.01% of all P3 data. This confirms our hypothesis that training on a well-chosen subset of P3 is more beneficial for target task performance than training on a uniform sample of all available data.

Remarkably, DEFT-XL outperforms the majority baselines on two target datasets (QasperEvidence and ANLI R2) where T0-3B does not, and DEFT-base on one (QasperEvidence). This observation further confirms that multitask models trained on uniformly sampled data might be suffering from negative interference between tasks.

[1]For example, ANLI instances were converted to '{premise} Question: {hypothesis} True, False, or Neither?', with the answers as 'true', 'false', or 'neither'.

[2]Sanh et al. (2021) report that they train T5-XL on at most 500K instances per prompted dataset in P3, which amounts to about 10% of the pool.

[3]We randomly sample these from the train split. For tasks with training splits smaller than 1000, we just use all training data for retrieval.

Task	DEFT-XL	T0-3B	Rand-XL	T5-XL	DEFT-base	Rand-base	T5-base	Maj. Class
CaseHold	**37.2**	30.9	19.0	11.4	**17.4**	14.2	11.4	6.6
DROP	**31.0**	27.4	24.3	11.3	1.9	**2.2**	0.2	-
QasperEv.	**22.5**	19.9	17.9	8.2	**22.1**	13.8	8.2	19.9
ANLI R1	**39.8**	35.0	35.3	32.9	29.6	**33.3**	32.9	33.4
ANLI R2	**37.5**	32.6	35.3	33.5	31.9	33.3	**33.5**	33.4
CB	**60.7**	58.9	**60.7**	44.6	**73.2**	50.0	44.6	50.0
HellaSwag	**33.1**	28.2	27.4	23.0	**26.7**	24.8	23.0	25.7
StoryCloze	**95.3**	86.5	79.1	53.0	**84.8**	59.0	53.0	51.4
WinoGrande	50.6	50.0	49.2	**50.8**	-	-	-	50.4
WSC	39.4	**50.0**	47.1	36.3	**36.5**	**36.5**	36.3	63.5
COPA	**95.0**	74.0	80.0	60.0	**70.0**	43.0	60.0	55.0
WiC	**54.9**	51.1	51.4	51.7	47.0	50.5	**51.7**	50.0
Average	**49.7**	45.4	43.9	34.7	**40.1**	32.8	32.3	-

Table 1: Performance of XL (3B) and base size models across datasets. 'Rand' refers to performance of models trained on randomly chosen P3 subsets of equivalent size to the ones chosen by DEFT. 'T5' refers to performance of a non-finetuned T5 model. 'Maj. Class' refers to the performance of predicting only the majority class label. DROP and QasperEv. results are F1 scores, CaseHold micro F1, and all else accuracy.

	T0-3B+IA3	DEFT-Few (1k Queries)	DEFT-Few (20-70 Queries)
ANLI R1	$44.9_{3.0}$	$47.3_{1.4}$	$\mathbf{46.7_{1.9}}$
ANLI R2	$39.5_{1.7}$	$40.8_{2.8}$	$41.1_{3.1}$
CB	$78.9_{3.9}$	$82.5_{2.6}$	$86.9_{4.1}$
HellaSwag	$34.7_{0.6}$	$\mathbf{42.5_{2.1}}$	$\mathbf{46.6_{2.0}}$
StoryCloze	$93.0_{0.6}$	$\mathbf{96.2_{0.2}}$	$\mathbf{96.4_{0.2}}$
WinoGrande	$50.6_{1.3}$	$\mathbf{55.9_{3.0}}$	$\mathbf{56.6_{3.4}}$
WSC	$64.8_{3.5}$	$63.3_{5.2}$	$58.0_{3.6}$
COPA	$82.0_{2.7}$	$\mathbf{95.4_{1.5}}$	$\mathbf{93.0_{3.0}}$
WiC	$54.9_{1.9}$	$57.7_{2.9}$	$55.6_{2.3}$
Average	60.4	**64.6**	**64.6**

Table 2: Performance of IA3 few-shot finetuned models using T0-3B and DEFT XL-size checkpoints. We report the mean over 5 runs for T0-3B+IA3 and DEFT-Few (1k Queries) and 3 runs for DEFT-Few (20-70 Queries) with the standard deviation as subscript. **Bolded** numbers indicate that value is statistically significantly better than T0-3B+IA3 ($p < .05$ in a two-tailed paired t-test).

