FlipDA: Effective and Robust Data Augmentation for Few-Shot Learning

Anonymous ACL submission

Abstract

Most previous methods for text data augmentation are limited to simple tasks and weak baselines. We explore data augmentation on hard tasks (i.e., few-shot natural language understanding) and strong baselines (i.e., pretrained models with over one billion parameters). Under this setting, we reproduced a large number of previous augmentation methods and found that these methods bring marginal gains at best and sometimes degrade the performance much. To address this challenge, we propose a novel data augmentation method FlipDA that jointly uses a generative model and a classifier to generate label-flipped data. Central to the idea of FlipDA is the discovery that generating label-flipped data is more crucial to the performance than generating label-preserved data. Experiments show that FlipDA achieves a good trade-off between effectiveness and robustness—it substantially improves many tasks while not negatively affecting the others.

1 Introduction

Data augmentation is a method to augment the training set by generating new data from the given data. For text data, basic operations including replacement, insertion, deletion, and shuffle have been adopted widely and integrated into a wide range of augmentation frameworks (Zhang et al., 2015; Wang and Yang, 2015; Xie et al., 2020a; Kobayashi, 2018; Wei and Zou, 2019). Generative modeling methods such as back-translation have also been employed to generate augmented samples (Fadaee et al., 2017; Sennrich et al., 2016). However, there are two major limitations. First, some general augmentation methods are based on weak baselines without using large-scale pretrained language models. Recent work showed that some of the data augmentation methods are less useful when combined with large pretrained models (Longpre et al., 2020). Second, most prior studies are carried on simple tasks such as single-sentence classification where it is easier to generate legit augmented samples. For harder tasks such as natural language inference (e.g., telling whether sentence A entails sentence B), it is not clear whether previous methods still help.

This work takes a step further to study data augmentation under strong baselines and hard tasks. Our study employs large-scale pretrained language models such as DeBERTa (He et al., 2020c) with over one billion parameters as baselines. Moreover, we target a very challenging setting—few-shot natural language understanding (NLU). Following (Schick and Schutze, 2021), we consider challenging NLU tasks including question answering, textual entailment, coreference resolution, and word sense disambiguation, and use only 32 training examples for each task. Under this setting, we reproduced several widely-used prior methods for data augmentation. Our experiments lead to two unexpected discoveries: (1) most of prior augmentation methods bring only marginal gains at best and are not effective for most tasks; (2) in many cases, using data augmentation results in instability in performance and even entering a failure mode; i.e., performance may drop by a lot or fluctuate severely depending on which pretrained model is used. The above issues prevent these augmentation methods from practical usage for few-shot learning.

We propose a novel method FlipDA that achieves both effectiveness and robustness for hard few-shot tasks. Preliminary experiments showed that label-flipped data often largely improve the generalization of pretrained models, compared to augmented data that preserve the original labels. Based on this observation, FlipDA first generates data using word substitution based on a pretrained T5 (Raffel et al., 2020) and uses a classifier to select label-flipped data. Experiments demonstrate FlipDA substantially improves performance on many of the hard tasks, outperforming previous augmentation baselines in terms of average performance by a large margin. Moreover, FlipDA is robust across different pretrained models and different tasks, avoiding failure modes.

2 Related Work

Data Augmentation. An important type of augmentation methods are based on *word substitution*, such as synonym replacement (Zhang et al., 2015), KNN replacement (Wang and Yang, 2015; Vijayaraghavan et al., 2016), Unif replacement (Xie et al., 2020a), TF-IDF replacement (Xie et al., 2020a), Bi-RNN replacement (Kobayashi, 2018), and other entity replacement methods (Raiman and Miller, 2017; Miao et al., 2020; Yue and Zhou, 2020) etc. EDA (Wei and Zou, 2019) combines four simple augmentation methods, and back translation (BT) (Fadaee et al., 2017; Sennrich et al., 2016; Yu et al., 2018) is also widely used. Unfortunately, EDA and BT are shown to be less useful with large pretrained models (Longpre et al., 2020).

Other types of augmentation methods are based on the *perturbation in the feature space* (Zhang et al., 2018a; Guo et al., 2020; Chen et al., 2020b,a; Miao et al., 2020; Kumar et al., 2019), *generation* (Xia et al., 2020; Li et al., 2019; Yoo et al., 2019; Ng et al., 2020; Liu et al., 2020; Hou et al., 2018), and *large pretrained models* (such as GPT-2, BERT, and BART) (Kumar et al., 2020; Anaby-Tavor et al., 2020; Yoo et al., 2021), etc.

Self-training. Self-training (III, 1965) iteratively augments training data by labeling unlabeled data with a trained model (Yarowsky, 1995; Riloff, 1996). Knowledge distillation and pseudo-labeling are special forms of self-training (Hinton et al., 2015; Lee et al., 2013; Reed et al., 2015). Strong data augmentation (Zoph et al., 2020), equal-or-larger model (Xie et al., 2020b), additional noise (Xie et al., 2020b; He et al., 2020a), and feedback of the student's performance (Pham et al., 2020) are helpful for self-training.

Self-training bears similarity to the second phase of FlipDA where a teacher model is used to filter samples. Different from self-training, FlipDA leverages the advantages of label flipping to improve performance and does not rely on unlabeled data.

Label Flipping. Our manual label flipping augmentation procedure is analogous to (Kaushik et al., 2020) and (Gardner et al., 2020). Kaushik et al. (2020) aimed to mitigate the effects of learning spurious features. Gardner et al. (2020) targeted reducing systematic gaps in the dataset. In contrast, we target improving few-shot generalization. Moreover, we measure the performance on an existing i.i.d. test set while Kaushik et al. (2020) and Gardner et al. (2020) created more challenging test sets. Most importantly, we propose an automatic method of label flipping, going beyond manual efforts.

Contrastive Learning. FlipDA is connected to contrastive learning (CL) (He et al., 2020b; Chen et al., 2020c) in that they both improve generalization by considering label differences. CL uses data augmentation to generate positive instances and uses samples existing in the dataset as negative samples, while FlipDA shows that negative samples can be automatically generated. While previous work on CL focuses on training with large datasets, our experiments show that augmenting a small dataset can improve few-shot generalization.

3 Few-Shot Data Augmentation

3.1 Setting

Few-Shot NLU Tasks. This work considers a collection of "difficult" NLU tasks from SuperGLUE (Wang et al., 2019) that require in-depth understanding of the input in order to obtain high performance, including coreference resolution (Levesque et al., 2011), causal reasoning (Gordon et al., 2012), textual entailment (de Marneffe et al., 2019; Dagan et al., 2005), word sense disambiguation (Pilehvar and Camacho-Collados, 2019), and question answering (Clark et al., 2019; Khashabi et al., 2018; Zhang et al., 2018b). Following Schick and Schutze (2021), we used only 32 training examples to construct a few-shot setting to further increase the difficulty.

Large-Scale Pretrained Models. Our setting assumes a large-scale pretrained language model (Devlin et al., 2019; Lan et al., 2020; He et al., 2020c) is available and few-shot learning is performed based on the pretrained model. This setting is crucial since previous studies found that using a strong pretrained model as the baseline eliminates the benefits of data augmentation (Longpre et al., 2020) while large pretrained models are becoming more and more available. Our main result is based on DeBERTa (He et al., 2020c) with over one billion parameters. We also provide results with ALBERT which has fewer parameters (Lan et al., 2020).

Preliminary Experiments with Prior Methods. Our preliminary experiments with a large number of previous methods (in Section 4) lead to a conclusion that there is not an effective and robust method available for this hard setting. We will discuss how we tackle this challenge by proposing a novel data augmentation method FlipDA in later sections.

3.2 Desiderata: Effectiveness and Robustness

We propose key desiderata for data augmentation methods under the setting of few-shot learning.

1. **Effectiveness.** A data augmentation method should be able to improve performance on certain tasks in a significant manner.
2. **Robustness.** A data augmentation method should not suffer from a failure mode in all cases. Failure modes are common for few-shot learning where some minor changes might cause substantial performance drop. We argue this should be used as a key evaluation metric. We consider two types of robustness: (1) robustness w.r.t. different base pretrained models and (2) robustness w.r.t. various tasks.

3.3 Effectiveness: Manual Label Flipping Improves Performance

Since previous methods are not sufficiently effective and robust in our preliminary experiments (see Tables 5 and 6 in Section 4 for details), we use manual augmentation to investigate what kind of augmented data is beneficial for large pretrained models in the few-shot setting. We mainly study two types of data augmentation—one that preserves the labels and the other that flips the labels. Since manual augmentation is time consuming, we select a subset of representative SuperGLUE tasks here.

To augment label-flipped data, the following principle is applied—making minimal changes to the original text sample to alter the label. Augmentation includes word addition, deletion, and substitution. To augment label-preserved data, we substitute some of the words with semantically similar words but make sure that the label is unchanged.

