# Distilling Robustness into Natural Language Inference Models with Domain-Targeted Augmentation

**Joe Stacey**
Imperial College London
j.stacey20@imperial.ac.uk

**Marek Rei**
Imperial College London
marek.rei@imperial.ac.uk

## Abstract

Knowledge distillation optimises a smaller student model to behave similarly to a larger teacher model, retaining some of the performance benefits. While this method can improve results on in-distribution examples, it does not necessarily generalise to out-of-distribution (OOD) settings. We investigate two complementary methods for improving the robustness of the resulting student models on OOD domains. The first approach augments the distillation with generated unlabelled examples that match the target distribution. The second method upsamples data points among the training set that are similar to the target distribution. When applied on the task of natural language inference (NLI), our experiments on MNLI show that distillation with these modifications outperforms previous robustness solutions. We also find that these methods improve performance on OOD domains even beyond the target domain.[1]

## 1 Introduction

Large pre-trained language models can achieve impressive performance across a range of natural language understanding tasks (He et al., 2021; Touvron et al., 2023; Brown et al., 2020). However, as performance has increased, so has the number of model parameters (Zhao et al., 2023). While large models can be impractical for many applications, knowledge distillation can be used to reduce their size (Sanh et al., 2019; Xu and McAuley, 2022; Gou et al., 2021). During distillation, a smaller student model is trained to mimic the behaviour of a more complex teacher model on its training data, often improving the performance of the student model on in-distribution examples. However, this does not necessarily lead to robust improvements that generalise to out-of-distribution settings (Du et al., 2023; Rashid et al., 2021b; Li et al., 2021; Shao et al., 2021). We investigate two methods for improving out-of-distribution performance of the resulting student models: 1) Augmenting the distillation by generating new unlabelled task-specific examples that match the target distribution, and 2) Upsampling examples among the training data that are similar to the target distribution. We show that these two approaches are orthogonal and can be effectively combined together.

We use language models (LM) and a multi-step prompting process to generate additional unlabelled examples that target a particular task and domain. While labels for these examples can also be generated (Liu et al., 2022), our experiments show that the LM-generated labels are unreliable and lead to poor performance when used directly for supervised training of the student model. Instead, we use these examples to gather predicted probability distributions from the teacher models, then optimise the student models to predict similar distributions during distillation. This approach overcomes the issue of noisy labels and manages to considerably improve student model performance on out-of-distribution examples. In contrast to prior work on generating in-domain examples using the training set (Rashid et al., 2021b; Li et al., 2021; Tang et al., 2019; Haidar et al., 2022), this study is the first to investigate the use of generated data to target specific out-of-distribution domains.

While domain-targeted augmentation of the data improves performance in many settings, we found that it has little effect on minority examples[2]. Therefore, we investigate an additional method of upsampling minority examples during distillation, which substantially improves the performance of the student model on adversarial NLI test sets. Both of these methods can be combined together to improve robustness across a range of different NLI settings. While most distillation is performed with

[1] https://github.com/joestacey/robust_KD

[2] A term used to describe examples that counter common spurious patterns in a dataset (Tu et al., 2020)



*Findings of the Association for Computational Linguistics ACL 2024*, pages 2239–2258

a single teacher model, we also experiment with these methods by distilling from an ensemble of models. Ensembles can be used to better identify minority examples, while also increasing the general robustness of the teacher predictions. To the best of our knowledge, this is the first work to investigate model ensembles for better identification of minority examples.

We evaluate the distillation methods on the task of Natural Language Inference (NLI). In particular, we aim to improve the robustness of models trained on SNLI (Bowman et al., 2015) and evaluated on MNLI (Williams et al., 2018) (and vice versa) – a setting where prior work has consistently found negative results or limited improvements (Teney et al., 2020; Mahabadi et al., 2020; Belinkov et al., 2019a; Stacey et al., 2020, 2022a; Kumar and Talukdar, 2020; Zhao and Vydiswaran, 2021). We find that our simple but novel approach proves to be highly effective, combining the strengths of both LLMs and classification models.

## 2 Methods

Given either a large teacher model or an ensemble of teacher models, we aim to distil these models into a single student model that will perform well on different, out-of-distribution datasets, while also performing well on the in-distribution data used to train the teacher model. In the case of MNLI, our out-of-distribution data consists of multiple, different domains. By generating additional data for some of these domains (our target domains), we can measure how much performance improves on both the target domains and other out-of-distribution data comprised of different domains.

### 2.1 Knowledge Distillation

To maximise in-distribution performance, knowledge distillation often supervises a student model using a combination of the training labels and the soft predictions from a teacher model (Hahn and Choi, 2019; Du et al., 2023). We initially use the training labels, before the student model learns from the teacher model predictions for both the original training data and the augmented data. In effect, we are distilling one fine-tuned model into another fine-tuned model, which we find gives us the best performance. Similar to Li et al. (2021), we consider squared errors for our distillation loss:

$$Loss = \sum_{n=1}^{N}\sum_{c=1}^{C}(p_{n,c} - q_{n,c})^2 \quad (1)$$

where $p_{n,c}$ are the student predicted probabilities for the $c$-th class and $n$-th observation, and $q_{n,c}$ are the corresponding teacher predicted probabilities.

For labelled examples (i.e. not for our augmented data), we only include a distillation loss if either: 1) the teacher predictions are correct, or 2) the teacher model has a larger predicted probability for the correct class compared to the student model. We find that this further improves the performance of the knowledge distillation baseline.

We additionally consider robustness in a self-distillation setting, using the same model architecture for both the student and teacher models (Furlanello et al., 2018). In this case, we experiment with distilling from an ensemble of teacher models rather than using a single teacher model. We consider whether using an ensemble of teacher models improves robustness, and whether our proposed methods are still effective in this setting. In these cases, the ensemble distillation loss can be described as:

$$Loss = \sum_{n=1}^{N}\sum_{c=1}^{C}(p_{n,c} - \frac{1}{E}\sum_{i=1}^{E} q_{i,n,c})^2 \quad (2)$$

where $p_{n,c}$ are the predicted probabilities from the student model for class $c$ for the $n$-th observation. E represents the total number of models in our ensemble, with $q_{i,n,c}$ representing the predicted probabilities for the $c$-th class from the $i$-th teacher model on the $n$-th observation.

### 2.2 Generating Domain-Targeted Data

For our domain-targeted augmentation (DTA) method, we consider the MNLI genres as our target domains. Each of these domains is different from the single genre contained within SNLI. To improve performance on these out-of-distribution domains, we generate examples from a GPT-3 model (Brown et al., 2020) to mimic text that may appear in these genres. To ensure we are testing zero-shot performance and not few-shot performance, we do not provide our generator with any examples from the target genres. Instead, we provide the generator with a high-level description about the genre, and ask the model to generate a premise. For example, for the popular magazine article genre, we use the prompt: 'Provide a sentence from a popular magazine article'. We then generate corresponding hypotheses, asking the model to create a hypothesis for each class (see Appendix F for our full prompts). While the labels associated with

each generated example are unreliable, we use this approach to ensure that our generated examples are relatively balanced across the different classes. This method produces related sentence pairs, with a mixture of entailment, neutral and contradiction relationships, but without a reliable label that we can use during training.

The MNLI-matched and MNLI-mismatched validation sets each consist of five, distinct genres. We generate additional data for 4 of the 5 genres contained within the MNLI-matched validation set (we exclude the telephone transcripts genre). Then, as MNLI-mismatched consists of 5 genres that are not in MNLI-matched, we use MNLI-mismatched to test how well the new data augmentation helps models generalise to new, additional genres. In total, we generate 47,955 unlabelled sentence pairs for the 4 MNLI-matched genres.

### 2.3 Augmentation to Address the NLI Word-Overlap Heuristic

To examine whether using generated, unlabelled data during distillation can also help to address specific, known dataset biases, we also introduce a word-overlap augmentation (WOA). WOA is a variation of our DTA method that aims to overcome the NLI word-overlap bias (McCoy et al., 2019). The word-overlap bias is a heuristic where sentence pairs with a high overlap of words are more likely to be predicted as entailment. HANS measures model performance on this heuristic, containing examples where a high lexical overlap no longer correlates with the entailment class.

