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

Jing Zhou *
Tsinghua University †
zhouj18@mails.tsinghua.edu.cn

Yanan Zheng *
Tsinghua University ‡ & BAAI§
zyanan@mail.tsinghua.edu.cn

Jie Tang
Tsinghua University‡ & BAAI§
jietang@tsinghua.edu.cn

Jian Li ¶
Tsinghua University †
lijian83@mail.tsinghua.edu.cn

Zhilin Yang ¶
Tsinghua University †
zhiliny@mail.tsinghua.edu.cn

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 tradeoff between effectiveness and robustness—it substantially improves many tasks while not negatively affecting the others. [6]

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 [75, 62, 65, 30, 63]. Generative modeling methods such as back-translation have also been employed to generate augmented samples [13, 57]. However, there are two major limitations of the previous studies. 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 [41]. Second, most of the previous studies are carried on simple tasks such as single-sentence classification where it is easier to generate legit augmented

*The authors have contributed equally to this work.
†Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua Univeristy, Beijing, China
‡Department of Computer Science and Technology, Tsinghua University, Beijing, China
§Beijing Academy of Artificial Intelligence, Beijing, China
¶Corresponding Authors.
[6]Our source code is available at https://github.com/zhouj8553/FlipDA

Preprint. Under review.

samples. For harder tasks such as natural language understanding (e.g., telling whether sentence A entails sentence B), it is not clear whether previous methods still help.

In this work, we take a step further to study data augmentation under strong baselines and hard tasks. Our study employs large-scale pretrained language models such as DeBERTa [20] with over one billion parameters as baselines. Moreover, we target a very challenging setting—few-shot natural language understanding (NLU). We consider challenging NLU tasks including question answering, textual entailment, coreference resolution, and word sense disambiguation. We adopt SuperGLUE which was constructed to include some of the most difficult language understanding tasks for current NLP approaches [59]. Following [56], we used only 32 training examples to construct a few-shot setting. Under this setting, we reproduced a large number of widely-used prior methods for data augmentation. Our experiments lead to two unexpected discoveries: (1) most of previous 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. During preliminary experiments, we observed 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 [48] and uses a classifier to select label-flipped data. Experiments demonstrate that 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. Data augmentation aims to increase the amount of training data, and then utilize it to improve the performance of the model. A wide range of augmentation methods are based on word substitution, such as synonym replacement [75], KNN replacement [62, 58], unif replacement [65], tf-idf replacement [65], bi-rnn replacement [30] etc. Synonym replacement and KNN replacement are the most popular among them. Entity replacement is useful in question answering [49], opinion mining [43], and entity detection [72] tasks. EDA [63] combines four simple augmentation methods (i.e., synonym replacement, random deletion, random swap, and random insertion) and achieves good results on several single sentence classification tasks. Back translation [13, 57] is widely used and becoming standard practice in machine translation tasks. Back translation is also adapted to question answering [71] tasks. Pitifully, EDA and back translation methods are shown to be less useful with large pretrained models [41].

Some augmentation methods are based on the perturbation in the feature space, for example, mixup [73, 17], Mixtext [3], LADA [2], Snippext [43], and other augmentation methods in the feature space [32]. However, we observed limited improvement using these methods under our setting, suggesting that feature space augmentation might not be very much compatible with pretrained models and hard NLU tasks.

Various generation-based augmentation methods have been proposed. Variational autoencoders [64, 38, 55, 70] and denoising autoencoders [44] are applied to generate augmented examples, but these methods usually rely on a large amount of training data, which is not suitable for few-shot learning tasks. Along this direction, CRQDA [40] trains a model to generate answerable and unanswerable questions. A sequence-to-sequence data augmentation method [22] is proposed for dialogue language understanding. In contrast to our approach, these methods are more task-specific and cannot be directly applied to general NLU tasks.