Table 1: Manual data augmentation results. We manually write augmented examples that preserve or flip the label. Flipping the labels substantially improves performance on CB, RTE and WSC by up to 10 points, while preserving the labels only has minor gains.

Tasks	No DA	Preserves	Flips
BoolQ	78.21±0.27	**78.55**±0.49	77.68±0.08
CB-Acc	81.55±4.12	82.14±3.57	**91.07**±3.09
CB-F1	72.16±7.02	77.07±4.91	**88.14**±3.93
COPA	90.33±1.15	**91.33**±0.58	90.33±0.58
RTE	68.11±3.28	67.63±2.61	**76.05**±0.75
WSC	79.49±2.22	78.53±2.78	**85.58**±0.96

Results are shown in Table 1.[1] Flipping labels substantially improves performance on three of the tasks by up to 10 points, while preserving the labels only has minor gains. In contrast, many of prior methods on data augmentation focus on creating data examples that are assumed to have the same labels as the original ones. This might explain why previous augmentation methods are not sufficiently effective for the few-shot setting. Some of the label-flipped augmented examples are shown in Table 2. We conjecture that label flipping augmentation provides useful information about the important components in a sentence that determine the label. In other words, augmented samples provide intermediate supervision that explains the predictions, improving generalization in a few-shot setting.

There is a caveat about this manual augmentation experiment. Although we follow certain principles and pay much attention to the augmentation quality, the manual augmentation procedure is inevitably subjective and hard to reproduce. For reference, we will make our manually augmented dataset publicly available. More importantly, we will design an automatic method (FlipDA) in the following sections for objective evaluation and reproducibility.

3.4 Robustness: What Contribute to Failure Modes?

We also analyze why augmentation methods usually suffer from failure modes. Most augmentation methods are based on a label preserving assumption, while it is challenging for automatic methods to always generate label-preserved samples. We first examine the samples generated by prior automatic methods EDA (Wei and Zou, 2019) and KNN (Wang and Yang, 2015) in Table 4. In the first example, a keyword "rabies" is deleted, which not only results in a grammatically incorrect expression but also eliminates the key information to support the hypothesis. In the second example, the "Lake Titicaca" is replaced by "Lake Havasu", which results in a label change from entailment to non-entailment. If a model is trained on these noisy augmented data with the label preserving assumption, performance degradation is expected.

We further experimented with EDA (Wei and Zou, 2019) on the RTE task (Dagan et al., 2005) to verify the cause of failure modes. Using EDA de-

[1] For each original example, we produce one augmented example for each type. The augmented data and the original data are combined for training. Following Schick and Schutze (2021), we train each pattern with three seeds and ensemble these (pattern, seed) pairs. We repeat this ensemble process 3 times and report their mean and standard deviation.

Table 2: Label-flipped examples from manual augmentation. The augmentation principle is to make minimal changes that are sufficient to alter the labels. Black denotes original examples, and blue denotes augmented examples. The second task WSC is coreference resolution, which is to extract the referred entitiy from the text. In this case, “label” is defined as the referred entity (denoted in red), and label flipping is defined as modifying the entity.

RTE	**Premise:** This case of rabies in western Newfoundland is the first case confirmed on the island since 1989. **Hypothesis:** A case of rabies was confirmed. **Entailment:** True **Hypothesis:** A case of smallpox was confirmed. **Entailment:** False
WSC	**Text:** The city councilmen refused the demonstrators a permit because they advocated violence. **Text:** The city councilmen refused the criminals a permit because they advocated violence.

Table 3: Performance of correcting the wrong-labeled augmented data by EDA on RTE. W-Del denotes replacing the wrong-labeled augmented samples with corresponding original samples, and W-Flip denotes flipping the labels of the wrong-labeled augmented samples to be the correct ones. The results show that in this case data augmentation with the label-preserving assumption substantially contributes to performance drop.

	No DA	EDA	W-Del	W-Flip
ALBERT	61.40	58.33	59.39	61.07
DeBERTa	81.95	77.38	80.75	83.39

creases the performance by a few percentage points with both ALBERT and DeBERTa, entering a failure mode. We identified two types of noise in the augmented samples: (1) grammatical errors that lead to the difficulty of understanding and (2) modification of key information that alters the labels. We experimented with (1) replacing these noisy samples with the original ones and (2) correcting the labels of the noisy samples. [2] As Table 3 shows, both replacing and correcting noisy samples largely improve performance to prevent the failure mode. Moreover, correcting the labels brings large gains, indicating label flipping tends to alleviate the issue.

To reiterate, these experiments involve subjective factors and are merely meant to show the intuition of FlipDA, rather than proving its superiority.

3.5 FlipDA: Automatic Label Flipping

Observations in Sections 3.3 and 3.4 show that label-flipping could benefit few-shot NLU in both effectiveness and robustness. Reducing grammatical errors is also key to preventing failure modes. This motivates our development of FlipDA that automatically generates and selects label-flipped data without label-preserving assumption.

FlipDA consists of 4 steps as shown in Figure 1:

1. Train a classifier (e.g., finetuning a pretrained model) without data augmentation.
2. Generate label-preserved and label-flipped augmented samples.
3. Use the classifier to select generated samples with largest probabilities for each label.
4. Retrain the classifier with the original samples and the additional augmented samples.

Formally, given a few-shot training set $\{(x_i, y_i)\}_i$ where x_i is text (possibly a set of text pieces or a single piece) and $y_i \in \mathcal{Y}$ is a label. We finetune a pretrained model f to fit the conditional probability for classification $f(x, y) = \hat{p}(y|x)$. In the second step, we generate augmented samples from the original ones. For each training sample x_i, we generate a set of augmented samples $S_i = \{\tilde{x}_{i,1}, \tilde{x}_{i,2}, \cdots\}$. In our implementation, we first use a cloze pattern (Schick and Schutze, 2021) to combine both x and y into a single sequence, and then randomly mask a fixed percentage of the input tokens. This is followed by employing a pretrained T5 model (Raffel et al., 2020) to fill the blanks to form a new sample x' (see Appendix A.3 for more details). We find it beneficial to remove the sample if T5 does not predict y given x'. Note that using T5 to generate augmented samples does introduce additional knowledge and reduce grammatical errors, but naively using T5 for augmentation without label flipping and selection does not work well (see ablation study in Section 4). After generating the augmented samples, we use the classifier f for scoring. Specifically, let S_i be a set of augmented samples generated from the original sample (x_i, y_i). For each label $y' \neq y_i$, we construct a set

$$S_{i,y'} = \{x | x \in S_i \text{ and } y' = \arg\max_y \hat{p}(y|x)\}$$

which contains all augmented samples with y' being highest-probability class. Given the set $S_{i,y'}$, we select the sample with the highest predicted probability

$$x', y' = \arg\max_{x \in S_{i,y'}, y = y'} \hat{p}(y|x)$$

where x' is a sample in the generated set, y' is the flipped label, and the estimated probability $\hat{p}(y'|x')$ scored by the model f is the largest in $S_{i,y'}$. Af-

[2] For label correction, if a sample has severe grammatical mistakes and is not understandable by human, we always mark it as “not entailment”. This is related to an interesting phenomenon that label flipping is usually asymmetric for NLU tasks. We will discuss more of the phenomenon in Section 4.5.

Table 4: Augmented example with wrong labels. The first is by EDA, and the second is by KNN. Black denotes original examples, blue denotes augmented examples and red denotes key entity. The phenomenon of asymmetric label transformation (e.g., flipping from "entailment" to "not entailment" is more common) is further studied in Section 4.5.

