DReCa: A General Task Augmentation Strategy for Few-Shot Natural Language Inference

Shikhar Murty **Tatsunori B. Hashimoto** **Christopher D. Manning**
Computer Science Department, Stanford University
{smurty,thashim,manning}@cs.stanford.edu

Abstract

Meta-learning promises "few-shot" learners that can adapt to new distributions by repurposing knowledge acquired from previous training. However, meta-learning has thus far failed to achieve this in NLP due to the lack of a well-defined task distribution, leading to alternatives that treat datasets as tasks. Such an ad hoc task distribution has two negative consequences. The first one is due to a lack of quantity—since there's only a handful of datasets, meta-learners tend to overfit their adaptation mechanism. The second one is due to a lack of quality—since NLP datasets are highly heterogenous, many learning episodes have poor transfer between their support and query sets, which dis-incentivizes the meta-learner from adapting. To alleviate these issues, we propose DReCa (**D**ecomposing datasets into **Re**asoning **Ca**tegories), a simple method for discovering and using latent reasoning categories in a dataset, to form additional high quality tasks. DReCa works by splitting examples into label groups, embedding them with a fine-tuned BERT model and then clustering each group into reasoning categories. Across 4 NLI fewshot problems, we demonstrate that using DReCa improves the performance of meta-learners by 1.5–4 accuracy points.

1 Introduction

Over the last few years, we have seen tremendous progress on fundamental natural language understanding problems. At the same time, there is increasing evidence that these models learn superficial correlations that fail to generalize beyond the training distribution (Jia and Liang, 2017; Gururangan et al., 2018; McCoy et al., 2019). How can we move from doing well on datasets toward more human-like understanding of tasks?

A key desideratum for human-like understanding is few-shot adaptation. From a practical perspective, adaptation is central to many NLP applications since new words and concepts appear every month, leading to distribution shifts. People can effortlessly deal with these distribution shifts by learning these new concepts quickly and we would like our models to have similar capabilities. Recently, pre-trained transformers have led to impressive results on many NLP problems, but they still require 1000s of samples where humans might require only a few.

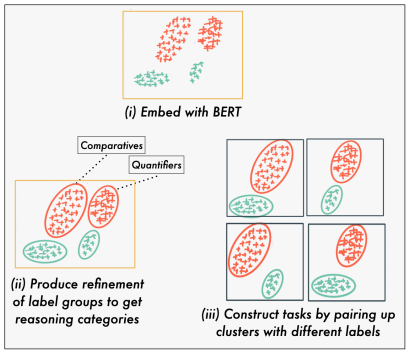


Figure 1: Overview of our approach. We embed all examples with BERT, and then apply k-means over each label group separately. Then, we group clusters from distinct label groups to form tasks.

Could these pre-trained transformers also be made to achieve few-shot adaptation? One promising direction is meta-learning. Meta-learning promises "few-shot" classifiers that can adapt to new tasks by repurposing skills acquired from training tasks. An important prerequisite for successful application of meta-learning is a task-distribution from which a large number of tasks could be sampled to train the meta-learner. While meta-learning is very appealing, applications in NLP have thus far proven challenging due to the absence of a well-defined set of tasks that correspond to re-usable skills. This has led to less effective ad hoc alternatives, like treating entire datasets as tasks.

Treating entire datasets as tasks has two major is-

sues. First, because there's only a small number of supervised datasets available for any NLP problem, we run into learner overfitting (Rajendran et al., 2020): due to the small number of training tasks, the meta-learner overfits its adaptation mechanism, and doesn't generalize to new tasks. Second, the heterogeneity of NLP datasets can lead to learning episodes that encourage memorization overfitting (Yin et al., 2020; Rajendran et al., 2020), a phenomenon where a meta-learner ignores the support set, and doesn't learn to adapt.