To generate the data, first we ask our generator to provide a short sentence, specifying a conjunction that must be included from a list of 60 conjunctions (ensuring variety in the linguistic structure of our premises). To prevent the model creating a second sentence very similar in meaning, the list of words is then shuffled with the conjunction removed (see Figure 3). We use the generator to exclude examples where both sentences have essentially the same meaning, or where the generator finds one of the sentences to be incoherent. In total, 4,695 additional examples were created. Similar to our domain-targeted augmentation, the augmented data contains both entailment and non-entailment examples, but no labels are provided.

### 2.4 Distilled Minority Upsampling

While our domain-targeted augmentation improves performance on different, unseen domains, it is unlikely to help with in-domain minority examples. We therefore introduce distilled minority upsampling (DMU) as a new method for improving model robustness, upsampling minority examples during knowledge distillation. We are motivated by creating complementary methods that can improve robustness on both minority examples and on different, out-of-distribution domains.

DMU is inspired by Just Train Twice (JTT) (Liu et al., 2021a), which Du et al. (2023) introduce to a student-teacher setting by identifying examples that a teacher model has misclassified and upsampling these examples when training the student model. Unlike JTT, our DMU method: 1) upsamples the minority examples during distillation rather than during fine-tuning, combining the benefits of both knowledge distillation and the additional supervision for minority examples, and 2) identifies minority examples as observations that the student model, rather than the teacher model, has misclassified. These changes result in a step-change in performance, with DMU substantially outperforming JTT on SNLI-hard (Gururangan et al., 2018). Additionally, we improve DMU by using an ensemble of models to identify the minority examples, defining minority examples as observations misclassified by any model in the ensemble.

A summary of how both DTA and DMU are applied together is provided in Figure 1.

## 3 Experiments

### 3.1 Distillation Setup

We experiment extensively with our new robustness methods across different distillation settings. While generative language models have shown excellent performance on a wide range of tasks, recent work has found that they still underperform on NLI compared to discriminative models (Chen et al., 2023; Wei et al., 2023). Therefore, we use generative models only for example generation and fine-tune pre-trained discriminative models for the NLI classification task. We evaluate the following combinations of teacher and student models: 1) a TinyBERT (Jiao et al., 2020) student model and a BERT (Devlin et al., 2019) teacher model, 2) a BERT student model and a BERT teacher model, 3) a DeBERTa (He et al., 2021) student model and a DeBERTa teacher model, 4) a BERT student model and a DeBERTa teacher model and 5) a TinyBERT student model and a DeBERTa teacher model. In settings 2 and 3 (using self-distillation), we exper-

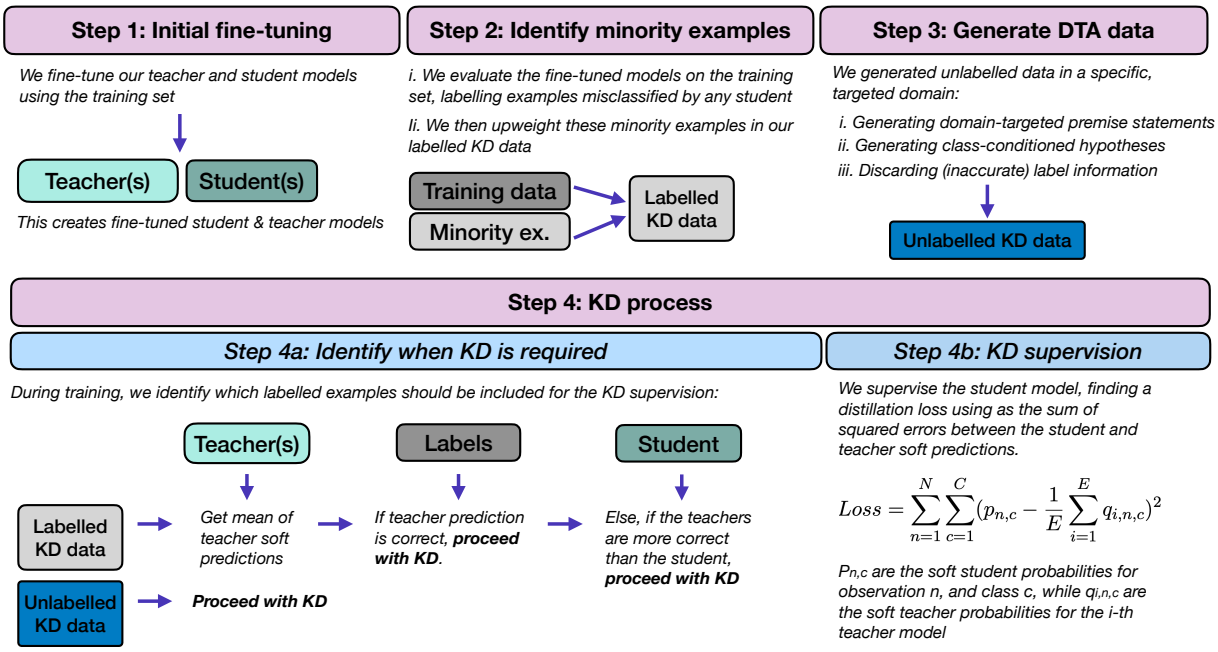


Figure 1: The full process for applying our DTA and DMU methods together. This diagram includes an explanation of how ensembles can be used in both DTA (with an ensemble of student models) and DMU (with an ensemble of student and/or teacher models).

iment with using an ensemble of teacher models. For each result, we report an average from 10 different seeds, performing significance testing[3] in each case.

We additionally experiment with applying DMU when distilling a RoBERTa-large (Liu et al., 2019) teacher model into a distil-RoBERTa student model, providing a comparison with previous work. This involves training on MNLI, and testing out-of-distribution performance on HANS. We additionally test our word-overlap augmentation method in this same setting.

### 3.2 Domain-Targeted Augmentation

When applying our domain-targeted augmentation (DTA) method, we primarily use SNLI as our in-distribution data and MNLI as our out-of-distribution data, using our generated examples for MNLI as unlabelled data during the distillation. We test out-of-distribution performance on both MNLI-matched and MNLI-mismatched, despite only generating data for the genres contained within MNLI-matched. We perform additional experimentation using MNLI as the training data and SNLI as the out-of-distribution data. The data is generated for SNLI using the same process as described for MNLI, with 47,898 unlabelled examples created for SNLI. To show the effect of our domain-targeted augmentation, we compare our results to standard knowledge distillation only using the training data. We also provide a baseline that uses the augmented data as labelled data during training, using the label that the hypotheses were conditioned over during the data generation. An additional smoothing baseline is also provided, as proposed by Du et al. (2023), which raises each class prediction by the teacher to the power of 0.9 before normalizing these scores.

### 3.3 Distilled Minority Upsampling

As DMU provides additional supervision to minority examples which counter common spurious correlations, this is likely to improve performance on minority examples rather than on other unseen out-of-distribution datasets. We therefore train our model on SNLI and evaluate performance on SNLI-hard (Gururangan et al., 2018), a test set that has been widely used to test robustness (Mahabadi et al., 2020; Belinkov et al., 2019a,b; Sanh et al., 2021). Additionally, we experiment with using an ensemble of models to better identify the minority examples, before performing the distillation using an ensemble of teacher models. We also compare DMU with a JTT baseline (Du et al., 2023).