Premise: This case of rabies in western Newfoundland is the first case confirmed on the island since 1989. **Hypothesis:** A case of rabies was confirmed. **Entailment:** True **Premise:** this case of in western newfoundland is the first case confirmed on the island since 1989. **Hypothesis:** a case of rabies was confirmed. **Entailment:** False
Premise: ... including a peasant rally near Santa Cruz and a visit to naval installations on Lake Titicaca ... **Hypothesis:** Lake Titicaca has a naval installation. **Entailment:** True **Premise:** ... includes a peasant rally near santa cruz and a visit to naval installations on lake titicaca ... **Hypothesis:** lake havasu has a naval installation . **Entailment:** False

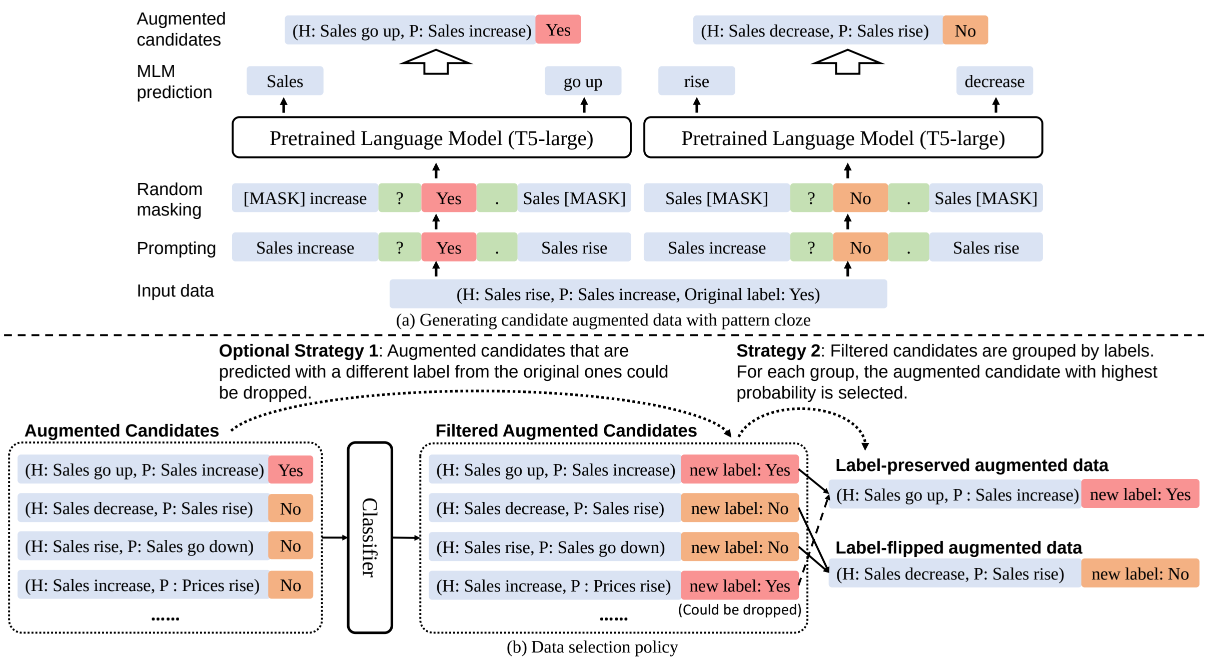

Figure 1: An illustration of (a) our prompt-based augmentation algorithm for both preserved/flipped labeled data, and (b) our data selection policy. Whether to use Strategy 1 depends on the relative power of the augmentation model and the classification model. If the augmentation model is accurate enough, drop the candidates with inconsistent labels, and otherwise, keep it.

ter selecting the label-flipped example (x', y'), we add (x', y') to the augmented training set. In other words, we only add an example into the training set if the model f considers the flipped label to be correct. We apply this procedure to each possible label $y' \neq y_i$. In case $S_{i,y'}$ is empty, we do not add any examples to the training set. In practice, we find it beneficial to also add the example with the highest probability of label preserving, using the same procedure. After augmenting the training set, we retrain the classifier f to obtain the final model.

4 Experiments

4.1 Experimental Setup

Baselines. We take seven augmentation methods as the baseline, including Synonym Replacement (SR) (Zhang et al., 2015), KNN Replacement (KNN) (Wang and Yang, 2015), Easy Data Augmentation (EDA) (Wei and Zou, 2019), Back Translation (BT) (Fadaee et al., 2017), TinyBERT (T-BERT) (Jiao et al., 2019), T5-MLM, and MixUP (Zhang et al., 2018a). For more details about baseline selection and implementation, please refer to Appendix A.2.

Evaluation Protocol We evaluate augmentation methods based on PET (Schick and Schutze, 2021). Following PET, we take a set of pre-fixed hyperparameters (see Appendix A.1). Considering few-shot learning is sensitive to different patterns and random seeds (Dodge et al., 2020; Schick and Schutze, 2021), we reported the average performance over multiple patterns and 3 iterations.

We evaluate FlipDA on 8 tasks with 2 pretrained models. For effectiveness, we use exactly the same metrics (i.e., accuracy, F1, and EM) as PET (Schick and Schutze, 2021). For robustness, we propose a new metric MaxDrop (MD), which measures the maximum performance drop compared to not using augmentation over multiple tasks

Table 5: Performance of baseline methods and FlipDA based on PET and ALBERT-xxlarge-v2 ("baseline" denotes the original PET with no data augmentation. Underline denotes values that outperform "baseline". Bold denotes the best-performed ones of the task). "Avg." is the average of scores and "MD" (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	BoolQ Acc.	CB Acc./F1	COPA Acc.	RTE Acc.	WiC Acc.	WSC Acc.	MultiRC EM/F1a	ReCoRD Acc./F1	Avg.	MD
Baseline	72.47	82.74/74.84	88.33	61.40	51.27	77.03	33.04/74.64	86.19/86.75	71.20	-
SR	74.98	83.33/78.12	87.50	59.24	51.25	78.74	34.09/75.55	85.63/86.12	71.64	2.16
KNN	74.51	82.14/74.39	85.50	61.91	51.62	75.00	32.72/75.20	84.77/85.31	70.73	2.83
EDA	72.68	81.10/73.58	84.50	58.33	51.81	75.85	28.74/73.05	85.39/85.95	69.63	3.83
BT-10	74.59	82.44/77.72	83.00	55.93	50.77	76.82	32.96/74.69	85.34/85.88	70.08	5.47
BT-6	75.36	82.89/76.55	86.50	57.46	51.01	77.78	34.85/75.82	85.83/86.41	71.16	3.94
T-BERT	72.60	85.42/82.35	84.67	58.66	51.10	78.95	30.47/73.20	84.57/85.12	70.82	3.66
T5-MLM	73.86	83.48/75.01	87.33	62.27	51.08	**79.17**	33.79/74.06	85.15/85.69	71.54	1.05
MixUP	75.03	83.93/79.28	70.33	62.06	52.32	68.70	34.06/74.66	80.93/81.70	68.22	18.00
FlipDA	**76.98**	**86.31/82.45**	**89.17**	**70.67**	**54.08**	78.74	**36.38/76.23**	**86.43/86.97**	**74.63**	**0.00**

Table 6: Performance of baseline methods and FlipDA based on PET and DeBERTa-v2-xxlarge. "baseline" denotes the original PET without data augmentation. Underlines denote values that outperform the "baseline". 'FlipDA cls" denotes the same classifier as in FlipDA for filtering candidate augmented data. Bold denotes the best-performing ones of the task. Wave-lines denote methods with FlipDA classifiers that outperform the original (without FlipDA classifier) version. "Avg." is the average of scores and "MD" (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	BoolQ Acc.	CB Acc./F1	COPA Acc.	RTE Acc.	WiC Acc.	WSC Acc.	MultiRC EM/F1a	ReCoRD Acc./F1	Avg.	MD
Baseline	78.30	85.42/79.31	87.67	81.95	58.74	80.13	40.40/78.14	90.24/90.77	77.36	-
SR	77.37	87.20/80.28	87.00	76.29	58.88	80.88	35.70/76.25	89.06/89.55	76.18	5.66
+FlipDA cls	80.37	83.48/79.01	85.50	82.79	59.75	78.10	37.51/76.84	89.27/89.77	76.81	2.17
KNN	75.35	83.78/75.61	85.00	75.45	59.63	79.38	29.84/69.14	88.26/88.75	74.06	9.78
+FlipDA cls	78.51	87.50/82.53	88.33	82.79	58.66	76.39	38.86/77.29	90.31/90.78	77.29	3.74
EDA	74.42	83.63/76.23	85.83	77.38	59.28	78.74	37.02/77.05	88.11/88.60	75.12	4.57
+FlipDA cls	76.20	87.35/82.35	88.17	82.31	59.94	79.81	42.84/79.30	90.29/90.77	77.86	2.10
BT-10	75.38	88.24/84.03	85.33	79.66	59.46	76.71	38.88/77.79	90.08/90.56	76.42	3.42
+FlipDA cls	79.97	85.71/80.50	87.50	78.58	60.08	77.24	40.97/78.25	90.39/90.94	77.09	3.37
BT-6	76.78	86.46/82.56	84.00	81.47	58.69	75.11	40.53/79.01	90.20/90.73	76.35	5.02
+FlipDA cls	79.63	84.67/77.94	77.00	82.91	59.58	77.56	39.03/77.64	90.41/90.95	75.88	10.67
T-BERT	70.53	86.01/82.77	86.17	72.80	57.49	78.85	34.94/75.17	86.94/87.47	74.06	9.15
+FlipDA cls	80.24	86.16/81.25	83.00	82.19	59.49	79.59	40.78/78.64	90.65/91.17	77.35	4.67
T5-MLM	77.39	83.04/73.71	88.17	81.23	60.73	**82.37**	35.02/74.98	89.71/90.25	76.66	4.27
MixUP	63.41	71.13/60.83	72.00	68.59	57.70	68.38	39.24/76.88	60.12/60.93	64.33	29.98
FlipDA	**81.80**	**88.24/87.94**	**90.83**	**83.75**	**65.12**	78.85	**44.18/80.00**	**91.02/91.56**	**80.23**	**1.28**

for a given method. Given tasks t_1,...,t_n, a target method M, and a baseline method M_B, MD is defined as MD=$\max_{t \in \{t_1,...,t_n\}} \max(0, \text{score}_{t,M_B} - \text{score}_{t,M})$, where $\text{score}_{t,M}$ (score_{t,M_B}) denotes the performance of method M (M_B) on task t. Smaller values indicate better robustness w.r.t tasks.