To improve the quality as well as quantity of tasks, we propose **D**ecomposing datasets into **Re**asoning **Ca**tegories or DReCa. DReCa is a *meta* data augmentation strategy that takes as input the original set of tasks (entire datasets), and then decomposes them to approximately recover the latent reasoning categories underlying these datasets. This allows us to approximately recover reasoning categories, e.g., various syntactic constructs within a dataset, linguistic categories such as quantifiers and booleans. These reasoning categories are then used to construct additional few-shot classification tasks, augmenting the original task distribution. We illustrate these steps in illustrated in Fig. 1. DReCa first embeds the examples using a BERT model fine-tuned over all the datasets. We then run k-means clustering over these representations to produce a refinement of the original tasks.

Experiments demonstrate the effectiveness of our simple approach. First, we adapt the classic sine-wave regression problem from Finn et al. (2017) to reflect the challenges of our setting, and observe that standard meta-learning procedures fail to adapt. However, a model that meta-learns over the underlying reasoning types shows a substantial improvement. Next we consider the problem of natural language inference (NLI). We show that meta-learners augmented with DReCa improve over baselines by 1.5–4 accuracy points across four separate NLI few-shot problems without requiring domain-specific engineering, or additional unlabeled data.

2 Related Work

Few Shot Classification in NLP. The goal of learning from few examples has been studied for various NLP applications. Common settings include few shot adapting to new relations (Han et al., 2018), words (Holla et al., 2020) domains (Bao et al., 2020; Yu et al., 2018; Geng et al., 2019) and language pairs (Gu et al., 2018). In these works, since task distributions are well defined, they do not have the same overfitting challenges. On the other hand, many works deal with fewshot adaptation in settings with no clear task distribution such as Dou et al. (2019); Bansal et al. (2019) but do not address meta-overfitting.

Overfitting and Task Augmentation. The memorization problem in meta-learning is studied in Yin et al. (2020) who propose a meta-regularizer to encourage the meta-learner to adapt, but this not directly applicable to NLP. Task Augmentation for mitigating overfitting in meta-learners is first studied in Rajendran et al. (2020) in the context of few-shot label adaptation. Hsu et al. (2019) propose CACTUs, a clustering based approach for unsupervised meta-learning in the context of few-shot label adaptation for images, but do not study meta-overfitting. Most closely related to our work is the recent work by Bansal et al. (2020). They propose SMLMT, a task augmentation strategy that require a large text corpus to construct augmented tasks. On the other hand, DReCa creates task augmentations based solely on the provided training data. In Section-6, we compare our model against SMLMT, and demonstrate comparable performance.

3 Setting

3.1 NLI

We consider the problem of Natural Language Inference or NLI (MacCartney and Manning, 2008; Bowman et al., 2015), also known as Recognising Textual Entailment (RTE) (Dagan et al., 2005). Given a sentence pair $x = (p, h)$ where p is referred to as the premise sentence, and h is the hypothesis sentence, the goal is to output a binary label $\hat{y} \in \{0, 1\}$ indicating whether the hypothesis h is entailed by the premise p or not. For instance, the sentence pair *(The dog barked, The animal barked)* is classified as entailed, whereas the sentence pair *(The dog barked, The labrador barked)* would be classified as not entailed. As shown in Table. 1, NLI datasets typically encompass a broad range of linguistic phenomenon. Apart from the reasoning types shown in Table. 1, examples may also vary in terms of their genre, syntax, annotator writing style etc. leading to extensive linguistic variability. Taken together, these factors of variation make NLI datasets highly heterogenous.

Reasoning types	Example
Restrictive Modifiers	*The boy with the green jacket went back* $\implies$ *The boy went back*
Intersective Adjectives	*The white rabbit ran* $\implies$ *The rabbit ran*
Comparatives	*Bill is taller than Jack* $\not\Longrightarrow$ *Jack is taller than Bill*
Negation	*The dog barked* $\not\Longrightarrow$ *The dog did not bark*
Coreference Resolution	*The man went to the restaurant since he was hungry* $\implies$ *The man was hungry*
(Negation, Comparatives)	*Bill is taller than Jack* $\implies$ *Jack is not taller than Bill*

Table 1: Some common reasoning types within NLI. These can also be composed to create new types.