In addition, large pretrained models have been used for data augmentation. [31] utilize large pretrained models, such as GPT-2, BERT, and BART, for conditional data augmentation. LAMBADA [1] finetunes a GPT-2 model with the priming technique to get augmented examples. GPT3Mix [69] uses GPT-3 along with prompting to generate augmented data for classification tasks. Our method is similar to this line of work in that we also use pretrained models for generating augmented data. However, there are the following key differences. First, it is challenging for these prior methods to

handle long sequences or multiple sentences. In our preliminary experiments, we were not able to use these methods to generate proper data samples (see details in Section 4). Second, besides generating augmented samples, we found it crucial to use label-flipped data for augmentation, which is a unique and critical aspect of FlipDA.

Self-training. Self-training [24] is a semi-supervised learning algorithm, which labels unlabeled data with a trained model (teacher model), and then uses the labeled data and unlabeled data jointly to train a new model (student model). This process may repeat for a few iterations. Self-training is applied in a lot of fields, including sense disambiguation [68], pattern extraction [52], parsing [42, 23, 51], image classification [67], neural sequence generation [18], speech recognition [26], and so on. Knowledge distillation and pseudo-labeling are special forms of self-training [21, 34, 50]. [76] observed that different from pre-training, self-training is helpful with strong data augmentation. Noisy Student [66] obtained good results on ImageNet by using an equal-or-larger student model and adding noise to the student. [18] concluded that dropout is important for self-training in sequence generation tasks. [45] proposed to update the parameters of the teacher model using the feedback of the student's performance on labeled data.

Self-training bears similarity to the second phase of FlipDA where a teacher model is used to select samples. Different from previous self-training methods, FlipDA discovers and leverages the usefulness of label flipping to improve performance and does not rely on unlabeled data. Moreover, as shown in Section 4, the data selection strategy of FlipDA is superior to some of the baselines such as Noisy Student.

Adversarial Attacks. Adversarial attacks add small perturbations to the original inputs while not affecting the label of the original sample to fool the model. [39] proposed to select the most vulnerable words first and then utilize the BERT model to replace the words, which is followed by generating grammatically fluent and semantically consistent examples. [15] and [37] further utilize BERT to replace, insert, or merge words in the sentences for better attack performance. Adversarial attacks and FlipDA both generate samples in the neighborhood of the original samples, but their goals are very much different. Adversarial attacks aim at fooling the model with label-preserved samples, while FlipDA aims at improving the performance of tasks with label-flipped samples. In addition, adversarial attacks require generating high quality samples, while fluency is not a major concern for augmentation as long as it improves performance.

Label Flipping. Our manual label flipping augmentation procedure is analogous to [27] and [14]. [27] aimed to mitigate the effects of learning spurious features. [14] proposed to use manual label flipping to reduce systematic gaps in the dataset. In contrast, we target improving few-shot generalization via data augmentation. Moreover, we measure the performance on an existing i.i.d. distributed test set while [27] and [14] created more challenging/meaningful test sets. Most importantly, we propose an automatic method of label flipping, going beyond manual efforts.

3 Data Augmentation for Few-Shot Learning

3.1 Setting

In this work, we consider data augmentation on hard tasks and strong baselines.

Few-Shot NLU Tasks. Natural language understanding is a collection of tasks that require in-depth understanding of the input in order to obtain high performance. NLU tasks range from coreference resolution [35], causal reasoning [16], textual entrailment [9, 7], and word sense disambiguation [47] to question answering [5, 28, 74]. These tasks are usually formulated as mapping a sentence or multiple sentences to a certain label. In this work, for systematic evaluation, we adopt SuperGLUE which contains a set of natural language understanding tasks and is designed to benchmark progress on "difficult" language understanding capabilities for current NLP approaches [60]. Following [56], 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 [11, 33, 20] is available and few-shot learning is performed based on the pretrained model. This is a crucial setting because previous studies found that using a strong pretrained model as the baseline eliminates the benefits of data augmentation [41] while large pretrained models are becoming more and more available. Our main result is based on DeBERTa [20] with over one billion parameters, and we also provide additional results with ALBERT which has fewer parameters [33].