[3]We use two-tailed bootstrapping hypothesis testing (Efron and Tibshirani, 1993) to test statistical significance.

| Model | In-Distribution | | Out-of-Distribution | | |
|---|---|---|---|---|---|
| | SNLI-dev | SNLI-test | SNLI-hard | MNLI-mm | MNLI-m |
| | *BERT -> TinyBERT:* | | | | |
| BERT teacher | 91.03 | 90.59 | 80.31 | 75.01 | 74.97 |
| TinyBERT baseline | 77.99 | 78.25 | 56.98 | 55.50 | 54.24 |
| Baseline w/ labelled aug. data | 77.72 | 78.18 | 56.73 | 46.47 | 45.55 |
| JTT[1,2] | 76.96 | 76.25 | 55.93 | 52.66 | 52.00 |
| KD (standard distillation) | 80.11 | 80.34 | 60.02 | 57.69 | 55.82 |
| KD + Smoothing[1] | 80.09 | 80.33 | 60.07 | 57.71 | 55.83 |
| *Ours:* | | | | | |
| DMU | 80.00 | 80.25↓ | 65.94↑ | 55.86↓ | 54.08↓ |
| DTA | **80.16** | **80.51**↑ | 60.26↑ | **59.94**↑ | **57.17**↑ |
| DTA with DMU | 80.11 | 80.43↑ | **66.04**↑ | 59.01↑ | 56.54↑ |

Table 1: Accuracy of a TinyBERT model (4.4 million parameters), compared to a BERT model (110 million parameters) distilled into a TinyBERT model. We compare performance of standard knowledge distillation to our approach using domain-targeted data augmentation and our DMU approach. We also compare with KD with smoothing (Du et al., 2023)[1] and a JTT baseline (Liu et al., 2021a; Du et al., 2023)[1,2]. MNLI-m and MNLI-mm refer to the MNLI matched and mismatched validation sets respectively. All distillation and DMU results are an average from 10 seeds. ↑ and ↓ represent statistically significant results ($p < 0.05$), with all p-values displayed in Table 12. The best results are in bold.

## 4 Results

### 4.1 Domain-Targeted Augmentation

Our domain-targeted augmentation (DTA) significantly improves performance on the out-of-distribution MNLI-matched dataset for every condition we tested. In the case where a BERT model is distilled into a TinyBERT model, out-of-distribution performance on MNLI-matched is +1.35% higher when compared to applying a knowledge distillation baseline (Table 1). When distilling a DeBERTa teacher model into either a BERT or TinyBERT student model, we see improvements of 1.8% and 1.61% percentage points respectively (see Table 6 and Table 8).

Not only does the augmented data improve performance on MNLI-matched in the targeted domains, but also on MNLI-mismatched, which consists of different domains. We observe statistically significant improvements on MNLI-mismatched for every combination of teacher and student models that we tested (see Table 1, Table 2, Table 6 and Table 8), showing that our method can also improve performance on domains that were not included in the augmented data. While not the focus of our work, we also observe a very small but statistically significant improvement on the in-distribution SNLI-test set in each case. Finally, we see similar improvements when training on MNLI and testing on SNLI (see Table 7).

If the augmented data is used as labelled data (using the label the generated hypothesis was conditioned on), out-of-distribution performance on MNLI is substantially worse than the baseline. Interestingly however, the inclusion of this labelled data has little impact in-distribution.

### 4.2 Distilled Minority Upsampling

For each condition we tested, DMU significantly improves performance on the SNLI-hard test set (see Table 1, Table 6 and Table 8). When using a BERT teacher and TinyBERT student model, this improvement is substantial, with 5.92% higher accuracy than the knowledge distillation baseline (Table 1). This improvement on SNLI-hard contrasts with the results from DTA, which does not substantially improve performance on these minority examples.

While previous work using JTT in a teacher-student setting uses the teacher model to identify minority examples (Du et al., 2023), we find better performance when using the student model to identify minority examples (see Appendix Table 11 for results using teacher-identified minority examples).

| | Out-of-Distribution | |
|---|---|---|
| | MNLI-mm | MNLI-m |
| | *BERT -> BERT:* | |
| BERT baseline | 75.01 | 74.97 |
| KD | 75.42 | 75.50 |
| DTA (Ours) | 75.77↑ | 75.86↑ |
| $KD_{ens}$ (Ours) | 75.90 | 75.98 |
| $DTA_{ens}$ (Ours) | **76.42**↑ | **76.45**↑ |
| | *DeBERTa -> DeBERTa:* | |
| DeBERTa baseline | 84.78 | 84.56 |
| KD | 85.24 | 84.83 |
| DTA (Ours) | 85.68↑ | 85.20↑ |
| $KD_{ens}$ (Ours) | 85.52 | 85.29 |
| $DTA_{ens}$ (Ours) | **86.18**↑ | **85.77**↑ |

Table 2: DTA is tested in a self-distillation setting, using either a single teacher or an ensemble of teachers. All distillation results are an average from 10 seeds. ↑ and ↓ represent statistically significant results ($p < 0.05$), testing the significance of DTA compared to standard knowledge distillation. The best results are in bold.

| Method | Test | SNLI-Hard | Δ |
|---|---|---|---|
| Baseline | 78.25 | 56.98 | |
| JTT | 76.25 | 55.93 | -1.05 |
| KD | 80.34 | 60.02 | +3.04 |
| DMU | 80.25↓ | 65.94↑ | +8.96 |
| $DMU_{up}$ | 80.88↑ | 66.42↑ | +9.44 |
| $DMU_{full}$ | **81.01**↑ | **66.48**↑ | +9.50 |

Table 3: Performance of JTT (Liu et al., 2021a; Du et al., 2023) compared DMU using a TinyBERT student and BERT teacher model. $DMU_{up}$ uses a single teacher model but upsamples examples that any model in an ensemble incorrectly predicted, while $DMU_{full}$ also uses an ensemble of teachers during distillation. All DMU results are an average from 10 seeds. ↑ and ↓ represent results that are statistically significant results ($p < 0.05$).

DMU and DTA are complementary, and when applying both methods we see statistically significant improvements on MNLI-matched, MNLI-mismatched and SNLI-hard in every condition tested (see Table 1, Table 6 and Table 8). While including DMU with DTA can reduce performance on MNLI compared to only using DTA (see Table 1), there are corresponding substantial improvements on SNLI-hard. On the other hand, including DTA with DMU mitigates some of the limitations of using DMU, which can otherwise have reduced performance on MNLI-matched and MNLI-mismatched, or reduced performance in-distribution (see Table 1 and Table 8).

### 4.3 Distilling from an Ensemble of Teacher Models

Performing distillation with an ensemble of teacher models has significantly better performance, both in-distribution and out-of-distribution, compared to distillation with a single teacher model. This is the case for both our BERT and DeBERTa models (see Table 9). Additionally, we find that our domain-targeted data augmentation significantly improves performance when combined with an ensemble of teacher models. This is the case when using either a BERT or DeBERTa model as the student and teacher (see Table 2 and Table 10). While these improvements are small, in the case of BERT including our augmented data with the ensemble improves performance on MNLI matched and mismatched by 58% and 47% relative to the baseline.

We also find that we can improve DMU by using an ensemble to identify minority examples, upsampling examples that have been incorrectly predicted by any model in an ensemble ($DMU_{up}$ in Table 3). In this case, $DMU_{up}$ improves performance by 9.44% compared to the baseline student model. This ensemble consists of the student model, and 7 other models consisting of the same architecture. Additionally, instead of using a single teacher model ($DMU_{up}$), an ensemble of teacher models can also be used during the distillation ($DMU_{full}$ in Table 3), slightly improving performance.