3.2 Meta learning

The goal of meta-learning is to output a black box meta-learner $f\colon (\mathcal{S}_i, x_q^i) \mapsto \hat{y}$ that takes as input a *support* set $\mathcal{S}_i$ of labeled examples and a query point x_q^i and returns a prediction $\hat{y}$. In the usual meta-learning setting, these support and query sets are defined as samples from a task $\mathcal{T}^i$, which is a collection of labeled examples $\{(x^i, y^i)\}$. In N-way k-shot adaptation, each $\mathcal{T}^i$ is an N-way classification problem, and f is given k examples per label to adapt. A simple baseline for meta-learning is to train a supervised model on labeled data from training tasks, and then fine-tune it at test time on the support set. This can be powerful, but ineffective for very small support sets. A better alternative is episodic meta-learning, which explicitly trains models to adapt using training tasks

Episodic Training. In the standard setup for training episodic meta-learners, we are given a collection of training tasks. We assume that both train and test tasks are i.i.d. draws from a task distribution $\rho(\mathcal{T})$. For each training task $\mathcal{T}_i^{\text{tr}} \sim \rho(\mathcal{T})$, we create *learning episodes* which are used to train the meta-learner. Each learning episode consists of a support set and a query set $\mathcal{Q}_i = \{(x_q^i, y_q^i)\}$. To make predictions on a query x_q^i, the meta-learner f uses $\mathcal{S}_i$ to adapt. The goal of episodic meta-learning is to ensure that the meta-learning loss $\mathcal{L}(f(\mathcal{S}_i, x_q^i), y_q^i)$ is small on training tasks $\mathcal{T}_i^{\text{tr}}$. Since train tasks are i.i.d. with test tasks, this results in meta-learners that achieve low loss at test time.

Several algorithms have been proposed for meta-learning that follow this general setup, such as Matching Networks (Vinyals et al., 2016), MANN (Santoro et al., 2016), Prototypical Networks (Snell et al., 2017) and MAML (Finn et al., 2017). In this work, we use MAML as our meta-learner.

MAML. In MAML, the meta-learner f takes the form of gradient descent on a model $h_\theta\colon x \mapsto y$ using the support set,

$$f(\mathcal{S}_i, x_i^q) = h_{\theta_i'}(x_i^q) \tag{1}$$

where θ_i' denotes *task specific* parameters obtained after gradient descent. The goal of MAML is to produce an initialization θ, such that after performing gradient descent on h_θ using $\mathcal{S}_i$, the updated model $h_{\theta_i'}$ can make accurate predictions on $\mathcal{Q}_i$. MAML consists of an *inner loop* and an *outer loop*. In the inner loop, the support set $\mathcal{S}_i$ is used to update model parameters θ, to obtain task-specific parameters θ_i',

$$\theta_i' = \theta - \alpha \nabla_\theta \sum_{(x_s^i, y_s^i) \in \mathcal{S}_i} \mathcal{L}(h_\theta(x_s^i), y_s^i). \tag{2}$$

These task specific parameters are then used to make predictions on $\mathcal{Q}_i$. The outer loop takes gradient steps over θ such that *task-specific* parameters θ_i' perform well on $\mathcal{Q}_i$. Since θ_i' is itself a differentiable function of θ, we can perform this outer optimization using gradient descent,

$$\theta \leftarrow \text{Opt}\left(\theta, \nabla_\theta \sum_{(x_q^i, y_q^i) \in \mathcal{Q}_i} \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i)\right). \tag{3}$$

Here, Opt is an optimization algorithm typically chosen to be Adam. The outer loop gradient is typically computed in a mini-batch fashion by sampling a batch of episodes from distinct training tasks. The gradient $\nabla_\theta \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i)$ involves back-propagation through the adaptation step which requires computing higher order gradients. This can be computationally expensive so a first order approximation (FoMAML),

$$\nabla_\theta \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i) \approx \nabla_{\theta_i'} \mathcal{L}(h_{\theta_i'}(x_q^i), y_q^i) \tag{4}$$

is often used instead (Finn et al., 2017).