Preliminary Experiments with Prior Methods. Our preliminary experiments with a large number of previous methods lead to a conclusion that there is not an effective and robust method available for this hard setting. With the previous methods, the gains are limited while it is possible to enter a failure mode with substantial performance drop. More details will be discussed in Section 4. We will discuss how we tackle this challenge by proposing a novel data augmentation method FlipDA in the following 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 mainly consider two types of robustness in this work: (1) robustness w.r.t. different base pretrained models and (2) robustness w.r.t. different tasks.

In other words, we want a data augmentation method that improves some tasks while not hurting the others, so as to achieve strong performance in terms of both effectiveness and robustness.

3.3 Effectiveness: Manual Label Flipping Improves Performance

Table 1: Results of manual data augmentation. 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

Since previous methods are not sufficiently effective and robust in our preliminary experiments (see Tables 5 and 6 in Section 4 for more 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 in this study.

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.

Results are shown in Table 1. In this experiment, for each original example, we produce one label-flipped example and one label-preserved example. We combine the augmented data with the original data to train a model. Our experiments follow the original setting of PET/iPET[56], which is to 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. 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 previous methods on data augmentation focus on creating data examples that are assumed to have the same labels as the original examples. This contradiction might explain the observation that previous augmentation methods are not sufficiently effective for the hard few-shot setting.

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 last 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") **"hypothesis"**: "A case of smallpox was confirmed." ("not entailment")
CB	**"hypothesis"**: "even someone as sensible as Miss van Williamsburgh would try to make a play of this sort" **"premise"**: ""For such a person, finding a protector might not be so difficult, even in Edinburgh." Jean smiled. He might have known that even someone as sensible as Miss van Wiliamsburgh would try to make a play of this sort." ("entailment") **"premise"**:""For such a person, finding a protector might not be so difficult, even in Edinburgh." Jean smiled. Do you think that even someone as sensible as Miss van Wiliamsburgh would try to make a play of this sort?" ("neutral")
WSC	**"sentence"**: "The city councilmen refused the demonstrators a permit because they advocated violence." **"hypothesis"**:"The city councilmen refused the criminals a permit because they advocated violence."

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, which improves generalization in a few-shot setting.

There is a caveat about this manual augmentation experiment. Although we follow a certain principle (i.e., making minimal changes to alter the label) 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 in the following sections for objective evaluation and reproducibility. That said, the findings in this section motivate the core idea of FlipDA.

3.4 Robustness: What Contribute to Failure Modes?

We also use preliminary experiments to analyze why augmentation methods usually suffer from failure modes. Most augmentation methods are based on a label preserving assumption that the newly generated data samples have the same labels as the original ones. However, it is challenging for automatic methods to always generate samples that preserve the labels in a hard NLU setting.

We first examine the samples generated by previous automatic methods including EDA and KNN. Examples are shown in Table 4. In the first example, a keyword "rabies" is deleted by the augmentation method, which not only results in a grammatically incorrect expression but also eliminates the key information to support the hypothesis. In the second example, the entity name in the hypothesis "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.

To further verify the cause of failure modes, we experimented with EDA [63] on the RTE task in SuperGLUE [59]. Using EDA decreases the performance by a few percentage points with both ALBERT and DeBERTa, entering a failture mode. To analyze the reason, 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 these noisy samples. Note that 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 for NLU tasks, label flipping is usually asymmetric. We will discuss more of this phenomenon is Section 4.5.

As shown in Table 3, both operations of replacing and correcting noisy samples largely improve performance to prevent the failure mode. Moreover, correcting the labels brings large gains and results in favorable performance compared to the baseline without data augmentation. This indicates that the label-preserving assumption and grammatical errors contribute to failure modes, and label flipping tends to alleviate the issue.

Similar to experiments in Section 3.3, experiments in this section involve subjective factors like human judgement. We would like to reiterate that these experiments are merely meant to demonstrate the intuition and insights of FlipDA, rather than proving the superiority of FlipDA.

Table 3: Performance of correcting the wrong-labeled augmented data by EDA on RTE. Wrong-Deleted denotes replacing the wrong-labeled augmented samples with corresponding original samples, and Wrong-Flipped 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.