4.2 Main Results

Results are presented in Table 5 and Table 6. We observe that FlipDA achieves the best performance among all data augmentation methods in both effectiveness (Avg.) and robustness (MD) on both ALBERT-xxlarge-v2 and DeBERTa-v2-xxlarge.

Specifically, FlipDA achieves an average performance of 74.63 on ALBERT-xxlarge-v2 and an average of 80.23 on DeBERTa-v2-xxlarge, both of which outperform baselines by around 3 points. It suggests FlipDA is effective in boosting the performance of few-shot tasks by augmenting high-quality data without causing too many side effects.

FlipDA shows improvements on all tasks except WSC, while all the other methods only work on a few tasks (denoted with underlines). Such observations are consistent with the MaxDrop results, where FlipDA achieves the lowest MaxDrop value of 0.0 on ALBERT-xxlarge-v2 and 1.28 on DeBERTa-v2-xxlarge. This implies FlipDA is robust to different types of tasks, while other augmentation methods could only be effective for partial tasks and not sufficiently robust.

4.3 Ablation Study of FlipDA

Effectiveness of Pattern-based Data Cloze To study different methods of obtaining candidate augmented data, we feed candidates obtained by different methods into the same classifier (as FlipDA uses). Table 6 shows the ablation results.

FlipDA outperforms all the other baseline meth-

Table 7: Ablation study on label-flipped data v.s. label-preserved data on DeBERTa-v2-xxlarge. Bold denotes the best-performed results. Underlines denote the second-best results. "Avg." is the average of scores and "MD" (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	BoolQ Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a
Baseline	78.30	81.95	58.74	40.40/78.14
Both	**81.80**	**83.75**	**65.12**	**44.18/80.00**
Flipped	80.91	83.51	62.34	42.70/79.37
Preserved	77.04	80.99	60.08	39.55/78.30

ods with a classifier (i.e., with "FlipDA cls"). Other methods of obtaining augmented data candidates cannot reach similar performance as FlipDA when combining with FlipDA classifier, which proves the effectiveness of our pattern-based data cloze strategy with T5. Reasons could be that T5-based augmentation produces samples with less grammatical errors. (will further discuss in Sec 4.7). Moreover, T5-style blank filling could produce samples that are more compatible with label flipping.

Effectiveness of FlipDA Classifier We then compare the performance of different methods with and without the FlipDA classifier. According to Table 6, most baseline methods with the FlipDA classifier outperform the original version in terms of both effectiveness (Avg.) and robustness (MD). This demonstrates that the FlipDA classifier which is capable of flipping labels and filtering data is effective in augmenting high-quality data and improving few-shot NLU performance. The only exceptions is BT-6. The reason could be data augmented by back translation usually lack diversity, and using the FlipDA classifier further decreases diversity and hurts its performance.

The improvement brought by the FlipDA classifier is more consistent on BoolQ, RTE, and MultiRC. This may be because these tasks involve predicting single token with two opposite choices, and thus label flipping might happen more often. Some of the other tasks such as COPA and WSC involve predicting multiple tokens, which makes generating label-flipped data more difficult. This leads to less substantial improvement on these tasks.

4.4 Analysis of Label-Flipping v.s. Label-Preservation

A follow-up question is how label-flipped data and label-preserved data respectively contribute to the overall improvements. We run decoupling label-flipped data and label-preserved data. Results are in Table 7, where bold text represents the best-

Table 8: Results of different label transformation on DeBERTa. RTE: A/B denotes entail/not-entail, indicating whether the given premise entails with the given hypothesis. BoolQ: A/B denotes False/True, representing the answer for the given yes-no questions. WiC: A/B refers to F/T, indicating whether the target word shares the same meaning in both given sentences. MultiRC: A/B denotes 0/1, representing whether the given answer is correct for the given question.

Method	BoolQ Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a
A→A	78.89	76.17	55.66	36.57/76.77
A→B	78.34	**80.87**	**57.99**	**40.94/78.93**
B→B	74.55	75.57	57.30	39.73/78.03
B→A	**80.33**	76.90	56.20	**40.10/78.41**

Table 9: Results of different strategies for choosing augmented data on DeBERTa (xxlarge). "Avg." is the average of scores and "MD" (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	BoolQ Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a
Baseline	78.30	81.95	58.74	40.40/78.14
Noisy Student	**82.13**	82.79	64.11	39.99/77.43
Default	81.80	83.75	65.12	**44.18/80.00**
Global TopP	81.22	81.11	64.19	42.56/79.16
Global TopK	80.71	81.35	**65.13**	41.14/78.52
Diverse TopK	**81.99**	**84.59**	63.85	42.64/79.13

performed methods. We conclude that augmenting both label-flipped and label-preserved data leads to the best average performance. Besides, values with underlines denote the second-best performance, most of which are augmenting only label-flipped data. Augmenting only label-preserved data leads to the worst performance, even slightly underperforming the non-augmentation baseline. This demonstrates the high effectiveness of label-flipping. This aligns well with our analysis in Section 3.3. More results are in Appendix A.7 and A.8.2.

4.5 Analysis of Label Transformation

Section 4.4 proves that label-flipped augmented data are more effective in improving few-shot performance than label-preserved ones. It is even more intriguing to study which direction of label flipping is able to benefit the few-shot performance to the maximum extent. We experiment with 4 binary classification tasks, i.e., RTE, BoolQ, WiC, and MultiRC. Each task has 4 directions of label transformation. We conduct experiments that augment data in each of the four directions respectively and compare their effectiveness.

Results on DeBERTa [3] are shown in Table 8. We can see that some tasks are asymmetric, i.e., transforming in one direction is more beneficial

[3]Results on ALBERT are in Appendix A.8.3.

Table 10: Some augmented examples selected by our model (DeBERTa) in RTE. Black denotes original examples, and blue denotes augmented examples.

Entailment $\rightarrow$ Not Entailment	**Premise:** The university server containing the information relating to Mason's ID cards was illegally entered by computer hackers. **Hypothesis:** Non-authorized personnel illegally entered into computer networks. **Premise:** The university server that holds the information about Mason 's ID number was not compromised by hackers **Hypothesis:** security personnel illegally hack into computer systems
Not Entailment $\rightarrow$ Entailment	**Premise:** Vodafone's share of net new subscribers in Japan has dwindled in recent months. **Hypothesis:** There have been many new subscribers to Vodafone in Japan in the past few months. **Premise:** Vodafone 's number of net new subscribers to Japan has increased in recent months **Hypothesis:** There have been net new subscribers to Vodafone in Japan in recent months

than the other, such as BoolQ, RTE, and WiC. We conjecture that it is because it is relatively easy for a model to generate samples with answers in some direction (from "yes" to "no" in BoolQ, from 'entailment' to "not entailment" in RTE, and so on). While some tasks are symmetric, i.e., the difference between the two directions is not significant, such as MultiRC. On all tasks, even though some direction is better than others, augmenting with only one direction will affect the label distribution. This will likely lead to a lower performance than the baseline. Augmenting with all directions is still necessary for the best performance.

4.6 Analysis of Strategies for Augmented Data Selection

We propose four plausible strategies for augmented data selection, and quantitatively evaluate them.

1. **Default Strategy.** It is described in Section 3.5, with no hyper-parameters.
2. **Global Top**K. For each label transformation direction, all the candidate augmented data are gathered and sorted by their predicted probabilities, and the top-K (or top-$r\%$) samples with the highest probabilities are selected.
3. **Global Top**P. Similar to Global Top$K$, but augmented data with predicted probabilities higher than a threshold P are selected.
4. **Diverse Top**K. Similar to Global Top$K$ except that a mechanism is used to balance between the original samples. Concretely, we first select the top-1 augmented samples of each original sample (ranked by decreasing probabilities), and then select the top-2, top-3, etc, until K samples have been selected.

Since FlipDA can be viewed as a self-training algorithm, we also add a self-training algorithm Noisy Student (Xie et al., 2020b) as another baseline. We treat the augmented data as unlabeled data and add noises with a dropout rate of 0.1.

Table 9 shows the results of different strategies on different tasks. More results are in Appendix A.7 and Appendix A.8.4. For Global Top$P$, we set the threshold P at 0.9 or 0.95, whichever is better. For Global Top$K$ and Diverse Top$K$, we select the top 10% or 20% augmented examples, whichever is better. Our strategies outperform Noisy Student. Among our four data selection strategies, the Default strategy and Diverse Top$K$ perform the best. Both methods emphasize diversity by using augmented data from different samples. This demonstrates the importance of data diversity and balance for augmented data selection.