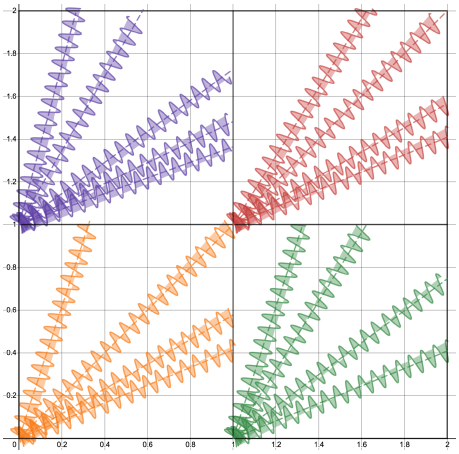


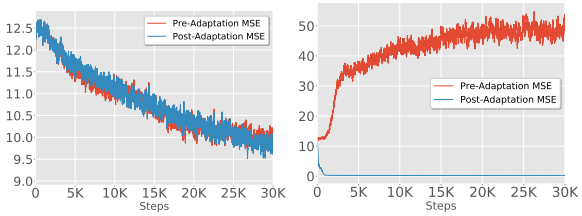
Figure 2: A snapshot of 4 datasets from our synthetic 2d sine wave regression problem. Each dataset is a unit square with multiple reasoning categories; A reasoning category is a distinct sinusoid along a ray that maps $x = (x_1, x_2)$ to y.

4 Overfitting in Meta Learning

As mentioned earlier training tasks in NLP are often entire datasets, due to the lack of a well-formed task distribution. This results in a small number of heterogeneous training tasks, which can lead to learner and memorization overfitting. Learner overfitting occurs when the meta-learner is exposed to a very small number of tasks at meta-training time causing it to not generalize to test tasks. Memorization overfitting when the meta-learner ignores its support set and doesn't learn to adapt at all. We illustrate memorization overfitting challenges through a simple few-shot generalization problem based on 2D sine wave regression.

Dataset. We extend a standard meta-learning toy regression problem from Finn et al. (2017) to our setting. The key hypothesis here is that meta-learning on a small number of heterogenous tasks leads to poor performance. To reflect these challenges, we construct a meta-learning problem with a dataset-based task distribution where each dataset consists of multiple reasoning categories (Fig. 2). Much like the original sine wave problem, the key challenge in adapting to a new reasoning category involves estimating the phase angle of the sine wave mapping from a small number of support set examples.

Our construction consists of multiple datasets. Each dataset is defined as a unit square sampled from a 10×10 grid over $x_1 = [-5, 5]$ and $x_2 = [-5, 5]$. Within each dataset, we construct multiple reasoning categories by defining each reasoning category to be a sine wave with a distinct phase. This is illustrated in Fig. 2 where each 1×1 represents a dataset, and sine waves along distinct rays correspond to reasoning categories. The target label y for the regression task is defined for each category by a randomly sampled phase $\phi \in [0.1, 2\pi]$ and $y = \sin(\|x - \lfloor x \rfloor\|_2 - \phi)$. At meta-training time, we sample a subset of these 100 squares as our training datasets, and then evaluate few shot adaptation to reasoning categories from held out datasets at meta-test time.



(a) (b)

Figure 3: Learning curves for MAML-Base (a) and MAML-Oracle (b). The lack of a gap between pre-adaptation (orange) and post-adaptation (blue) losses for MAML-Base indicates strong memorization overfitting. On the other hand, we see a big gap for MAML-Oracle which indicates that this model learns to adapt.