	ALBERT	DeBERTa
No DA	61.40	81.95
EDA	58.33	77.38
Wrong-Deleted	59.39	80.75
Wrong-Flipped	61.07	83.39

Table 4: Augmented example with wrong labels. The first is by EDA, and the second is by KNN. Black denotes original examples, and blue denotes augmented examples. 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") **"premise"**: "this case of in western newfoundland is the first case confirmed on the island since 1989." **"hypothesis"**: "a case of rabies was confirmed." ("not entailment")
"premise": "... The security vice-minister, Marcos Farfan, said that police have surveillance photographs of Mr Dwyer at various public events attended by Mr Morales, including a peasant rally near Santa Cruz and a visit to naval installations on Lake Titicaca ..." **"hypothesis"**: "Lake Titicaca has a naval installation." ("entailment") **"premise"**: "... the security vice - minister , marcos farfan , said that investigators have surveillance photographs of mr farrell at various public events hosted by mr morales , includes a peasant rally near santa cruz and a visit to naval installations on lake titicaca ..." **"hypothesis"**: "lake havasu has a naval installation ." ("not entailment")

3.5 FlipDA: Automatic Label Flipping

The above observations in Sections 3.3 and 3.4 show that a label flipping data augmentation method without the label-preserving assumption might be beneficial for few-shot NLU, in terms of both effectiveness and robustness. Moreover, reducing grammatical errors is also key to preventing failure modes. This motivates our development of the new method FlipDA that automatically generates and selects label-flipped data, which does not rely on the label-preserving assumption.

FlipDA mainly consists of the following steps, as shown in Figure 3.5:

1. Train a classifier (e.g., finetuning a pretrained model) without data augmentation
2. Generate label-kept and label-flipped augmented samples

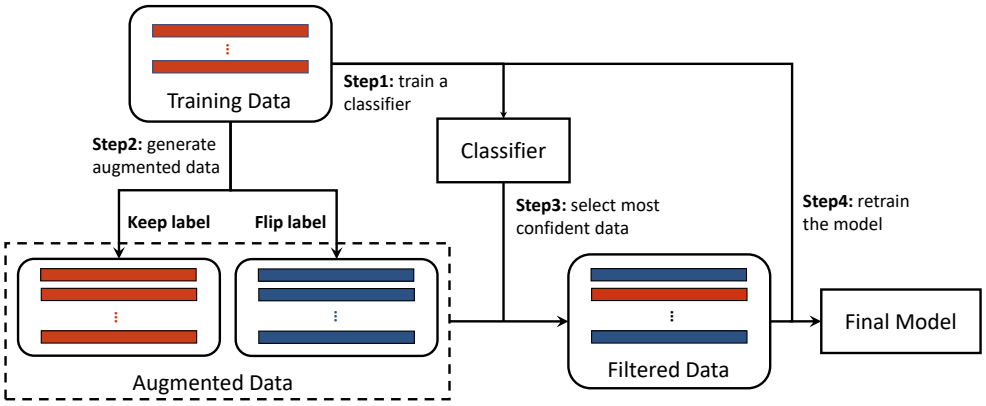

Figure 1: Training process of FlipDA. We first train a classifier with standard PET. And then, we generate augmented data with both kept/flipped labels. Thirdly, we utilize the trained classifier to filter the augmented data. Finally, we retrain the model with the selected augmented data and get a new model.

3. Use the classifier to select generated samples with largest probabilities for each possible label
4. Retrain the classifier with 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 $f(x, y) = \hat{p}(y|x)$. For example, the model f can be a pretrained BERT [11] or its variants [33, 11]. 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 [56] 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 [48] to fill the blanks to form new samples (see Appendix A.3 for more details). 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 $S_{i,y'}$. After 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

In this section, we conduct extensive experiments on the few-shot version of natural language understanding benchmark SuperGLUE [60] (also known as FewGLUE [56]). Results demonstrate that FlipDA is effective in promoting few-shot performance by generating label-flipped data while being robust towards different pretrained models as well as tasks.