### 4.4 Improving Robustness against the Word-Overlap Heuristic

We find that our word-overlap augmentation (WOA) improves performance on the adversarial HANS dataset after training on MNLI (Table 4), although without setting a new state-of-the-art result. Previous work augments the training data with a large number of training examples (392,702, the same size as the MNLI training set) (Li et al., 2021), whereas we only augment the data with 4,695 examples which we upsample (x10). When we include our small number of domain-targeted observations, we achieve more than half the out-of-distribution improvements compared to the prior work while

| Method | MNLI-m | HANS | #aug |
|---|---|---|---|
| *Teacher and student models* | | | |
| RoBERTa-large | 89.6 | 76.6 | |
| DistilRoBERTa | 83.8 | 59.9 | |
| *Without augmentation* | | | |
| Annealing-KD[1] | **84.5** | 61.2 | - |
| KD | 84.1 | 61.8 | |
| $DMU_{full}$ | 84.2 | **65.9** | - |
| *With augmentation* | | | |
| ComKD[2] | **87.2** | **68.6** | 393k |
| $WOA_{ens}$ | 84.3 | 65.1 | 5k |
| $WOA^{**}{}_{ens}$ | 81.6 | 68.3 | 5k |

Table 4: Accuracy is displayed on MNLI-matched (in-distribution), and HANS (out-of-distribution). ** refers to the setting where we only perform the distillation step on the augmented data. We compare our results to previous sota results improving robustness for knowledge distillation: [1] Jafari et al. (2021), and [2] Li et al. (2021). The best results are in bold.

only using a fraction (1.2%) of the additional examples. Directly applying our DMU method proves to be highly effective in this setting, outperforming previous work without any data augmentation.

### 4.5 Comparison to Previous OOD-Performance on MNLI

Improving performance out-of-distribution on MNLI after training on SNLI remains a challenging task. Despite extensive prior work evaluating models in this condition, few approaches yield out-of-distribution improvements. We compare this prior work to our own results using self-distillation with BERT and DeBERTa. We find that distillation using both our domain-targeted augmentation and using an ensemble of teachers outperforms all previous work (Table 5). While adversarial training using a single hypothesis-only adversary (Belinkov et al., 2019a) produced larger improvements, this work involved hyper-parameter tuning on MNLI-mismatched for a model evaluated on MNLI-matched, and vice versa. On the other hand, our experiments also do not assume the availability of any MNLI examples to use as a validation set. Previous related work includes debiasing techniques that aim to improve zero-shot performance (Belinkov et al., 2019a; Stacey et al., 2020; Mahabadi et al., 2020; Teney et al., 2020), in addition to previous work incorporating human explanations when training (Zhao and Vydiswaran, 2021; Kumar and Talukdar, 2020; Stacey et al., 2022a).

## 5 Related Work

### 5.1 Improving Robustness in Knowledge Distillation

To improve robustness in knowledge distillation, previous methods have involved smoothing the teacher predictions (Du et al., 2023; Jafari et al., 2021), or using additional unlabelled training data during the distillation (Rashid et al., 2021b; Li et al., 2021). The smoothing methods either smooth the teacher model predictions more at the beginning of training (Jafari et al., 2021), or based on the difficulty of each example (Du et al., 2023). We find that our results outperform a smoothing baseline proposed by Du et al. (2023).

Most similar to our approach of using additional, unlabelled data during distillation, Rashid et al. (2021b); Li et al. (2021); Haidar et al. (2022) augment their model with additional training examples that are created by perturbing existing observations. This involves randomly masking words, before replacing these words using a generator that is trained to maximise the difference between the student and teacher predicted probabilities (Rashid et al., 2021b; Li et al., 2021; Haidar et al., 2022) and their intermediate representations (Haidar et al., 2022). Further work involves perturbing existing examples without an adversarial objective (Tang et al., 2019; Jiao et al., 2020), and perturbing additional language data not related to NLI into additional NLI examples (Rashid et al., 2021a). Previous work measures adversarial robustness using HANS (Li et al., 2021; Rashid et al., 2021b; Du et al., 2023; Haidar et al., 2022). We compare our DMU and WOA methods to Li et al. (2021) and Jafari et al. (2021), the state-of-the-art results for robust knowledge distillation on HANS with and without additional data augmentation.

### 5.2 Upsampling Minority Examples

Tu et al. (2020) introduce the term minority examples to describe instances which counter the spurious correlations present in a dataset. Upsampling these minority examples during training has been shown to improve model robustness (Liu et al., 2021a; Yaghoobzadeh et al., 2021; Du et al., 2023). While Liu et al. (2021a) identify minority examples as training examples that are misclassified

| Method | Baseline | MNLI-mm Acc | MNLI-mm Imp | MNLI-m Acc | MNLI-m Imp | MNLI-all Acc | MNLI-all Imp |
|---|---|---|---|---|---|---|---|
| *Hyper-parameter tuning on MNLI* | | | | | | | |
| Negative sampling[1] | LSTM | 43.66 | -3.91 | 43.76 | -2.10 | 43.71 | -3.01 |
| Hyp-only adversary[1] | LSTM | 49.24 | **+1.67** | 47.24 | **+1.38** | 48.24 | **+1.52** |
| Ensemble-adversaries[2] | LSTM | 52.81 | -0.10 | 54.18 | +0.80 | 53.49 | +0.35 |
| Product of Experts[3] | BERT | 73.49 | -0.49 | 73.61 | -0.79 | 73.55 | -0.64 |
| Debiased Focal Loss[3] | BERT | 74.00 | +0.02 | 73.58 | -0.82 | 73.79 | -0.40 |
| *No hyper-parameter tuning on MNLI* | | | | | | | |
| Rationale supervision[4] | BERT | 73.36 | +0.84 | 73.19 | +0.91 | 73.28 | +0.87 |
| KD | BERT | 75.50 | +0.53 | 75.42 | +0.41 | 75.46 | +0.47 |
| NILE[5] | RoB. | 77.22 | -2.07 | 77.07 | -2.22 | 77.15 | -2.14 |
| LIREx[6] | RoB. | 79.79 | +0.06 | 79.85 | -0.27 | 79.82 | -0.10 |
| KD | DeB. | 84.83 | +0.27 | 85.24 | +0.46 | 85.04 | +0.37 |
| *Ours:* | | | | | | | |
| $KD_{ens}$+aug | BERT | 76.42 | **+1.41** | 76.45 | **+1.48** | 76.43 | **+1.44** |
| $KD_{ens}$+aug | DeB. | **86.18** | +1.40 | **85.77** | +1.21 | **85.98** | +1.31 |

Table 5: A comparison of work testing zero-shot performance on the MNLI matched and mismatched sets after training on SNLI (MNLI-all combines both validation sets). Performance of each method is compared to their respective baselines to show when further out-of-distribution improvements are achieved. RoB. stands for RoBERTa, while DeB stands for DeBERTa. [1] Belinkov et al. (2019a), [2] Stacey et al. (2020), [3] (Mahabadi et al., 2020), [4] Stacey et al. (2022a), [5] Kumar and Talukdar (2020), [6] Zhao and Vydiswaran (2021). Methods 4, 5 & 6 include the use of human annotated rationales (Camburu et al., 2018).

by a model trained on that data, Yaghoobzadeh et al. (2021) additionally consider instances that have been properly classified at some point during training, but are then misclassified later in training. Rather than upsampling minority examples, Korakakis and Vlachos (2023) introduce a minimax objective to improve robustness, upweighting the loss of examples during training to maximise the training loss. Du et al. (2023) adapt these ideas to a student-teacher setting, upsampling training examples that a teacher model has misclassified when training a student model. We directly compare our DMU method to this approach, showing substantial improvements in robustness.

### 5.3 Language Model Data Augmentation for Knowledge Distillation

Using language models to generate additional data has previously shown promising results in a data-free setting. Ma et al. (2022) generate synthetic examples based on the topics present in the training data, using the generated data to perform knowledge distillation in a data-free setting.

Alternatively, without being provided with a more specific prompt, language models can be fine tuned on the training data to generate additional training examples that can be used with the distillation (He et al., 2022b). Language models can also be used to modify spans in NLI examples, creating new counterfactual examples (Chen et al., 2022), before an NLI model decides whether the perturbed counterfactual examples have the desired class. Rather than creating counterfactual training data, or perturbing existing training examples, we generate data that specifically targets new, additional domains.

### 5.4 Knowledge Distillation with Ensembles

Knowledge distillation is most commonly applied to distil a single, more complex teacher model into a smaller student model with fewer parameters (He et al., 2022a; Salmony and Faridi, 2022; Gou et al., 2021). However, ensembles of models can also be distilled into a single model (Hinton et al., 2015; Asif et al., 2020; Freitag et al., 2017). We find that the in-distribution improvements from using an ensemble of teacher models are accompanied by out-of-distribution improvements. Moreover,

using these ensembles are complementary to our domain-targeted data augmentation method.