Experiments. We use similar hyperparameters as Finn et al. (2017) elaborated in Appendix A.1.

We start by considering MAML-Base, a meta-learner that is trained directly over a dataset-based task distribution. Concretely, we define each training task as a dataset and randomly sample episodes to train the meta-learner. Note that since episodes are drawn uniformly at random from an entire dataset, we expect support and query sets to often contain points from disjoint reasoning categories. In such scenarios, adaptation is not possible since the model cannot estimate the phase angle for query examples based on support examples. Thus, we expect pre and post adaptation losses to be similar. This is indeed reflected in the learning curves in Fig. 3(a). We observe that the orange and blue lines, corresponding to pre and post adaptation losses respectively, almost overlap. In other words, the model ignores the support set entirely. This is what we mean by *memorization overfitting*.

Next we consider MAML-Oracle, a meta-learner that is trained on tasks based on the underlying reasoning categories. In this setting, sup-

port and query sets are both drawn from the *same* sine wave, thus the model should be able to estimate phase angle for query examples from the support. This suggests that we should expect post adaptation loss to be lower than the pre adaptation loss. Empirically, from Fig. 3(b), we observe large gaps between pre and post adaptation losses which indicates that memorization overfitting has been mitigated. This leads us to the main question: Could we discover these reasoning categories?

Can we discover reasoning categories? In an attempt to discover these latent reasoning categories, we train a feedforward neural net (paramaterized similarly as h_θ) on the union of all the datasets, and use the final layer representation to cluster examples. We then use these clusters instead of the true reasoning categories to augment the original task distribution.

We now show learning curves on held out test tasks in Fig. 4. As expected MAML-Base fails to adapt to new reasoning categories, indicating that it was unable to acquire the required skill from its training tasks. On the other hand, MAML-Oracle is able to adapt very well, which confirms our hypothesis that a large number of high quality tasks helps. Finally, we see that using MAML trained on the augmented task distribution is able to match the performance of the oracle.

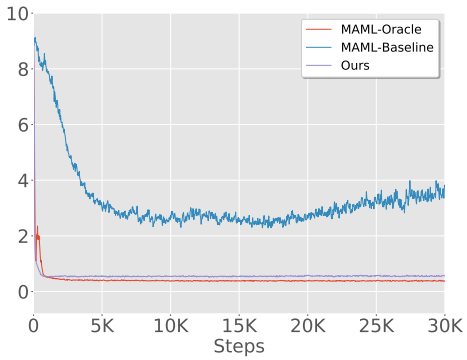


Figure 4: Results on the Toy sine-wave regression task. We observe that the oracle meta-learner outperforms the baseline, and our proposed approach is able to bridge the gap.

5 Our Approach

Experiments on the 2D sine wave regression problem confirm our hypothesis about the challenges of meta-learning with heterogenous task distributions. Since NLI datasets require a wide range of skills, we expect similar challenges on few-shot NLI as well. Motivated by the success of the clustering approach from Section 4, we now demonstrate that a similar procedure can extract reasoning categories for NLI. The key hypothesis here is that high quality sentence pair representations, such as those obtained from a fine-tuned BERT model, can bring out the micro-structure of NLI datasets.

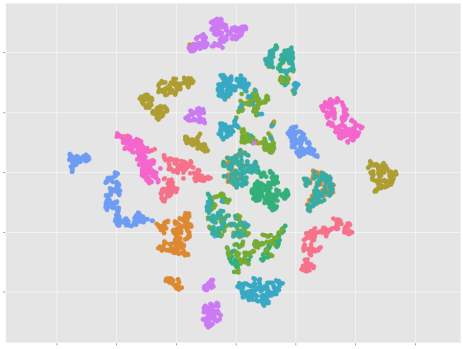


Figure 5: t-SNE plot of BERT vectors after fine-tuning on HANS. We see distinct clusters corresponding to the various reasoning categories.