4.7 Case Study

We show two label-flipped augmented cases on the RTE task by FlipDA in Table 10. Please refer to Appendix A.9 for more augmented examples.

The first case adds "not" to the premise and therefore the label flips. The second case changes "dwindles" to its antonym "increased", and then the label changes from "Not Entailment" to "Entailment". We can see that the way to change or keep the label is rich and natural. Moreover, the generation quality is improved compared to cases generated by EDA in Table 4, which also addresses the concerns of generation quality raised in Section 3.4.

5 Conclusions

We propose to study few-shot NLU based on large-scale pretrained models. Two key desiderata, i.e., effectiveness and robustness, are identified. Based on the empirical insight that label flipping improves few-shot generalization, we propose FlipDA with automatic label flipping and data selection. Experiments demonstrate the superiority of FlipDA, outperforming previous methods in terms of both effectiveness and robustness. In the future, it will be crucial to further increase the diversity and quality of augmented data for better performance.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In AAAI, pages 7383–7390. AAAI Press.

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. 2020a. Local additivity based data augmentation for semi-supervised NER. In EMNLP (1), pages 1241–1251. Association for Computational Linguistics.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020b. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 2147–2157. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020c. A simple framework for contrastive learning of visual representations. In ICML, volume 119 of Proceedings of Machine Learning Research, pages 1597–1607. PMLR.

Christopher Clark, Kenton Lee, Ming-Wei Chang, T. Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. ArXiv, abs/1905.10044.

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

Marie-Catherine de Marneffe, M. Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. CoRR, abs/2002.06305.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In ACL (2), pages 567–573. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In EMNLP (Findings), volume EMNLP 2020 of Findings of ACL, pages 1307–1323. Association for Computational Linguistics.

A. Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In SemEval@NAACL-HLT.

Demi Guo, Yoon Kim, and Alexander M. Rush. 2020. Sequence-level mixed sample data augmentation. In EMNLP (1), pages 5547–5552. Association for Computational Linguistics.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020a. Revisiting self-training for neural sequence generation. In ICLR. OpenReview.net.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020b. Momentum contrast for unsupervised visual representation learning. In CVPR, pages 9726–9735. IEEE.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020c. Deberta: Decoding-enhanced bert with disentangled attention. ArXiv, abs/2006.03654.

Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. 2015. Distilling the knowledge in a neural network. ArXiv, abs/1503.02531.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In COLING, pages 1234–1245. Association for Computational Linguistics.

H. J. Scudder III. 1965. Probability of error of some adaptive pattern-recognition machines. IEEE Trans. Inf. Theory, 11(3):363–371.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling BERT for natural language understanding. CoRR, abs/1909.10351.

Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes A difference with counterfactually-augmented data. In ICLR. OpenReview.net.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and D. Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In NAACL-HLT.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New

Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 452–457. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. CoRR, abs/2003.02245.

Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. 2019. A closer look at feature space data augmentation for few-shot intent classification. In DeepLo@EMNLP-IJCNLP, pages 1–10. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. ArXiv, abs/1909.11942.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML, volume 3.

H. Levesque, E. Davis, and L. Morgenstern. 2011. The winograd schema challenge. In KR.

Juntao Li, Lisong Qiu, Bo Tang, Dongmin Chen, Dongyan Zhao, and Rui Yan. 2019. Insufficient data can also rock! learning to converse using smaller data with augmentation. In AAAI, pages 6698–6705. AAAI Press.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Jiancheng Lv, Nan Duan, and Ming Zhou. 2020. Tell me how to ask again: Question data augmentation with controllable rewriting in continuous space. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 5798–5810. Association for Computational Linguistics.

Shayne Longpre, Yu Wang, and Chris DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020, pages 4401–4411. Association for Computational Linguistics.

Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-supervised opinion mining with augmented data. In WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pages 617–628. ACM / IW3C2.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: self-supervised manifold based data augmentation for improving out-of-domain robustness. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 1268–1283. Association for Computational Linguistics.

Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le. 2020. Meta pseudo labels. CoRR, abs/2003.10580.

Mohammad Taher Pilehvar and José Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In NAACL-HLT.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. ArXiv, abs/1910.10683.

Jonathan Raiman and John Miller. 2017. Globally normalized reader. In EMNLP, pages 1059–1069. Association for Computational Linguistics.

Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In ICLR (Workshop).

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In AAAI/IAAI, Vol. 2, pages 1044–1049. AAAI Press / The MIT Press.

Corby Rosset, Chenyan Xiong, M. Phan, Xia Song, Paul N. Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. ArXiv, abs/2007.00655.

Timo Schick and H. Schutze. 2021. It's not just size that matters: Small language models are also few-shot learners. ArXiv, abs/2009.07118.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics.

Prashanth Vijayaraghavan, Ivan Sysoev, Soroush Vosoughi, and Deb Roy. 2016. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns. In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016, pages 413–419. The Association for Computer Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. arXiv preprint arXiv:1905.00537.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pages 2557–2563. The Association for Computational Linguistics.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification 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 2019, Hong Kong, China, November 3-7, 2019, pages 6381–6387. Association for Computational Linguistics.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip S. Yu. 2020. CG-BERT: conditional text generation with BERT for generalized few-shot intent detection. CoRR, abs/2004.01881.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. In NeurIPS.

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. 2020b. Self-training with noisy student improves imagenet classification. In CVPR, pages 10684–10695. IEEE.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In ACL, pages 189–196. Morgan Kaufmann Publishers / ACL.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. CoRR, abs/2104.08826.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In AAAI, pages 7402–7409. AAAI Press.

Adams Wei Yu, David Dohan, Minh-Thang Luong, R. Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. ArXiv, abs/1804.09541.

Xiang Yue and Shuang Zhou. 2020. PHICON: improving generalization of clinical text de-identification models via data augmentation. In Proceedings of the 3rd Clinical Natural Language Processing Workshop, ClinicalNLP@EMNLP 2020, Online, November 19, 2020, pages 209–214. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018a. mixup: Beyond empirical risk minimization. In ICLR (Poster). OpenReview.net.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018b. Record: Bridging the gap between human and machine commonsense reading comprehension. ArXiv, abs/1810.12885.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 649–657.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. In NeurIPS.

A Appendix

A.1 More Details about the PET Baseline Implementation

All experiments are carried out in a Linux environment with a single V100 GPU (32G). In order to run each experiment in a single GPU, we fix the bottom 16 layers' (bottom 1/3 layers) parameters of DeBERTa due to the limitation of GPU memory.

On ALBERT, all the parameters and patterns are kept the same as PET/iPET(Schick and Schutze, 2021). We find that the patterns on RTE give extremely poor results on DeBERTa, so we change the patterns of RTE on DeBERTa for a fair evaluation. Let's denote the hypothesis h and the premise p, the new pattern is "pQuestion:h?Answer:___.", while keeping the verbalizer the same as PET/iPET (maps "entailment" to "yes", "not entailment" to "no"). On DeBERTa, we also reduce the learning rate from 1e-5 to 5e-6 on RTE and WiC, which can improve the baseline a lot. Other settings are kept the same as in ALBERT.

We run each pattern and repetition with seed 42. Different from PET/iPET, to keep the order of the train data loader for different patterns, we will give the train data loader a seed of 10, 20, and 30 for three repetitions.

A.2 Details of Baseline Augmentation Methods

We compare our FlipDA with various data augmentation baseline methods. We do not choose some generation-based methods (Xia et al., 2020; Yoo et al., 2019; Li et al., 2019), because they usually need a lot of training data, which is not suitable for few-shot learning tasks. We also attempted to experiment with methods like LAMBADA (Anaby-Tavor et al., 2020) and GPT3Mix (Yoo et al., 2021). Because SuperGLUE tasks often involve dependency between sentence pairs, correlation between augmented sentences is necessary in order for the data to be meaningful. However, we were not able to generate well-formed, meaningful data from either LAMBADA or GPT3Mix. For example, in RTE, we want a premise and a shorter hypothesis that may be contained in the premise, but methods like GPT3Mix usually keep on generating long paragraphs in an uncontrollable manner. Moreover, these methods rely on priming, which is not suitable for datasets with long sentences.

Synonym Replacement (SR) (Zhang et al., 2015) augments data by randomly choosing $r\%$ words from original texts (stop words excluded), and replacing them with synonyms from WordNet[4]. Our implementation is based on parts of the code of EDA [5]. We fix the word replacement ratio to 0.1. We augment 10 times for each sample and then mix them with original samples copied for 10 times.