The only dataset for which our approach performs worse than T0-3B is WSC (the Winograd Schema Challenge). However, we note that this dataset has a small number of evaluation samples (104) and hence these differences may not be significant. Relatedly, T0-3B underperforms a majority class baseline on this dataset.

4.3 Few-shot Finetuning of DEFT Models

Next, we assume we are able to label a small number of task-specific examples, and further train our DEFT models. We reuse the models trained in section 4.2 and further train them on the few-shot data used by Liu et al. (2022) using IA3. As seen in table 2, DEFT models with few-shot finetuning ('DEFT-Few (1k Queries)') perform on average 7% better than T0-3B models with few-shot finetuning ('T0-3B+IA3'), with statistically significant gains on 4 datasets. This shows that DEFT models serve as better starting points for few-shot finetuning than T0-3B, providing similar or better performance across all datasets despite being exposed to much less training data. Notably, DEFT-Few significantly outperforms T0-3B+IA3 on WinoGrande, for which zero-shot DEFT did not significantly outperform zero-shot T0-3B. These results show that DEFT models are more amenable to few-shot fine-tuning than T0-3B.

	anli_r1	anli_r2	casehold	cb	copa	drop	hellaswag	qasper	story_cloze	wic	winogrande	wsc	P3
adversarial_qa	0.28	0.28	0.15	0.04	0.00	0.47	0.07	0.24	0.00	0.01	0.00	0.03	0.02
ag_news	0.02	0.03	0.00	0.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.04
amazon_polarity	0.04	0.03	0.01	0.02	0.00	0.00	0.08	0.04	0.00	0.00	0.00	0.00	0.04
app_reviews	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
cnn_dailymail_3.0.0	0.01	0.00	0.27	0.02	0.00	0.05	0.05	0.01	0.01	0.00	0.00	0.00	0.04
common_gen	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.05	0.01	0.04
cos_e_v1.11	0.00	0.00	0.00	0.02	0.53	0.00	0.01	0.01	0.00	0.01	0.02	0.01	0.01
cosmos_qa	0.00	0.00	0.00	0.26	0.02	0.00	0.22	0.01	0.32	0.00	0.02	0.18	0.04
dbpedia_14	0.09	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
dream	0.00	0.00	0.00	0.16	0.02	0.00	0.01	0.01	0.01	0.01	0.02	0.03	0.00
duorc_ParaphraseR	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.04
duorc_SelfRC	0.01	0.01	0.04	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.04
gigaword	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
glue_mrpc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.02	0.02	0.00
glue_qqp	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.21	0.00	0.60	0.00	0.01	0.04
imdb	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
kilt_tasks_hotpotqa	0.18	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
multi_news	0.00	0.00	0.30	0.01	0.00	0.02	0.02	0.02	0.00	0.00	0.00	0.00	0.03
paws_labeled_final	0.08	0.06	0.00	0.02	0.00	0.00	0.00	0.05	0.00	0.11	0.09	0.13	0.04
qasc	0.00	0.00	0.00	0.03	0.02	0.00	0.00	0.03	0.00	0.06	0.06	0.05	0.01
quail	0.00	0.00	0.08	0.03	0.00	0.00	0.12	0.03	0.00	0.00	0.00	0.00	0.02
quarel	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.03	0.02	0.03	0.05	0.10	0.00
quartz	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.02	0.02	0.00
quoref	0.01	0.01	0.03	0.00	0.00	0.21	0.01	0.01	0.00	0.00	0.00	0.00	0.02
ropes	0.02	0.02	0.01	0.02	0.00	0.04	0.27	0.08	0.07	0.00	0.01	0.03	0.01
rotten_tomatoes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
samsum	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.01
sciq	0.01	0.00	0.00	0.00	0.01	0.00	0.02	0.02	0.00	0.01	0.00	0.01	0.01
social_i_qa	0.00	0.00	0.00	0.17	0.39	0.00	0.05	0.01	0.52	0.03	0.62	0.32	0.02
trec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.01	0.00	0.00	0.00
wiki_bio	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
wiki_hop_original	0.04	0.04	0.01	0.00	0.00	0.11	0.00	0.01	0.00	0.00	0.00	0.00	0.04
wiki_qa	0.17	0.18	0.01	0.02	0.00	0.00	0.00	0.15	0.00	0.01	0.00	0.01	0.01
wiqa	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.03
xsum	0.01	0.01	0.07	0.01	0.00	0.07	0.01	0.01	0.00	0.00	0.00	0.00	0.04
yelp_review_full	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.04

Figure 2: Proportion of the retrieved training data for each evaluation dataset (columns) that comes from each dataset in P3 (rows). The final column shows these values for all of P3.