## 6 Conclusion

We introduce domain-targeted augmentation and DMU as two methods to improve out-of-distribution robustness in NLI. In the case of the domain-targeted augmentation, using the additional, generated examples during knowledge distillation proves to be a highly effective technique, outperforming all previous work that measures out-of-distribution robustness on MNLI. Not only do we find that performance is better on the targeted domains, but performance is also better for domains that were not included in the augmented data. We also find that our DMU method produces substantial improvements on SNLI-hard, helping the student model to make better predictions for minority examples. Using ensembles can help both methods, improving how minority examples are identified for DMU, and improving the teacher distributions for our domain-targeted augmentation.

We also find that our WOA method can improve robustness on HANS, showing that using unlabelled data during distillation can also target specific, known dataset biases.

## Limitations

The main limitation of our domain-targeted data augmentation is the cost of generating the unlabelled examples using GPT3 (approximately 100USD for the experiments provided). As a result, while we perform extensive experimentation on NLI datasets (with over 200 experiments), we do not also apply this method to other NLP tasks. We choose to focus on NLI, as many previous works on robustness evaluate on this task. Our experimentation is also limited to single sentence NLI datasets such as MNLI, SNLI and HANS, and therefore the findings may not necessarily generalise to NLI datasets with longer hypotheses and premises such as ANLI (Nie et al., 2020) or ConTRoL (Liu et al., 2021b).

In this work we show that including the domain-targeted augmentation benefits other domains that the data was not generated for. We demonstrate this by creating data to mimic the domains within MNLI-matched, before testing performance on MNLI-mismatched which contains a different set of domains. However, as there are similarities between examples in MNLI matched and mismatched, further work could test the extent that these benefits generalise to different tasks or domains.

Additionally, in Table 5, we provide all results known to us from methods that train on SNLI and test zero-shot performance on MNLI. While this previous work contains a variety of methods, including different debiasing techniques, not all NLI debiasing methods have been evaluated in this setting.

Finally, for our DTA method it is possible that our GPT-3 generator model has seen data from the target domain during pretraining. However, on inspection, the unlabelled examples generated by GPT-3 did not closely resemble the data in MNLI. This is likely because we generated the data in several stages, first asking the model for an 'example extract from [target domain]' to create a premise statement. This prompt refers to a broad topic and is very unlikely to result in GPT-3 generating premises specifically from MNLI. The second sentence (the hypothesis) is generated to be relevant specifically to a given premise - if the premise is not from MNLI then the hypothesis would not be either.

## Ethics Statement

The data generation process in this paper did not involve any human annotation, and the generated data does not contain personal information. All data is generated using GPT-3 (please see Appendix E for further information).

## Acknowledgements

We would like to thank He He for all her valuable feedback on this work. Joe Stacey was supported by the Apple Scholars in AI/ML PhD fellowship.

## References

Umar Asif, Jianbin Tang, and Stefan Harrer. 2020. Ensemble knowledge distillation for learning improved and efficient networks. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 953–960. IOS Press.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019a.

Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019b. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 256–262, Minneapolis, Minnesota. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Xuanting Chen, Junjie Ye, Can Zu, Nuo Xu, Rui Zheng, Minlong Peng, Jie Zhou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. How robust is GPT-3.5 to predecessors? A comprehensive study on language understanding tasks. *CoRR*, abs/2303.00293.

Zeming Chen, Qiyue Gao, Kyle Richardson, Antoine Bosselut, and Ashish Sabharwal. 2022. DISCO: distilling phrasal counterfactuals with large language models. *CoRR*, abs/2212.10534.

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.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. 2023. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1766–1778, Dubrovnik, Croatia. Association for Computational Linguistics.

Bradley Efron and R Tibshirani. 1993. An introduction to the bootstrap.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

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.

Sangchul Hahn and Heeyoul Choi. 2019. Self-knowledge distillation in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 423–430, Varna, Bulgaria. INCOMA Ltd.

Md. Akmal Haidar, Mehdi Rezagholizadeh, Abbas Ghaddar, Khalil Bibi, Philippe Langlais, and Pascal Poupart. 2022. CILDA: contrastive data augmentation using intermediate layer knowledge distillation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 4707–4713. International Committee on Computational Linguistics.

Haoyu He, Xingjian Shi, Jonas Mueller, Sheng Zha, Mu Li, and George Karypis. 2022a. Towards automated distillation: A systematic study of knowledge distillation in natural language processing. In *AutoML Conference 2022*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2022b. Generate, annotate, and learn: NLP with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. 2021. Annealing knowledge distillation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2493–2504. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Michalis Korakakis and Andreas Vlachos. 2023. Improving the robustness of NLI models with minimax training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14322–14339, Toronto, Canada. Association for Computational Linguistics.

Sawan Kumar and Partha Talukdar. 2020. NILE : Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, Online. Association for Computational Linguistics.

Tianda Li, Ahmad Rashid, Aref Jafari, Pranav Sharma, Ali Ghodsi, and Mehdi Rezagholizadeh. 2021. How to select one among all ? an empirical study towards the robustness of knowledge distillation in natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 750–762, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021a. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR.

Hanmeng Liu, Leyang Cui, Jian Liu, and Yue Zhang. 2021b. Natural language inference in context - investigating contextual reasoning over long texts. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13388–13396. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xinyin Ma, Xinchao Wang, Gongfan Fang, Yongliang Shen, and Weiming Lu. 2022. Prompting to distill: Boosting data-free knowledge distillation via reinforced prompt. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4296–4302. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2021. Variational information bottleneck for effective low-resource fine-tuning. In *International Conference on Learning Representations*.

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.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Ahmad Rashid, Vasileios Lioutas, Abbas Ghaddar, and Mehdi Rezagholizadeh. 2021a. Towards zero-shot knowledge distillation for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,*

*EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6551–6561. Association for Computational Linguistics.

Ahmad Rashid, Vasileios Lioutas, and Mehdi Rezagholizadeh. 2021b. MATE-KD: Masked adversarial TExt, a companion to knowledge distillation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1062–1071, Online. Association for Computational Linguistics.

Monir Yahya Ali Salmony and Arman Rasool Faridi. 2022. Bert distillation to enhance the performance of machine learning models for sentiment analysis on movie review data. In *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 400–405. IEEE.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, et al. 2021. The multiberts: Bert reproductions for robustness analysis. *arXiv preprint arXiv:2106.16163*.

Rulin Shao, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. 2021. How and when adversarial robustness transfers in knowledge distillation? *CoRR*, abs/2110.12072.

Joe Stacey, Yonatan Belinkov, and Marek Rei. 2022a. Supervising model attention with human explanations for robust natural language inference. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11349–11357. AAAI Press.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Oana-Maria Camburu, and Marek Rei. 2023. Logical reasoning for natural language inference using generated facts as atoms. *CoRR*, abs/2305.13214.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, and Marek Rei. 2022b. Logical reasoning with span-level predictions for interpretable and robust NLI models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3809–3823. Association for Computational Linguistics.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. 2020. Avoiding the Hypothesis-Only Bias in Natural Language Inference via Ensemble Adversarial Training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8281–8291, Online. Association for Computational Linguistics.

Raphael Tang, Yao Lu, and Jimmy Lin. 2019. Natural language generation for effective knowledge distillation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 202–208, Hong Kong, China. Association for Computational Linguistics.

Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. 2020. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part X*, volume 12355 of *Lecture Notes in Computer Science*, pages 580–599. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Canwen Xu and Julian J. McAuley. 2022. A survey on model compression for natural language processing. *CoRR*, abs/2202.07105.

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet, T. J. Hazen, and Alessandro Sordoni. 2021. Increasing robustness to spurious correlations using forgettable examples.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Xinyan Zhao and V. G. Vinod Vydiswaran. 2021. Lirex: Augmenting language inference with relevant explanations. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Artificial Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14532–14539. AAAI Press.