KNN Replacement (KNN) (Wang and Yang, 2015) is similar with Synonym Replacement but differs in replacing randomly-chosen-words with one of the nearest words derived from GloVe[6]. Our implementation is based on parts of the code of TinyBert [7]. We fix the word replacement ratio to 0.1, and we replace each word with one of the closest 15 words (K=15) derived from GloVe. We use the word embedding version with 300 dimensions and 6 billion words. We augment 10 times for each sample and then mix them with original samples copied for 10 times.

Easy Data Augmentation (EDA) (Wei and Zou, 2019) mixes outputs from four data augmentation methods, including synonym replacement, random insertion, random swap, and random deletion. Our implementation is based on the code of EDA [5], which removes all punctuations. Here we implement a new version with punctuation marks since we find them important for hard tasks. All hyper-parameters are kept default, i.e., the four augmentation methods are all with a ratio of 0.1, and each example is augmented 9 times. Finally, we will mix the augmented data with the original data as is done in (Wei and Zou, 2019).

[4]https://wordnet.princeton.edu/
[5]http://github.com/jasonwei20/eda_nlp
[6]https://nlp.stanford.edu/projects/glove/
[7]https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/TinyBERT

Back Translation (BT) (Fadaee et al., 2017; Sennrich et al., 2016) translates each text into another language, and then back translates into the original language. We implemented two versions of BT with google translator. The first one is BT-10, in which we get the augmented data with 9 languages (Spanish, French, German, Afrikaans, Russian, Czech, Estonian, Haitian Creole, and Bengali) and then mix it with the original sentences. The second one is BT-6, in which we get the augmented data with 5 intermediate languages (Spanish, French, German, Russian, and Haitian Creole) and then mix it with the original sentences.

TinyBERT (T-BERT) (Jiao et al., 2019) generates augmented data by randomly (with probability p) replacing each token with either word predicted by a Bert-base-cased model (for single-piece word) or words derived by GloVe (for multiple-piece word). Our implementation is based on the code of TinyBert [7]. If the sentence length is above 512, we will cut off the sentence. All parameters are kept default. Finally, we mix the augmented data with original examples in equal quantities.

T5-MLM. We randomly (with probability p) masks some tokens, and then fills in the blanks with a large pretrained model. We use pattern-based data cloze to further improve its performance. This is the same as FlipDA with only label-preserved data and without data selection. You can refer to Appendix A.3 for more details. We augment with a mask ratio of 0.1 because we find a smaller mask ratio will be better without classification. We augment 10 times for each sample and then mix them with original samples copied for 10 times.

MixUP (Zhang et al., 2018a; Guo et al., 2020) augments data in the feature space, which linearly interpolates between two source sentence embeddings, and correspondingly linearly interpolates the two target embeddings. For each batch, we first sample $\lambda = Beta(0.5, 0.5)$, just as the author (Zhang et al., 2018a) recommended. Then, we do linear interpolation on the embedding space of two sentences, and make it the input of the model. Finally, we calculate the loss as the interpolation between its outputs and the two targets.

A.3 Details of Pattern-based Data Cloze Strategy

Because the target and the format of tasks in FewGLUE vary a lot, it is necessary for us to adjust the details for data augmentation for each dataset. We will always keep the same framework: (1) firstly, mask the sentence, (2) secondly, generate the new label (preserve or flip the label), and (3) finally fill in the blanks by T5. We also augment 10 times for each example as the candidates. (Augmenting with more times might help, but we only augment 10 times for the sake of time, and we have shown its effectiveness.)

The T5 model (Raffel et al., 2020) is not perfect, especially when it is not finetuned. During our experiments, we find it a good cloze model (good at filling in the blanks with information before or after the blanks) but not a good generation model (not good at generating meaning that is not in the original sentence). As a result, in some tasks whose sentence is short, we induce the T5 model to get some new information by adding extra sentences from other examples in the training data set.

BoolQ. Each example contains two sentences, a question q and a passage p. We need to tell whether the answer of the question is True. Let's denote the masked question $masked\_q$ and the masked passage $masked\_p$. If we want to get a True answer, we will feed "$masked\_q$?Yes, $masked\_p$" into the model. Otherwise, we will feed "$masked\_q$?No, $masked\_q$" into the model. The T5 model will fill the blanks in the masked sentences.

CB. Each example contains two sentences, a premise p and a hypothesis h. We need to tell the relationship between the premise and the hypothesis, entailment, contradiction, or neutral. Let's denote the masked premise $masked\_p$ and the masked hypothesis $masked\_h$. We will feed "“$masked\_h$” ?___. “$masked\_p$”" into the model. Similar to PET, the verbalizer maps "entailment" to "Yes", "contradiction" to "No" and "neutral" to "Maybe". The T5 model will fill the blanks in the masked sentences.

COPA. Each example contains a premise p and two choices c_1, c_2. We need to tell which one is the cause or effect of the premise. The sentences in the COPA dataset is much shorter than the others, and the relationship between the three sentences is much more difficult to be represented in one sentence. So we only masked the premise p into $masked\_p$. When we flip the label, we want to make the opposite choice the label, and we also change the question with probability 0.5. If the new question is "effect", we will feed "$masked\_p$ so that c_{new_la}" into the model. Otherwise, we will feed "$masked\_p$, because c_{new_la}" into the model. Here new_la denotes the new label.

RTE. Each example contains two sentences, a premise p and a hypothesis h. Our augmentation policy is same as BoolQ. Let's denote the masked hypothesis $masked\_h$ and the masked premise $masked\_p$. If we want to get a True answer, we will feed "$masked\_h$?Yes, $masked\_p$" into the model. Otherwise, we will feed "$masked\_h$?No, $masked\_q$" into the model. The T5 model will fill the blanks in the masked sentences.

WiC. Each example contains two sentences $s1$ and $s2$, and we need to tell whether the word "w" in them has the same meaning. If the new label is "same", we will feed "$masked\_s1$. $masked\_s2$. Word " w " means the same in the two sentences" into the model. Sadly, we find if we concatenate them together with a large mask ratio, after filling in the masks they will be similar. This is because the two sentences are too short and T5 is not "imaginative" enough. To solve this problem, if the new label is "different", we will augment each sentence separately. We also add one sentence sampled from the training set to urge it to generate a more diverse representation. We still do not find a perfect way to augment because if a word does not have several meanings, it will be nearly impossible to flip its label from "same" to "different". We are happy to see that our method can still benefit the model a lot even though it is far from perfect.

WSC. In our experiments, we find it hard for T5 to generate new entities. In this paper, we do not flip its label, but we do believe that there exists an automatic way to generate good augmented examples with different labels.

MultiRC. Each example contains a passage p, a question q, and several candidate answers a. For each answer, it will have a label la. Our method is somewhat limited in this task, because it has been "fliped" when it is constructed. For the $< p, q, a >$ with label True and $< p, q, a' >$ pair with label False, they have satisfied our key idea: similar but different label examples. Even though, we still try to flip it more. Let's denote the masked question $masked\_q$, the masked passage $masked\_p$, and masked answer $masked\_a$. We fill feed "$masked\_q$? Is the correct answer "$masked\_a$"?Yes/No. $masked\_p$" into the model.

ReCoRD Each example contains a passage p, a question q, several candidate entities es, and several possible answers as. We fill first replace the "@placeholder" in the question q with new answer a', which is randomly sampled from es in the "flip" version and otherwise is sampled from as. Let's denote the masked question $masked\_q$ and the masked passage $masked\_p$. We will feed "$masked\_q$. $masked\_p$" into the model. Finally, we will substitute the new answer a' in the generated question with "@placeholder".

A.4 Details of Pattern-based Filling-in Strategy

We conclude three essential factors for the filling-in strategy: the mask ratio, the decoding strategy, and the fill-in strategy. We divide the mask ratio into three levels: 0.3 (small), 0.5 (medium), and 0.8 (large). The decoding strategy consists of greedy search, random sampling (sample from top 15 words), and beam search (with a beam size of 10). The fill-in strategy consists of filling in the blanks at a time or filling in k blanks at a time iteratively. Our experiments show that the mask ratio is the key factor.

A.5 Hyper-parameter Search Space of FlipDA

We do not search all the possible parameters to save time and avoid overfitting. We are not surprised if there are some better results with a larger search space. Our search space is listed in Table 11.

We did preliminary experiments and found some guiding principles. We find that datasets with larger sentence lengths should have a smaller mask ratio, and respectively, datasets with smaller sentence lengths should have a larger mask ratio. (The WSC dataset should be considered separately because we do not flip its label.) We also find that if the sentence length is too large, such as MultiRC or ReCoRD, it is impossible to fill in all the blanks at a time, because the number of blanks may exceed 100. To solve this problem, we fill in 10 random blanks at a time, iteratively until all masks are filled. What's more, the COPA dataset is too short, so we also try to fill in 1 random blank at a time, iteratively until all masks are filled. We do not figure out the relationship between the characteristic of the datasets and the decoding strategies, so we search the three decoding strategies for all datasets. For most of the datasets, greedy or sample is better than beam search. For each dataset, we also try two modes: allowing the classifier to change the label or not. (Augmented candidates that are predicted with a different label from the original ones could be dropped.) Above all, for most of the datasets, we only search 6 hyper-parameter combinations, we think this will not lead to severe overfitting, and our algorithm is stable.