Few-shot Retrieval In this experiment we evaluate DEFT in a setting where we have access only to a small number of target-task labeled examples (exactly what is available to T0-3B+IA3), and no additional unlabeled examples. We use the few-shot data splits without labels to retrieve cross-task neighbors, finetune T5 models on the retrieved data, and then finally finetune using IA3 on the labeled few-shot data itself. In order to make up for the smaller query set, we retrieve the closest 2000 neighbors per query instance. As seen in Table 2, this still results in a model that outperforms T0-3B with few-shot tuning ('DEFT-Few (20-70 Queries)'). Crucially, this shows that DEFT followed by few-shot finetuning may be a better alternative to few-shot finetuning T0-3B even when both the methods have exactly the same target-task information available.

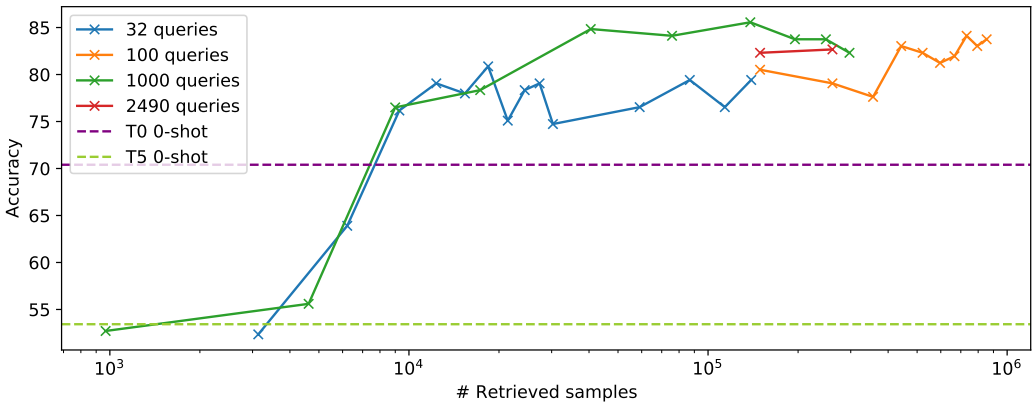

Figure 3: RTE accuracy by number of P3 samples retrieved for T5 XL across a varying number of neighbors.

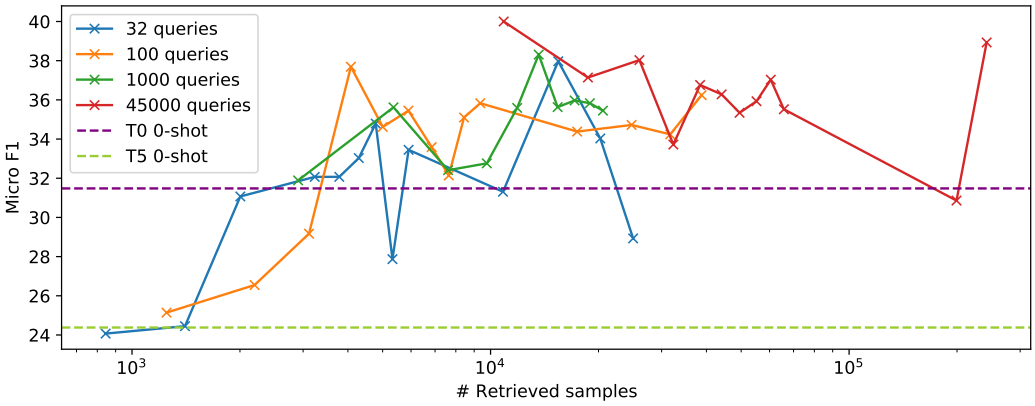

Figure 4: CaseHold micro F1 by number of P3 samples retrieved for T5 XL across a varying number of neighbors.

5 Analysis

5.1 Cross-Task Retrieval

What gets retrieved? We show what source datasets get selected during retrieval for each evaluation dataset in figure 2. For most target datasets, the majority of source datasets are *not* selected, further strengthening our hypothesis that much of the massive multitask pool is not relevant to a given target task, and no single mixture of datasets is optimal for all target tasks. We additionally find that no more than 27% of all instances within any source dataset is retrieved, suggesting that our approach is also effective at finding relevant subsets of data *within* large datasets.