## A Additional results

We provide additional results from training the student and teacher models on MNLI, and testing zero-shot performance on SNLI (Table 7). For these experiments, we use a BERT teacher model and TinyBERT student model. We find that these results mirror the results from training on SNLI and testing on MNLI (see Table 1), with 2.01% and 1.96% point improvements on the validation and test sets compared to applying knowledge distillation without the unlabelled augmented data. For these experiments, we use the mismatched validation set as our validation set for early stopping.

We also provide experimental results when using a DeBERTa model as a teacher model, and either a BERT model or a TinyBERT model as the student (see Table 6 and Table 8). In these settings, we also see improvements from our domain-targeted augmentation, with improvements of 1.84% on MNLI-mismatched, and improvements of 1.8% on MNLI-matched for a BERT student model, compared to improvements of 2.4% and 1.61% with a TinyBERT student model. When applying our DMU method with a DeBERTa teacher model, we also see statistically significant improvements on SNLI-hard (see Table 6 and Table 8).

Additionally, we test our DMU method when using the teacher models to identify minority examples instead of using the student models. In this case, we use either a single teacher model, or an ensemble of teacher models to identify these examples. Our results show worse performance when using a teacher model to identify minority examples, compared to when using a student model (see Table 11).

## B Performance of LLMs on NLI

There is currently a lack of evidence that LLMs outperform other transformer-based classification models on NLI, especially considering the number of model parameters. While LLMs have shown increasingly impressive performance across a range of different tasks, this is not the case with NLI. Chen et al. (2023) find significant robustness degradation on NLI when using GPT-3.5-turbo, despite finding better performance on other tasks. This work includes an evaluation of GPT-3.5-turbo on both SNLI and MNLI (Chen et al., 2023).

Similar findings have been found across other NLI datasets. For example, Wei et al. (2023) apply a Chain-of-Thought GPT-3.5-turbo model, with performance substantially below previous work using a DeBERTa-v3 baseline (Stacey et al., 2023).

## C Robustness in NLI

Improving model robustness for Natural Language Inference (NLI) is a well studied area, where robustness is measured either by testing performance on adversarial datasets such as HANS or the NLI stress tests (Naik et al., 2018). Alternatively, robustness is measured using unseen, out-of-distribution test sets such MNLI (Williams et al., 2018), a challenging robustness setting where existing debiasing methods often do not improve performance (Belinkov et al., 2019a; Mahabadi et al., 2020). There has been some success improving out-of-distribution performance on MNLI (Stacey et al., 2022a; Teney et al., 2020; Belinkov et al., 2019a), particularly in a reduced-data setting (Stacey et al., 2022b; Mahabadi et al., 2021), however, most methods do not lead to any improvements (Zhao and Vydiswaran, 2021; Kumar and Talukdar, 2020; Camburu et al., 2018; Belinkov et al., 2019a; Mahabadi et al., 2020).

## D Model Parameters

Our DeBERTa model consists of 184 million parameters, compared to 110 million parameters for BERT and 4.4 million parameters for tinyBERT. When distilling RoBERTa, our RoBERTa model consists of 355 million parameters, compared to 83 million for distil-RoBERTa. Over 200 experiments are conducted, consisting of approximately 2500 GPU hours using RTX6000 GPUs.

| Model | In-Distribution | | Out-of-Distribution | | |
|---|---|---|---|---|---|
| | SNLI-dev | SNLI-test | SNLI-hard | MNLI-mm | MNLI-m |
| | *DeBERTa -> BERT:* | | | | |
| DeBERTa teacher | 93.32 | 92.53 | 84.64 | 84.78 | 84.56 |
| BERT baseline | 91.03 | 90.59 | 80.31 | 75.01 | 74.97 |
| KD | 91.66 | 91.04 | 81.26 | 75.21 | 75.61 |
| DTA (Ours) | 91.77↑ | 91.14↑ | 81.42 | **77.05**↑ | **77.41**↑ |
| DMU (Ours) | 91.77↑ | **91.17**↑ | 81.62↑ | 75.05 | 75.55 |
| DTA with DMU (Ours) | **91.84**↑ | 91.16↑ | **81.64**↑ | 76.72↑ | **77.41**↑ |

Table 6: Our domain-targeted augmentation method is compared to a knowledge distillation baseline. These experiments use a DeBERTa teacher model and BERT student model. Results use one random seed. All distillation and DMU results show the accuracy from an average of 10 random seeds. ↑ and ↓ represent results that are statistically significant with $p < 0.05$. The best results are in bold.

| Model | In-Distribution | | Out-of-Distribution | | |
|---|---|---|---|---|---|
| | MNLI-mm | MNLI-m | SNLI-dev | SNLI-test | SNLI-hard |
| | *BERT -> TinyBERT:* | | | | |
| BERT teacher | 84.64 | 84.38 | 79.31 | 80.09 | 71.3 |
| TinyBERT baseline | 65.62 | 63.89 | 52.77 | 52.59 | 42.90 |
| KD | 68.23 | 66.72 | 55.78 | 55.90 | 46.15 |
| DTA (Ours) | **68.45**↑ | **66.97**↑ | **57.79**↑ | **57.86**↑ | **46.36** |

Table 7: Evaluating our domain-targeted augmentation when training on MNLI and testing zero-shot performance on SNLI. Performance is compared to knowledge distillation without the augmented data, and also a TinyBERT baseline. MNLI-mismatched is our validation set. As SNLI-hard specifically considers minorty examples for models trained on SNLI, we do not also test DMU in this setting. All distillation results show the accuracy from an average of 10 random seeds. ↑ and ↓ represent results that are statistically significant with $p < 0.05$. The best results are in bold.

| Model | In-Distribution | | Out-of-Distribution | | |
|---|---|---|---|---|---|
| | SNLI-dev | SNLI-test | SNLI-hard | MNLI-mm | MNLI-m |
| | *DeBERTa -> TinyBERT:* | | | | |
| DeBERTa teacher | 93.32 | 92.53 | 84.64 | 84.78 | 84.56 |
| TinyBERT baseline | 77.99 | 78.25 | 56.98 | 55.50 | 54.24 |
| KD | 80.00 | 80.24 | 59.85 | 57.67 | 55.90 |
| DTA (Ours) | **80.02** | **80.40**↑ | 60.11↑ | **60.07**↑ | **57.51**↑ |
| DMU (Ours) | 79.26↓ | 79.41↓ | 65.39↑ | 54.45↓ | 53.00↓ |
| DTA with DMU (Ours) | 79.33↓ | 79.65↓ | **65.44**↑ | 58.94↑ | 56.51↑ |

Table 8: Accuracy of a TinyBERT model, compared to a DeBERTa model distilled into a TinyBERT model. We compare performance of standard knowledge distillation to our approach using domain-targeted data augmentation. The best results are in bold. All distillation and DMU results show the accuracy from an average of 10 random seeds. ↑ and ↓ represent results that are statistically significant with $p < 0.05$. No early stopping was included for our DMU experiments. The best results are in bold.

| Model | In-Distribution | | Out-of-Distribution | | |
|---|---|---|---|---|---|
| | SNLI-dev | SNLI-test | SNLI-hard | MNLI-mm | MNLI-m |
| | | | *BERT -> BERT:* | | |
| BERT baseline | 91.03 | 90.59 | 80.31 | 75.01 | 74.97 |
| KD | 91.43 | 90.72 | 80.57 | 75.42 | 75.50 |
| $KD_{ens}$ (Ours) | **91.59**↑ | **90.94**↑ | **80.81**↑ | **75.90**↑ | **75.98**↑ |
| | | | *DeBERTa -> DeBERTa:* | | |
| DeBERTa baseline | 93.32 | 92.53 | 84.64 | 84.78 | 84.56 |
| KD | 93.56 | 92.70 | 84.90 | 85.24 | 84.83 |
| $KD_{ens}$ (Ours) | **93.73**↑ | **92.89**↑ | **85.06** | **85.52**↑ | **85.29**↑ |