4.1 Experimental Setup

Datasets Compared to tasks from other NLU benchmarks (e.g., GLUE [61]), most of which are single-sentence tasks, SuperGLUE consists of complicated NLU tasks that are all sentence-pair or sentence-triple tasks, which demand more understanding abilities. We conduct systematic experiments across 7 SuperGLUE tasks, ranging from question answering (BoolQ [6] & MultiRC [29]), textual entailment (CB [10] & RTE [8]), co-reference resolution (WiC [46]), causal reasoning (COPA [53]), and word sense disambiguation (WSC [36]). Each task consists of a 32-sample train set, a test set, a validation set, and an additional unlabeled set.

Baselines We compare our FlipDA with various data augmentation baseline methods. We do not choose some generation-based methods [64, 70, 38], 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 [1] and GPT3Mix [69]. 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.

We take seven augmentation methods as the baseline as follows. For more details about implementation please refer to Appendix A.2.

1. **Synonym Replacement** (SR) [75] augments data by randomly choosing $r\%$ words from original texts (stop words excluded), and replacing them with synonyms from WordNet[7].
2. **KNN Replacement** (KNN) [62] is similar with Synonym Replacement but differs in replacing randomly-chosen-words with one of the nearest words derived from GloVe[8].
3. **Easy Data Augmentation** (EDA) [63] mixes outputs from four data augmentation methods, including synonym replacement, random insertion, random swap, and random deletion.
4. **Back Translation** (BT) [13, 57] translates each text into another language, and then back translates into the original language[9].
5. **TinyBERT** (T-BERT) [25] 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).
6. **T5-MLM** [31] generates augmented data by randomly (with probability p) masking some tokens, and then filling in the blanks with a large pretrained model. We use pattern-based data cloze to further improve its performance. That is, it is the same as our FlipDA with only label-preserved examples and without data selection.
7. **MixUP** [73, 17] augments data in the feature space, which linearly interpolates between two source sentence embeddings, and correspondingly linearly interpolates the two target embeddings.

[7] https://wordnet.princeton.edu/

[8] https://nlp.stanford.edu/projects/glove/

[9] 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.

Evaluation Protocol To evaluate effectiveness, we use exactly the same metrics (e.g., accuracy, f1 score, and exact match) as PET/iPET [56]. PET is a prompt-based training framework, which converts all tasks into cloze problems, and substantially exceeds the previous sequence classification method. It has been proved that fine-tuning on small datasets would suffer from high variance and instability [12], which results in drastic changes even for minor changes in experimental conditions. Besides, PET/iPET [56] has also pointed out that the choice of patterns would influence much on the performance. To alleviate such influences, we run each experiment over multiple patterns and 3 iterations, and finally report the average performance.

Additionally, we also explore the robustness of FlipDA with respect to different pretrained models and tasks. We experiment on 7 different complicated tasks as mentioned above, and 2 pretrained language models of different scales, respectively ALBERT (ALBERT-xxlarge-v2) and DeBERTa (DeBERTa-xxlarge-v2). For robustness evaluation, we propose a new metric named MaxDrop (MD), which measures the maximum performance drop compared to not using augmentation over multiple tasks for a given method. In detail, with tasks t_1, t_2,..., and t_n, method med, and baseline base, MD is denoted as

$$\text{MD} = \max_{t \in \{t_1, t_2, \ldots t_n\}} \max(0, \text{score}_{t,\text{base}} - \text{score}_{t,\text{med}})$$

, where $\text{score}_{t,\text{med}}$ denotes the performance of method med on task t, and $\text{score}_{t,\text{base}}$ denotes the performance of method base on task t. Smaller values indicate the method is more robust towards various tasks, and vice versa.

We follow the same experimental setting as PET/iPET [56] where we take a set of fixed hyper-parameters. For ALBERT, we use exactly the same hyper-parameters as PET/iPET [56]. For DeBERTa, we select a set of fixed hyper-params according to practical considerations. Please refer to Appendix A.1 for details.

4.2 Main Results

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

Specifically, FlipDA achieves an average performance of 72.90 on ALBERT-xxlarge-v2 and an average of 78.65 on DeBERTa-v2-xxlarge, both of which outperform baselines by more than 3 points. The results strongly suggest that FlipDA is effective in boosting the performance of few-shot tasks by augmenting high-quality data without causing too many side effects.