A.6 Additional Discussion

Limitations for the WSC Task As is illustrated in the body part, label-flipped augmentation has inspiring advantages for few-shot learning performance, but it also has limitations. While FlipDA significantly outperforms existing baseline augmentation methods on most tasks, we also notice that its effect on the WSC task is a little behind some of the baselines. This is because, for the WSC task that disambiguates multi-token word senses, it is hard for T5 to generate its label-flipped cases. The T5 model is not good at making up similar entities that are not in the original sentence, and thus unable to produce desired candidate examples. We leave a better pattern-based cloze algorithm for such tasks to the future work. We anticipate that entity-centric pretrained models might alleviate this issue (Rosset et al., 2020).

Which Few-shot Setting to Use? Until now, it still remains an open problem of how to evaluate the performance of few-shot learning. Currently, there are mainly two mainstream few-shot settings. The first is to use a set of pre-fixed hyper-parameters that are determined according to practical consideration. The second is to construct a small dev set (e.g., a 32-sample-dev set), and then perform grid search and use the small dev set for hyper-parameters and model selection. Our experiments are based on the former setting. We respectively performed preliminary experiments using both settings and found that the first setting tends to be relatively more stable. We believe how to evaluate few-shot learning systems is an important research direction for future work, too.

A.7 More Results on DeBERTa

More Results on DeBERTa are in Table 12 and Table 13.

A.8 More Results on ALBERT

In the body part, we only report the ablation results on DeBERTa because the model is larger and seems more stable in our experiments. In this section, we report ablation results on ALBERT. Most of the conclusions are the same, but some details vary. We conjecture that this might be due to the instability of the training process, the quality of the classification model, or some other unknown issues.

Table 11: Hyper-parameter search space of our algorithm.

Dataset	Mask Ratio	Fill-in Strategy	Decoding Strategy
BoolQ	0.3/0.5	default	greedy/sample/beam search
CB	0.5	default	greedy/sample/beam search
COPA	0.8	default/rand_iter_1	greedy/sample/beam search
RTE	0.5	default	greedy/sample/beam search
WiC	0.8	default	greedy/sample/beam search
WSC	0.3	default	greedy/sample/beam search
MultiRC	0.3/0.5	rand_iter_10	greedy/sample/beam search
ReCoRD	0.3	rand_iter_10	greedy/sample/beam search

Table 12: Ablation study on label-flipped data v.s. label-preserved data on DeBERTa-v2-xxlarge. Bold denotes the best-performed results. Underlines denote the second-best results. "Avg." is the average of scores and "MD" (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	CB Acc./F1	COPA Acc.
Baseline	85.42/79.31	87.67
FlipDA (both)	**88.24/87.94**	**90.83**
Label-Flipped	<u>84.52/80.99</u>	<u>89.67</u>
Label-Preserved	83.48/78.68	87.67

Table 13: Results of different strategies for choosing augmented data on DeBERTa-v2-xxlarge. "Avg." is the average of scores and "MD" (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	CB Acc./F1	COPA Acc.
Baseline	85.42/79.31	87.67
Noisy Student	86.31/82.60	84.33
Default Strategy	88.24/87.94	**90.83**
Global TopP	88.10/85.59	89.33
Global TopK	88.54/85.69	87.83
Diverse TopK	**89.73/88.92**	90.0

A.8.1 Effectiveness of Pattern-based Data Cloze and FlipDA Classifier

From Table 14 we can see that FlipDA is still better than other baselines with a classifier, which means our pattern-based data cloze method will contribute to higher quality data with kept/flipped data. From the comparison between Table6 and Table 14, we can see that the classification is much more useful for DeBERTa than ALBERT. With DeBERTa, almost all augmentation methods will improve their performance with the classifier. With ALBERT, only some augmentation methods will improve its performance on some tasks. This is normal because a better classifier will lead to better classification results, i.e., better-selected augmentation data.

A.8.2 Analysis of Label-Flipping v.s. Label-Preservation

From Table 15, we can see that FlipDA is still the best, i.e., augmentation with both directions is better than with only one direction. Augmentation with only label-flipped data is better than with only label-preserved data in most tasks. This phenomenon is more obvious with DeBERTa than ALBERT, which may be because the classifier quality of DeBERTa is better than ALBERT. What's more, DeBERTa has learned better representations of similar phrases, so the label-kept examples will contribute less when we experiment with DeBERTa.

A.8.3 Analysis of Label Transformation

We took a closer at the effect of label transformation direction in Table 16. On BoolQ and RTE, the two flipped directions are better than the kept directions. On all datasets, adding data with more directions is better than with only one direction, even some direction seems extremely bad. This is the same as what we observed with DeBERTa.

A.8.4 Analysis of Strategies for Augmented Data Selection

From Table 17, we can see that Noisy Student performs well with the ALBERT model. It achieves good results on almost all the datasets except COPA. While with DeBERTa (see Table 9), the Noisy Student is somewhat weaker. This may be because the DeBERTa model fixes the bottom 1/3 layers' parameters to save Video Memory, and thus is not suitable for the perturbation on the embedding space. We have chosen the spatial dropout to alleviate the problem, and it will be much worser with other kinds of dropout. We think a better self-training policy could further improve the performance of data augmentation. All other observations of the effectiveness of different strategies are somewhat similar to that with DeBERTa.

A.9 Case Study

We have provided some flipped augmented examples on RTE in Table 10. Here we provide two kept cases on RTE and more augmented examples on other tasks, to be specific, BoolQ, WiC, and COPA.

Table 14: Ablation study on methods of obtaining candidate augmented data. The ablation study is based on ALBERT-xxlarge-v2. "cls" denotes the same classifier as FlipDA for filtering candidate augmented data. Bold denotes the best-performed ones. Wave-lines denotes those that outperforms the original (without FlipDA classifier) version.

Method	BoolQ Acc.	CB Acc./F1	COPA Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a	Avg.	MD
Baseline	72.47	82.74/74.84	88.33	61.40	51.27	33.04/74.64	67.68	-
SR + FlipDA cls	74.32	84.52/79.32	82.17	63.93	49.56	34.53/74.52	67.74	6.16
KNN + FlipDA cls	71.88	84.52/76.83	83.17	67.39	53.10	31.62/73.92	68.16	5.16
EDA + FlipDA cls	74.16	84.52/78.92	83.00	60.41	50.49	34.22/75.52	67.44	5.33
BT-10 + FlipDA cls	73.37	83.04/74.19	85.00	63.12	51.36	34.60/74.69	67.69	3.33
BT-6 + FlipDA cls	73.26	80.06/68.59	86.83	61.46	51.72	34.49/76.05	67.14	4.46
T-BERT + FlipDA cls	74.44	80.80/73.51	84.33	65.40	50.19	33.75/74.31	67.59	4.00
FlipDA	**76.98**	**86.31/82.45**	**89.17**	**70.67**	**54.08**	**36.38/76.23**	**71.93**	**0.00**

Table 15: Ablation study on label-flipped data v.s. label-preserved data on ALBERT-xxlarge-v2. Bold denotes the best-performed results. Underlines denotes the second-best results. "Avg." is the average of scores and "MD" (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	BoolQ Acc.	CB Acc./F1	COPA Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a	Avg.	MD
Baseline	72.47	82.74/74.84	88.33	61.40	51.27	33.04/74.64	67.68	-
FlipDA(both)	**76.98**	**86.31/82.45**	**89.17**	**70.67**	**54.08**	**36.38/76.23**	**71.93**	**0.00**
Label-Flipped	75.09	81.40/73.31	86.33	67.78	53.81	32.47/74.67	68.99	2.00
Label-Preserved	73.95	81.25/74.95	87.17	64.98	51.03	34.07/74.81	68.27	1.16

Table 16: Results of different label transformation on ALBERT-xxlarge-v2. RTE: A/B denotes entail/not-entail, indicating whether the given premise entails with the given hypothesis. BoolQ: A/B denotes False/True, representing the answer for the given yes-no questions. WiC: A/B refers to F/T, indicating whether the target word shares the same meaning in both given sentences.

Method	BoolQ Acc.	RTE Acc.	WiC Acc.
baseline	72.47	61.40	51.27
A→A	71.11	63.09	51.15
A→B	73.56	**66.71**	51.29
B→B	71.63	59.57	**52.61**
B→A	**74.36**	65.34	49.29

The four datasets cover tasks with different targets and sentence lengths.