Retrieval Hyperparameters When retrieving cross-task data, the amount and quality of data retrieved is highly dependent on the *query size* (i.e. the number of task-specific instances used for retrieval) and *number of neighbors* (i.e. the number of cross-task samples retrieved per task-specific instance). In figures 3 and 4, we show the effect of varying both query size and the number of neighbors on dataset performance on RTE (Dagan et al., 2005)[4] and CaseHold respectively. We sweep our query size from 32 to using the entire training set and sweep the number of neighbors retrieved from 1 to 5000. We find that increasing the amount of data retrieved, whether through increasing the number of neighbors or query set size, results in improved performance up to a point, and then

[4]We retrieve and evaluate on RTE just like the T0 evaluation tasks mentioned previously.

Unlabeled Examples	Labeled Examples	DEFT-T5	Full-finetuning
1000	0	24.4	-
0	308	-	19.9
1000	32	30.19	-
0	340	-	19.9

Table 3: Performance of DEFT-T5 and full-finetuning methods with the same annotation budget used for obtaining either labeled or unlabeled data or a combination for QasperEvidence. Using the limited budget towards obtaining more unlabeled examples for DEFT performs better. See section 5.2 for details.

either plateaus or decreases, providing evidence for our hypothesis that using 'too much' data can result in reduced downstream performance due to negative interference.

5.2 Practicality of assuming access to unlabeled data

Contrary to prior work, our approach assumes access to unlabeled data. This is a practical assumption given that unlabeled data is often readily available or is far cheaper to acquire than labeled data. This is especially true for tasks such as Qasper or CaseHold, which require experts to carefully read (sometimes quite long) texts to provide labels. We argue that DEFT's use of unlabeled data can make it a cost-efficient method to obtain a well-performing task-specific model when the data labeling budget is limited. We evaluate the cost-efficiency of using unlabeled data with DEFT using a case study with the QasperEvidence dataset.

We study the scenario where QasperEvidence data was collected and assume we have access to P3 and DEFT to make efficient use of it. Obtaining labeled instances for QasperEvidence cost 3.25 times acquiring unlabeled (question-paragraph) instances[5]. We first compare (Table 3 top) the performance on the test set of a T5-XL model trained on a small number of labeled instances with a DEFT-T5 model trained on cross-task nearest neighbors of 3.25 as many unlabeled instances. The DEFT setup makes it easy to use target-task labeled data when available. Table 3 (bottom) shows a comparison where the DEFT-T5 model is trained on a combination of some labeled instances and the cross-task nearest neighbors of the unlabeled instances, and the T5-XL model is trained fully on labeled data that would cost the same as the combination we assume DEFT-T5 has access to. As it can be seen, DEFT-T5 yields better models in both these cases.

6 Conclusion

In this work, we propose Data-Efficient FineTuning, a novel method for efficiently using multitask data by training task-specific models using only a small amount of unlabeled target task data. We use the unlabeled data to select subsets of the multitask data, and train models on these subsets. The resulting models outperform same-sized models trained on all available data (e.g., T0), suggesting that negative interference is still present even in large massively multi-task models. Our approach also results in models that work as better starting points for few-shot finetuning, outperforming models trained on more data. Our approach still performs strongly even when as few as only 32 unlabeled examples are available. Overall, our results strongly suggest that training on all available data, even with large models, is not the optimal choice for most downstream tasks, and that instead focusing on ways to better curate datasets into higher-quality, but smaller, mixtures is potentially a better path forward.

References

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. Ext5: Towards extreme multi-task scaling for transfer learning. In *International*

[5]Based on an estimate provided by the authors of the dataset. Questions were written after only the abstracts of papers were read, and evidence selection required reading entire papers.

Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=Vzh1BFUCiIX.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dubln, Ireland, 2022.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *MLCW*, 2005.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.365. URL https://aclanthology.org/2021.naacl-main.365.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The CommitmentBank: Investigating projection in naturally occurring discourse. 2019. To appear in proceedings of Sinn und Bedeutung 23. Data can be found at https://github.com/mcdm/CommitmentBank/.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2946–2953, 2013.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pp. 3929–3938. PMLR, 2020.