We start by studying an analogue to our clustering approach for HANS (McCoy et al., 2019), a diagnostic NLI dataset. HANS consists of 30 manually defined syntactic templates which can be grouped into 15 reasoning categories. We fine-tune BERT (Devlin et al., 2019) for 5000 randomly chosen examples from HANS. To obtain a vector representation for each example $x = (p, h)$, we concatenate the vector at the [CLS] token, along with a mean pooled representation of the premise and hypothesis. We then use t-SNE (Maaten and Hinton, 2008) to project these representations onto 2 dimensions. Each point in Fig. 5 is colored with its corresponding reasoning category, and we can observe a clear clustering of examples according to their reasoning category. Indeed, the fact that pre-trained transformers can be used to create meaningful clusters has been shown in other recent works (c.f. Aharoni and Goldberg (2020); Joshi et al. (2020)). The ability of finetuned BERT representations to discover reasoning categories suggests our more general approach, which we describe below.

DReCa: The goal of DReCa is to take a heterogeneous task (such as a dataset) and produce a decomposed set of tasks. In doing so, we hope to obtain a large number of relatively homogeneous tasks that can be used to avoid meta overfitting.

Given a training task $\mathcal{T}_i^{\text{tr}}$, we first group examples by their labels, and then embed exam-

Model	HANS-fewshot	DNC-fewshot	CombinedNLI	GLUE-SciTail
Multitask	80.76 ± 1.83	70.27 ± 0.71	65.47 ± 3.19	75.80 ± 2.58
MAML-Base	82.64 ± 1.80	70.59 ± 1.17	72.61 ± 0.85	76.38 ± 1.25
MAML-DReCa	**87.53 ± 2.38**	**73.86 ± 1.28**	**75.36 ± 0.69**	**77.91 ± 1.60**
SMLMT (Bansal et al., 2020)	–	–	–	76.75 ± 2.08
MAML-Oracle	86.74 ± 1.06	72.06 ± 1.16	–	–

Table 2: Results on NLI Fewshot learning. We report the mean and 95% confidence intervals assuming accuracies follow a Gaussian. Bolded cells represent the best mean accuracy for the particular dataset. For all settings except GLUE-SciTail, we consider 2 way 1 shot adaptation. For GLUE-SciTail, we consider 2 way 4 shot adaptation.

Dataset	#Reasoning Categories	Cluster purity
HANS-fewshot	10	85.6%
DNC-fewshot	19	76.4%

Table 3: Measuring cluster purity. Our model is effective at recovering underlying reasoning types.

ples within each group with an embedding function $\texttt{EMBED}(.)$. Concretely, for each N-way classification task $\mathcal{T}_i^{\text{tr}}$ we form groups $g_l^i = \{(\texttt{EMBED}(x_i^p), y_i^p) \mid y_i^p = l\}$. Then, we proceed to refine each label group into K clusters via k-means clustering to break down $\mathcal{T}_i^{\text{tr}}$ into groups $\{C^j(g_l^i)\}_{j=1}^K$ for $l = 1, 2 \ldots N$.

These cluster groups can be used to produce K^N DReCa tasks. Each task is obtained by choosing one of K clusters for each of the N label groups, and taking their union. At meta-training time, learning episodes are sampled uniformly at random from DReCa tasks with a probability λ and from one of the original tasks with probability $1 - \lambda$. To produce learning episodes from DReCa tasks, we simply sample support and query sets from these augmented tasks.

Since our clustering procedure is based on fine-tuned BERT vectors, we expect the resulting clusters to roughly correspond to distinct reasoning categories. Indeed, when the true reasoning categories are known we show in Section 6.3 that DReCa yields clusters that recover these reasoning categories almost exactly.

6 NLI Experiments

6.1 Datasets

We evaluate DReCa on 4 NLI few-shot learning problems which we describe below.