Table 9: Knowledge distillation for self-distillation is tested for a single teacher model compared to an ensemble of teacher models. All distillation results are an average from 10 random seeds. ↑ and ↓ represent results that are statistically significant with $p < 0.05$. The best results are in bold.

| Model | In-Distribution | | Out-of-Distribution | | |
|---|---|---|---|---|---|
| | SNLI-dev | SNLI-test | SNLI-hard | MNLI-mm | MNLI-m |
| | | | *BERT -> BERT:* | | |
| BERT baseline | 91.03 | 90.59 | 80.31 | 75.01 | 74.97 |
| KD | 91.43 | 90.72 | 80.57 | 75.42 | 75.50 |
| DTA (Ours) | 91.40 | 90.78 | 80.70 | 75.77↑ | 75.86↑ |
| $KD_{ens}$ (Ours) | 91.59 | 90.94 | 80.81 | 75.90 | 75.98 |
| $DTA_{ens}$ (Ours) | **91.65** | **91.00** | **81.00** | **76.42**↑ | **76.45**↑ |
| | | | *DeBERTa -> DeBERTa:* | | |
| DeBERTa baseline | 93.32 | 92.53 | 84.64 | 84.78 | 84.56 |
| KD | 93.56 | 92.70 | 84.90 | 85.24 | 84.83 |
| DTA (Ours) | 93.55 | 92.69 | 84.84 | 85.68↑ | 85.20↑ |
| $KD_{ens}$ (Ours) | 93.73 | **92.89** | **85.06** | 85.52 | 85.29 |
| $DTA_{ens}$ (Ours) | **93.74** | 92.82 | 84.97 | **86.18**↑ | **85.77**↑ |

Table 10: As Table 2 only shows results on MNLI, this table contains self-distillation results from all sets, including SNLI-dev, SNLI-test and SNLI-hard. All distillation results are an average from 10 random seeds. ↑ and ↓ represent results that are statistically significant with $p < 0.05$. We test the significance of the domain-targeted augmentation compared to standard knowledge distillation. The best results are in bold.

## E Details of experimental setup

To generate our domain-targeted data for MNLI, we use a text-curie-001 GPT-3 model to generate both the premises and hypotheses. However, when generating the additional data for HANS, we use text-davinci-003. We use the more expensive davinci model for this setting, as generating sentences that only contained specific words that we provided proved to be a more difficult task for text-curie-001.

We train all baseline models using a learning rate of $10^{-5}$ for 2 epochs using cross entropy loss, with the exception of Distil-RoBERTa (used as the student model in the MNLI-HANS setup). For Distil-RoBERTa and RoBERTa-large, to create a baseline similar to previous work, we train with learning rates of $2 \times 10^{-5}$ and $5 \times 10^{-6}$ respectively, in the case of Distil-RoBERTa training for

8 epochs. All baselines are trained with a linear learning rate (increasing for the first half of training, before decreasing for the second half). We use deberta-v3-base for our DeBERTa model, and bert-base-uncased for our BERT model. All baseline models are implemented from HuggingFace (Wolf et al., 2020). When using ensembles of BERT models, the eight models we use are different pre-trained models from Sellam et al. (2021) to maximise the variability between each BERT model.

The distillation stage is performed for 10 epochs with a learning rate of $10^{-6}$, with early stopping applied if there is no improvement within five epochs. The early stopping was not applied when using self-distillation, where we tested using an ensemble of teacher models. In this case, we chose the student model as the model with the best validation performance from the ensemble. Therefore, as the teacher models had lower validation performance than the student, the student validation performance was also likely to decrease during training. We also do not perform early stopping when evaluating on the adversarial HANS dataset, as performance on both MNLI-validation sets are likely to decrease as a result of improvements in HANS, or when distilling a DeBERTa teacher into a TinyBERT student model using DMU, where we do not see improvements in the validation set.

When applying Just Train Twice (JTT) or DMU, the minority examples are upsampled by 6 times, as Liu et al. (2021a) use for MNLI. We also upsample our augmented data for HANS (by 10 times), as we have fewer examples compared to MNLI (4,695 for HANS, compared to 47,955 for MNLI and 47,898 for SNLI). For $\text{DMU}_{\text{full}}$, we use an ensemble of 8 models, whereas the self-distillation experiments use an ensemble of 7 models (as one of the 8 models is used as the student model).

## F Full prompts

As described in Figure 2, first a prompt is provided to our generator model that asks the generator to create an example extract from a specified domain. For the popular magazine article domain, this prompt asks for an 'Example extract from a popular magazine article:', while for the travel guide the prompt is 'Example extract from a travel guide:', and for the fiction genre the prompt asks for 'Example extract from a fiction book:'. The fourth domain is extracts from government websites, where there are several different subcategories provided in MNLI, either using press releases, letters, speeches or reports. For this fourth domain, we therefore use the following prompts: 'Example extract from a press release on a public domain government website:', 'Example extract from a letter on a public domain government website:', 'Example extract from a speech on a public domain government website' and 'Example extract from a report on a public domain government website:'. The premise generation is zero-shot, with no examples provided to the generator. For the hypothesis generation, three examples from the in-distribution training data are provided. The three examples provided are different depending on the class (see Figure 4).

When generating data for SNLI, we generate the the premises using the prompt: 'Example flickr image caption:', with the hypotheses generated in the same method described above. As we are using MNLI training data for this setting, the examples in the prompts are provided from the MNLI training data (see Figure 5).

When generating premises for either MNLI or SNLI, only sentences that were at least 8 characters long were included. Premises that finished with a question mark were also not included in the augmented data. If more than one sentence was provided by the generator, and the first sentence did not meet this criteria, then we considered the

| Method | Test | SNLI-Hard | Δ |
|---|---|---|---|
| Baseline | 78.25 | 56.98 | |
| JTT | 76.25 | 55.93 | -1.05 |
| KD | 80.34 | 60.02 | +3.04 |
| *Ours:* | | | |
| $\text{DMU}_{\text{teach}}$ | 80.80 | 60.88 | +3.90 |
| $\text{DMU}_{\text{t-up}}$ | 80.78 | 61.15 | +4.17 |
| $\text{DMU}_{\text{t-full}}$ | **80.98** | **61.64** | **+4.66** |

Table 11: Performance of a JTT baseline (Liu et al., 2021a; Du et al., 2023) compared to our $\text{DMU}_{\text{teach}}$ method upsampling minority examples during the distillation that a single teacher model has misclassified. We up-sample examples that any model in an ensemble of teacher models incorrectly predicted while still using a single teacher model during the distillation ($\text{DMU}_{\text{t-up}}$), or also using an ensemble of teacher models for the distillation ($\text{DMU}_{\text{t-full}}$). For $\text{DMU}_{\text{t-full}}$, the same teacher models are used to identify the minority examples as those used during the distillation process. The baseline is a TinyBERT student, while JTT and KD methods use a BERT teacher.

second sentence as a possible premise.

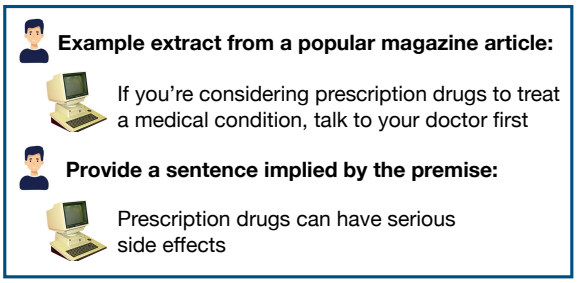


Figure 2: Our generator model is asked to create a sentence (premise) about a specified genre, before being asked to create a hypothesis that is either implied by the premise, contradicts the premise, or is neutral with respect to the premise. As the hypotheses generated are not faithful to the desired labels (as with this example), we use these examples as unlabelled data during knowledge distillation.