Xiaochuang Han and Yulia Tsvetkov. Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. *arXiv preprint arXiv:2205.12600*, 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1896–1907, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.171. URL https://aclanthology.org/2020.findings-emnlp.171.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.6980.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, pp. 47, 2011.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

Bill Yuchen Lin, Kangmin Tan, Chris Miller, Beiwen Tian, and Xiang Ren. Unsupervised cross-task generalization via retrieval augmentation. *ArXiv*, abs/2204.07937, 2022.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv:2205.05638*, 2022.

Yu A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *CoRR*, abs/2104.08773, 2021. URL https://arxiv.org/abs/2104.08773.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pp. 46–51, 2017.

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.

Pouya Pezeshkpour, Sarthak Jain, Byron C Wallace, and Sameer Singh. An empirical comparison of instance attribution methods for nlp. *arXiv preprint arXiv:2104.04128*, 2021.

Jason Phang, Thibault Févry, and Samuel R. Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *ArXiv*, abs/1811.01088, 2018.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL-HLT*, 2019.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

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

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, aug 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. Nearest neighbor zero-shot inference. *arXiv preprint arXiv:2205.13792*, 2022.

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

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385268. doi: 10.1145/3462757.3466088. URL https://doi.org/10.1145/3462757.3466088.

Dataset	Retrieval	Eval	#Shots	Retrieval from
CaseHold (Zheng et al., 2021)	1000	2900	-	Validation
DROP (Dua et al., 2019)	1000	8535	-	Validation
QasperEvidence (Dasigi et al., 2021)	42771	43673	-	Validation
ANLI R1 (Nie et al., 2020)	1000	1000	50	Train
ANLI R2 (Nie et al., 2020)	1000	1000	50	Train
CB (De Marneffe et al., 2019)	250	56	32	Train
HellaSwag (Zellers et al., 2019)	1000	10003	20	Train
StoryCloze (Mostafazadeh et al., 2017)	1000	1871	70	Train
WinoGrande (Sakaguchi et al., 2021)	1000	1767	50	Train
WSC (Levesque et al., 2011)	554	104	32	Train
COPA (Roemmele et al., 2011)	400	100	32	Train
WiC (Pilehvar & Camacho-Collados, 2019)	1000	638	32	Train

Table 4: Size of splits used for experiments across datasets. '#Shots' indicates the number of shots used in few-shot experiments, and 'retrieval from' indicates which split we selected retrieval data from.

A Dataset Details

Sizes and Splits For each dataset used, we provide the number of retrieval and validation examples used in table 4. We also indicate if the retrieval data was split from the validation or training split. Note that when retrieval data is taken from the validation set, we do not evaluate on that data to ensure no validation data indirectly leaks through the retrieval process. We additionally provide the number of shots used for each dataset. We follow the number of splits used by Liu et al. (2022) and use the data shared by the authors[6].

Prompts Here is a list of the prompts used for each dataset. {x} indicates a space that is filled in by instance data.

- **CaseHold**: What is the correct holding statement for the following text? Text: {context} (A): {ending 1} (B): {ending 2} (C): {ending 3} (D): {ending 4} (E): {ending 5}
- **DROP**: Passage: {passage} Question: {question} Answer:
- **QasperEvidence**: Question: {question} Paragraph: {paragraph} Is the answer to the question in the paragraph? Answer Yes or No.
- **ANLI**: {premise} Question: {hypothesis} True, False, or Neither?
- **CB**: {premise} Question: {hypothesis} True, False, or Neither?
- **HellaSwag**: Complete the description with an appropriate ending: First, {context a} Then, {context b} ... (a) {ending 1} (b) {ending 2} (c) {ending 3} (d) {ending 4}
- **StoryCloze**: {input sentence 1} {input sentence 2} {input sentence 3} {input sentence 4} What is a possible continuation for the story given the following options ? - {answer 1} - {answer 2}
- **WinoGrande**: {sentence} What does the _ in the above sentence refer to? {option1} or {option2}?
- **WSC**: Passage: {text} Question: In the passage above, does the pronoun '{span 1}' refer to '{span 2}'? Answer:
- **COPA**: {premise} As a consequence... Help me pick the more plausible option: - {choice 1} - {choice 2}
- **WiC**: {sentence 1} {sentence 2} Question: Is the word '{word}' used in the same sense in the two sentences above? Yes, No?

[6]https://github.com/r-three/t-few/tree/master/data/few_shot