HANS-fewshot is a few-shot classification problem over HANS (McCoy et al., 2019), a synthetic dataset for NLI. Each example in HANS comes from a hand-designed syntactic template which is associated with a fixed label (*entailment* or *not_entailment*). The entire dataset consists of 30 such templates which we use to define 15 reasoning categories. We then hold out 5 of these for evaluation, and train on the remaining 10. While this is a simple setting, it allows us to compare DReCa against an "oracle" with access to the underlying reasoning categories.

DNC-fewshot uses a subset of DNC (Poliak et al., 2018), a collection of multiple datasets recast as NLI. We manually write a collection of patterns representing a reasoning category, and match each example against these patterns. For each pattern, all examples that match this pattern form a task. This results in 25 distinct reasoning categories, out of which we hold out 8 tasks for evaluation.

CombinedNLI consists of a combination of 3 NLI datasets: MultiNLI (Williams et al., 2018), DNC and Semantic Fragments (Richardson et al., 2020) for training and RTE for evaluation. We convert both MultiNLI and Semantic Fragments to a 2-way classification by collapsing *contradiction* and *neutral* labels into a *not_entailment* label.

GLUE-SciTail where we train on all the NLI datasets from the GLUE benchmark (Wang et al., 2019) and evaluate on SciTail (Khot et al., 2018). This setting is comparable to Bansal et al. (2019) with the difference that we only meta-train on the NLI subset of GLUE, whereas Bansal et al. (2019) meta-train on all GLUE tasks. For GLUE-SciTail, we follow Bansal et al. (2019) and report 2-way 4-shot accuracy on SciTail.

6.2 Baselines

We compare our approach against several alternatives. **Multitask** is a non-episodic baseline that

Model	Accuracy
MAML-DReCa	87.53 ± 2.38
MAML-DReCa (No fine-tuning)	82.20 ± 2.25
MAML-DReCa (K = 5)	82.76 ± 2.07

Table 4: Ablations. We compare our full model against 2 variations.

trains h_θ on the union of all examples from each $\mathcal{T}_i^{\text{tr}}$, and then additionally fine-tunes the trained model separately on the support set of each test task. **MAML-Base** is a MAML model where every task corresponds to a dataset. When the true reasoning categories are known, we also compare with an oracle model **MAML-Oracle** which is trained over a mixture of dataset-based tasks as well as oracle reasoning categories. Finally, **MAML-DReCa** is our model which trains MAML over a mixture of the original dataset-based tasks as well as the augmented tasks from DReCa.

Evaluation. To control for variations across different support sets, we sample 5–10 random support sets for each test task. We fine-tune each of our models on these support sets and report means and 95% confidence intervals assuming the accuracies follow a Gaussian.

Training Details. For computational efficiency, we use first order MAML (FoMAML). We use BERT-base as the parameterization for h_θ. The inner loop optimization involves 10 gradient steps with Adam, with a support set of 2 examples (2-way 1-shot) for all except GLUE-SciTail where the support set size is 8 (2-way 4-shot). For DReCa, we use the fine-tuned BERT model to define $\texttt{EMBED}(;)$, similar to Section 5. The mixing weight λ is set to 0.5 for all our experiments.

Results. We find that DReCa improves model performance across all 4 datasets: MAML-DReCa improves over MAML-Base by +4.3 points on HANS-fewshot, +2.2 points on DNC-fewshot, +2.7 points on CombinedNLI and +1.6 points on GLUE-SciTail (Table 2). Moreover, we observe that MAML-DReCa is able to obtain comparable performance as MAML-Oracle (as confidence intervals overlap) on both HANS-fewshot and DNC-fewshot. On GLUE-SciTail, we also compare against the SMLMT model from (Bansal et al., 2020). We find that MAML-DReCa improves over this model by 1.5 accuracy points. However, we note that the confidence intervals of these approaches overlap, and also that (Bansal et al., 2019) consider the entire GLUE data to train the meta-learner whereas we only consider NLI datasets within GLUE.