RTE Two kept cases are in Table 18. In the first case, we can see that the T5-model changes the name of the tropical storm from "Debby" to "Maria", and it also changes the "tropical storm" to its hypernym "hurricane", and all these changes contribute to a different expression without affecting its label. The second case changes the future tense to the simple past tense, and it also changes "April" to "March" and "May" to "April" correspondingly. We can see that the way to change or keep the label is rich and natural.

WiC is a task to tell whether the word w in the two sentences has the same meaning. From Table 19, we can see that the two augmented sentences with direction to "True" is similar. This is determined by the characteristic of T5. In the second case, "feel" in "feel the gravity" means "perceive by a physical sensation", but in "felt so insignificant" means "have a feeling or perception about oneself in reaction to someone's behavior or attitude". The last example violates common sense, but it still can preserve the label and provide diversity, and thus boosting model performance.

BoolQ is a QA task that provides a passage and a question. The author needs to tell whether the answer to the question is True or False according to the given passage. We provide augmented examples of four directions. The augmented examples are in Table 20. The first case changes "green onyx" to "Brazilian onyx" without changing its label. The second case changes the passage to make the question True, even though it violates common sense. The third case copies some parts of the passage into the question, and then the label flips. The last case changes the keywords of the example but without changing its label.

COPA is a task that needs to choose the effect or cause of the premise from choice1 and choice2. PET treats it as a multi-token cloze question, i.e., predict the whole sentence of choice1 or choice2. We only change the premise or the question to flip or keep the label. The augmented examples are in Table 21. As described in Appendix A.3, there will be three types: keep the label, flip the label but keep the question, and flip the label and the

Table 17: Results of different strategies for choosing augmented data on ALBERT-xxlarge-v2. “Avg.” is the average of scores and “MD” (MaxDrop) measures the maximum performance drop over multiple tasks for a given method. All results are the the average over multiple patterns and 3 iterations.

Method	BoolQ Acc.	CB Acc./F1	COPA Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a	Avg.	MD
Baseline	72.47	82.74/74.84	88.33	61.40	51.27	33.04/74.64	67.68	-
Noisy Student	**78.01**	88.39/83.32	82.67	69.52	54.62	**37.02/76.53**	71.24	5.66
Default Strategy	76.98	86.31/82.45	**89.17**	**70.67**	54.08	36.38/76.23	**71.93**	**0.00**
Global TopP	77.73	**88.54/84.88**	87.50	67.30	54.30	35.47/76.47	71.59	0.83
Global TopK	76.86	87.50/84.42	85.33	69.43	51.97	36.48/75.36	70.91	3.00
Diverse TopK	77.27	88.39/83.18	88.67	70.61	**55.28**	32.40/73.64	71.77	0.82

Table 18: Some augmented examples selected by our model (DeBERTa) in RTE. Black denotes original examples, and blue denotes augmented examples.

Entailment → Entailment	**Premise:** Tropical Storm Debby is blamed for several deaths across the Caribbean. **Hypothesis:** A tropical storm has caused loss of life. **Premise:** Tropical Storm Maria is blamed for the deaths across the Caribbean **Hypothesis:** A hurricane has caused loss of life
Not Entailment → Not Entailment	**Premise:** The 10-men team is expected to arrive at the foot of the mountain in the end of April and began their journey to the 8,586-meter peak in early May. **Hypothesis:** Kanchenjunga is 8586 meters high. **Premise:** The 10-men team arrived at the foot of the mountain at the end of March and reached their goal of reaching the 8,586-meter peak in early April **Hypothesis:** Kanchenjunga is 8586 meters

question at the same time. The first case changes “ the archeologist” to “she” and “site” to “earth”, both of them keep the meaning of the sentence. The last three cases change almost the whole sentence, but they are in line with human knowledge.

Table 19: Some augmented examples selected by our model (DeBERTa) in WiC. Black denotes original examples, and blue denotes augmented examples. Underlines denotes the word to be determined.

True → True	**Context 1:** We vaccinate against scarlet fever. **Context 2:** The nurse vaccinated the children in the school. **Context 1:** We vaccinate the children against fever and malaria **Context 2:** The nurse vaccinated the children against fever and malaria
True → False	**Context 1:** You make me feel naked. **Context 2:** She felt small and insignificant. **Context 1:** You can feel the gravity **Context 2:** She felt so insignificant and useless
False → True	**Context 1:** Can you back up your claims? **Context 2:** I can't back this plan. **Context 1:** Can you please back to your home **Context 2:** I can't back from your house
False → False	**Context 1:** Turn and face your partner now. **Context 2:** The bunkers faced north and east, toward Germany. **Context 1:** Get up and face it now **Context 2:** The ship faced north and south from the coast

Table 20: Some augmented examples selected by our model (DeBERTa) in BoolQ. Black denotes original examples, and blue denotes augmented examples.

True → True	**Passage:** Onyx – Brazilian green onyx was often used as plinths for art deco sculptures created in the 1920s and 1930s. The German sculptor Ferdinand Preiss used Brazilian green onyx for the base on the majority of his chryselephantine sculptures. Green onyx was also used for trays and pin dishes – produced mainly in Austria – often with small bronze animals or figures attached. **Question:** is there such a thing as green onyx **Passage:** Onyx is Brazilian Onyx which was often used as the base for art glass sculptures created in the 1920s and 1930s . The German sculptor Ferdinand von Goethe used onyx as the base on the bases of his sculptures . It was also used for making pin plates and pin dishes and many artists produced on-oniex sculptures with various animals and figures attached **Question:** Is there such a stone as Brazilian onyx
True → False	**Passage:** Atomic number – The atomic number or proton number (symbol Z) of a chemical element is the number of protons found in the nucleus of an atom. It is identical to the charge number of the nucleus. The atomic number uniquely identifies a chemical element. In an uncharged atom, the atomic number is also equal to the number of electrons. **Question:** is the atomic number equal to the number of protons **Passage:** Atomic number is not equal to atomic number or protons. Atomic number (A, B, C, Z) of a chemical element is the number of electrons in the nucleus of an atom . The nucleus is composed by the electrons that are present in the nucleus . The numeric value uniquely identifies an element . In case of uncharged atom , the charge number is equal to the number of electrons **Question:** number of the atomic element equal to the number or protons
False → True	**Passage:** Peace bond – The use of peace bonds is rather uncommon in the U.S. justice system, but a deferred prosecution has a similar effect. Since there is no conviction or admission of any guilt, signing a peace bond in Canada does not usually result in U.S. inadmissibility under INA § 212 (a) (2). **Question:** is a peace bond an admission of guilt **Passage:** Peace bond is an important use of money that is widely used in the U.S. justice system , and deferred prosecution has similar effect . Since there is no promise or admission of guilt in any case , signing a peace bond does not usually result in any conviction under U § 2 (a) (b) **“question”**: Is a peace bond part of the criminal justice system
False → False	**Passage:** The Princess and the Goblin (film) – The Princess and the Goblin (Hungarian: A hercegnő és a kobold) is a 1991 British-Hungarian-American animated musical fantasy film directed by József Gémes and written by Robin Lyons, an adaptation of George MacDonald’s 1872 novel of the same name. **Question:** is the princess and the goblin a disney movie **Passage:** The Goblet and the Goblin (film) – The Hound and the Goblin (Hungarian : A hoz és a kobold) is a 1996 British-Hungarian-American film directed by Peter Gémes and produced by John Lyons , an adaptation of George MacDonald ’s novel of the same name **Question:** Is the goblin and the hobbit disney movie

Table 21: Some augmented examples selected by our model (DeBERTa) in COPA. In this task, we only change the premise or question to flip/keep the label. Black denotes original examples, and blue denotes augmented examples.

Keep-label	Keep-question	**Alternative 1:** She excavated ancient artifacts. **Alternative 2:** She read about the site's history. **Premise:** The archeologist dug up the site. **Question:** "effect" **Correct Alternative:** 0 **Premise:** She dug up the earth. **Question:** Effect **Correct Alternative:** 0
	Keep-question	**Alternative 1:** She began going to church. **Alternative 2:** She began travelling abroad. **Premise:** The woman had a religious awakening. **Question:** Effect **Correct Alternative:** 0 **Premise:** She had a lot of money. **Question:** Effect **Correct Alternative:** 1
Flip-label	Flip-question (Effect → Cause)	**Alternative 1:** Her friend sent her a greeting card. **Alternative 2:** Her friend cut off contact with her. **Premise:** The woman betrayed her friend. **Question:** Effect **Correct Alternative:** 1 **Premise:** A woman is happy. **Question:** Cause **Correct Alternative:** 0
	Flip-question (Cause → Effect)	**Alternative 1:** The cafe reopened in a new location. **Alternative 2:** They wanted to catch up with each other. **Premise:** The women met for coffee. **Question:** Cause **Correct Alternative:** 1 **Premise:** The cafe closed. **Question:** Effect **Correct Alternative:** 0