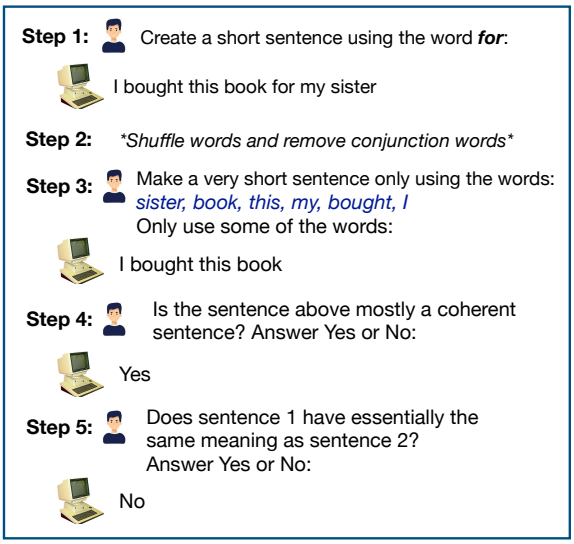


Figure 3: The process for generating augmented data for our word-overlap augmentation (WOA). In step 4, the model is asked if both the premise and the hypothesis are mostly coherent sentences. In this step, the premise-hypothesis pair is only added to our augmented dataset if the model answers 'yes' for both the premise and the hypothesis. Finally, the sentence pair is only included if the model answers 'no' to the final question in step 5.

Finally, the prompts used for HANS are provided in Figure 3.

## G Supporting P-values

In Table 12 we provide the full p-values supporting the statistical testing reported in Table 1, Table 2, Table 6, Table 8 and Table 10.

| Class | Prompts used to generate the hypothesis |
|---|---|
| Neutral | Premise: A few people in a restaurant setting, one of them is drinking orange juice.<br>Provide a sentence that is not implied by this premise: The people are eating omelettes.<br>Premise: A man, woman, and child enjoying themselves on a beach.<br>Provide a sentence that is not implied by this premise: A child with mom and dad, on summer vacation at the beach.<br>Premise: The school is having a special event in order to show the american culture on how other cultures are dealt with in parties.<br>Provide a sentence that is not implied by this premise: A high school is hosting an event.<br>Premise: <PREMISE><br>Provide a sentence that is not implied by this premise: |
| Entailment | Premise: A person on a horse jumps over a broken down airplane.<br>Provide a sentence implied by this premise: A person is outdoors, on a horse.<br>Premise: A boy is jumping on skateboard in the middle of a red bridge.<br>Provide a sentence implied by this premise: The boy does a skateboarding trick.<br>Premise: Two blond women are hugging one another.<br>Provide a sentence implied by this premise: There are women showing affection.<br>Premise: <PREMISE><br>Provide a sentence implied by this premise: |
| Contradiction | Premise: A man with blond-hair, and a brown shirt drinking out of a public water fountain.<br>Provide a sentence that contradicts this premise: A blond man wearing a brown shirt is reading a book on a bench in the park<br>Premise: High fashion ladies wait outside a tram beside a crowd of people in the city.<br>Provide a sentence that contradicts this premise: The women do not care what clothes they wear.<br>Premise: A boy is jumping on skateboard in the middle of a red bridge.<br>Provide a sentence that contradicts this premise: The boy skates down the sidewalk.<br>Premise: <PREMISE><br>Provide a sentence that contradicts this premise: |

Figure 4: Prompts used to generate hypotheses for MNLI, where <Premise> contains the premise generated by the generator model.

| Class | Prompts used to generate the hypothesis |
|---|---|
| Neutral | Premise: When the trust fund begins running cash deficits in 2016, the government as a whole must come up with the cash to finance Social Security's cash deficit by reducing any projected non-Social Security surpluses, borrowing from the public, raising other taxes, or reducing other government spending.<br>Provide a sentence that is not implied by this premise: The public would generally prefer to see the government reduce its spending in other areas to finance Social Security.<br>Premise: and it is nice talking to you all righty<br>Provide a sentence that is not implied by this premise: I talk to you every day.<br>Premise: yeah well you're a student right<br>Provide a sentence that is not implied by this premise: Well you're a mechanics student right?<br>Premise: <PREMISE><br>Provide a sentence that is not implied by this premise: |
| Entailment | Premise: One of our number will carry out your instructions minutely.<br>Provide a sentence implied by this premise: A member of my team will execute your orders with immense precision.<br>Premise: I burst through a set of cabin doors, and fell to the ground-<br>Provide a sentence implied by this premise: I burst through the doors and fell down.<br>Premise: right right well it's it's a beautiful city and but the problem is like first example when i was young they they took me to Las Vegas and that was the most boring place on earth<br>Provide a sentence implied by this premise: I think Las Vegas is the most boring place I know.<br>Premise: <PREMISE><br>Provide a sentence implied by this premise: |
| Contradiction | Premise: Fun for adults and children.<br>Provide a sentence that contradicts this premise: Fun for only children.<br>Premise: yeah so um also of course they they can they join the they can always join the military service they are considered citizens i believe<br>Provide a sentence that contradicts this premise: They can't join the military service<br>Premise: Vrenna and I both fought him and he nearly took us.<br>Provide a sentence that contradicts this premise: Neither Vrenna nor myself have ever fought him.<br>Premise: <PREMISE><br>Provide a sentence that contradicts this premise: |

Figure 5: Prompts used to generate data hypotheses for SNLI, where <Premise> contains the premise generated by the generator model.

| Model | In-Distribution | | Out-of-Distribution | | |
|---|---|---|---|---|---|
| | SNLI-dev | SNLI-test | SNLI-hard | MNLI-mm | MNLI-m |
| | | | *BERT -> TinyBERT:* | | |
| DMU | 0.0796 | 0.0034↓ | <0.0001↑ | <0.0001↓ | <0.0001↓ |
| DTA | 0.2076 | <0.0001↑ | 0.0068↑ | <0.0001↑ | <0.0001↑ |
| DTA with DMU | 0.9564 | 0.0042↑ | <0.0001↑ | <0.0001 ↑ | <0.0001↑ |
| | | | *BERT -> BERT:* | | |
| DTA | 0.4092 | 0.1580 | 0.0682 | 0.0004↑ | 0.0026↑ |
| DTA (ens) | 0.0846 | 0.2238 | 0.0916 | <0.0001↑ | <0.0001↑ |
| | | | *DeBERTa -> DeBERTa:* | | |
| DTA | 0.5366 | 0.7934 | 0.6014 | <0.0001↑ | <0.0001↑ |
| DTA (ens) | 0.6592 | 0.0570 | 0.2154 | <0.0001↑ | <0.0001↑ |
| | | | *DeBERTa -> BERT:* | | |
| DMU | 0.0102↑ | 0.0296↑ | 0.0134↑ | 0.0896 | 0.5460 |
| DTA | 0.0148↑ | 0.0372↑ | 0.1446 | <0.0001↑ | <0.0001↑ |
| DTA with DMU | 0.0026↑ | 0.0148↑ | 0.0008↑ | <0.0001↑ | <0.0001↑ |
| | | | *DeBERTa -> TinyBERT:* | | |
| DMU | <0.0001↓ | <0.0001↓ | <0.0001↑ | <0.0001↓ | <0.0001↓ |
| DTA | 0.7730 | <0.0001↑ | 0.0238↑ | <0.0001↑ | <0.0001↑ |
| DTA with DMU | <0.0001↓ | <0.0001↓ | <0.0001↑ | <0.0001↑ | <0.0001↑ |

Table 12: P-values for our main results tables, comparing our methods to standard knowledge distillation. We use two-tailed bootstrapping hypothesis testing (Efron and Tibshirani, 1993) to test statistical significance. ↑ represents results where there is a significant improvement compared to the baseline, whereas ↓ represents results that are significantly worse than the baseline. For the BERT -> BERT and DeBERTa -> DeBERTa settings, we compare our DTA method to standard knowledge distillation, while our DTA (ens) method is compared to standard knowledge distillation using an ensemble of teacher models.