6.3 Quantitative evaluation of clusters

To understand whether reasoning categories can be accurately recovered with our approach, we measure the purity of DReCa clusters when true reasoning categories are known. This is evaluated by first computing the number of examples belonging to the majority reasoning type for each cluster and then dividing by the total number of examples. Since this requires knowing oracle reasoning categories, we compute cluster purity for HANS-fewshot and DNC-fewshot. Results are in Table 3. We observe a very high cluster purity which provides some evidence that DReCa is able to recover true reasoning categories.

6.4 Model Ablations

We investigate the effects of number of clusters as well as the choice to use a fine-tuned BERT for clustering via ablation experiments. Our hypothesis is that fine-tuning representations is essential to bring out the micro-structure specific to our datasets, resulting in better augmented tasks. Moreover, a large number of augmented tasks would lead to improved performance due to less meta-overfitting. Results are in Table 4. We observe that MAML-Oracle (No fine-tuning) suffers a performance drop of 5.3 accuracy points suggesting that fine-tuning BERT is essential. Next, we see that decreasing the number of augmented tasks down to 25 ($K = 5$) from 400 also incurs a drop of 4.7 points compared to our full model.

7 Discussion

In this work, we take a closer look at using meta-learning tools for few-shot classification problems. One of the main ingredients for successful application of meta-learning is a large number of high quality training tasks to sample learning episodes for the meta-learner. We observe that such a task

distribution is usually not available for important NLP problems, leading to less desirable ad hoc alternatives that often treat entire datasets as tasks. In response, we propose DReCa as a simple and general purpose task augmentation strategy. From results on 4 NLI few-shot classification benchmarks, we conclude the effectiveness of our approach.

Many works suggest that there are fundamental challenges in creating systems that achieve human-like understanding of tasks like NLI. In this work, we studied conditions under which systems can learn with extremely few samples, and we believe that such systems would complement and enhance further study into more sophisticated challenges such as model extrapolation.

References

Roee Aharoni and Y. Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *ACL*.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2019. Learning to few-shot learn across diverse natural language classification tasks. ArXiv 1911.03863.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. ArXiv 2009.08445.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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

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

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1192–1197, Hong Kong, China. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Nithin Holla, Pushkar Mishra, H. Yannakoudakis, and Ekaterina Shutova. 2020. Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. *ArXiv*, abs/2004.14355.

Kyle Hsu, Sergey Levine, and Chelsea Finn. 2019. Unsupervised learning via meta-learning. In *International Conference on Learning Representations*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Pratik Joshi, S. Aditya, Aalok Sathe, and M. Choudhury. 2020. Taxinli: Taking a ride up the nlu hill. *ArXiv*, abs/2009.14505.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI 2018*.

L. V. D. Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of COLING*, pages 521–528, Manchester, UK.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting Diverse Natural Language Inference Problems for Sentence Representation Evaluation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Janarthanan Rajendran, Alex Irpan, and Eric Jang. 2020. Meta-learning requires meta-augmentation.

Kyle Richardson, H. Hu, L. Moss, and A. Sabharwal. 2020. Probing natural language inference models through semantic fragments. *ArXiv*, abs/1909.07521.

Adam Santoro, Sergey Bartunov, M. Botvinick, Daan Wierstra, and T. Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *ICML*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. 2020. Meta-learning without memorization. In *International Conference on Learning Representations*.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

A.1 2D Sine Wave Regression: Training Details

We use a two layer neural network with 40 dimensional hidden representations and ReLU nonlinearity as the parameterization of f. Following Finn et al. (2017), we take a single gradient step on the support set at meta-training time, and take 10 gradient steps at meta-test time. The MAML weights are optimized with Adam and the inner loop adaptation is done with SGD with a learning rate of 1e-2. For each outer loop update, we sample 5 tasks, and each episode consists of a support set of size 5 i.e. we consider 5 shot adaptation.