We do not implement back-translation on WiC and WSC, because they both need to keep part of the words in the sentence unchanged, which can not be satisfied by back-translation. Meanwhile, we also observe that 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 that FlipDA is robust to different types of tasks, while other data augmentation methods could only be effective for partial tasks, and are not sufficiently robust.

4.3 Ablation Study of FlipDA

We observe that most of the data augmentation methods show effectiveness in terms of certain tasks or base models to a certain extent, while FlipDA can achieve good performance on almost all the tasks. Therefore, we are especially interested in **the essential reasons** that make FlipDA effective & robust. In this section, we show the ablation results with model DeBERTa-v2-xxlarge. You can refer to more results with ALBERT in Appendix A.6.

Generally, FlipDA has two steps, where the first step prepares the candidate augmented data and the second step selects data and flips labels. In the following experiments, we first freeze the second step to study variants the first step, and then fix the first step to study the second.

Effectiveness of Pattern-based Data Cloze From the perspective of obtaining candidate augmented data, there are several different types of methods, including replacement-based methods (e.g., KNN replacement and synonym replacement), generation-based methods (e.g., back translation), feature space based methods (e.g., mixup), and our pattern-based data cloze method (i.e., FlipDA). To

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	Avg.	MD
Baseline	72.47	82.74/74.84	88.33	61.40	51.27	77.03	33.04/74.64	69.02	-
SR	74.98	83.33/78.12	87.50	59.24	51.25	78.74	34.09/75.55	69.61	2.16
KNN	74.51	82.14/74.39	85.50	61.91	51.62	75.00	32.72/75.20	68.68	2.83
EDA	72.68	81.10/73.58	84.50	58.33	51.81	75.85	28.74/73.05	67.34	3.83
BT-10	74.59	82.44/77.72	83.00	55.93	-	-	32.96/74.69	-	5.47
BT-6	75.36	82.89/76.55	86.50	57.46	-	-	34.85/75.82	-	3.94
T-BERT	72.60	85.42/82.35	84.67	58.66	51.10	78.95	30.47/73.20	68.81	3.66
T5-MLM	73.86	83.48/75.01	87.33	62.27	51.08	**79.17**	33.79/74.06	69.55	1.00
MixUP	75.03	83.93/79.28	70.33	62.06	52.32	68.70	34.06/74.66	66.34	18.00
FlipDA	**76.98**	**86.31/82.45**	**89.17**	**70.67**	**54.08**	78.74	**36.38/76.23**	**72.90**	**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	Avg.	MD
Baseline	78.30	85.42/79.31	87.67	81.95	58.74	80.13	40.40/78.14	75.49	-
SR	77.37	87.20/80.28	87.00	76.29	58.88	80.88	35.70/76.25	74.30	5.66
+FlipDA cls	80.37	83.48/79.01	85.50	82.79	59.75	78.10	37.51/76.84	74.99	2.17
KNN	75.35	83.78/75.61	85.00	75.45	59.63	79.38	29.84/69.14	72.00	9.78
+FlipDA cls	78.51	87.50/82.53	88.33	82.79	58.66	76.39	38.86/77.29	75.40	3.74
EDA	74.42	83.63/76.23	85.83	77.38	59.28	78.74	37.02/77.05	73.23	4.57
+FlipDA cls	76.20	87.35/82.35	88.17	82.31	59.94	79.81	42.84/79.30	76.05	2.10
BT-10	75.38	88.24/84.03	85.33	79.66	-	-	38.88/77.79	-	2.92
+FlipDA cls	79.97	85.71/80.50	87.50	78.58	-	-	40.97/78.25	-	3.37
BT-6	76.78	86.46/82.56	84.00	81.47	-	-	40.53/79.01	-	3.67
+FlipDA cls	79.63	84.67/77.94	77.00	82.91	-	-	39.03/77.64	-	10.67
T-BERT	70.53	86.01/82.77	86.17	72.80	57.49	78.85	34.94/75.17	72.18	9.15
+FlipDA cls	80.24	86.16/81.25	83.00	82.19	59.49	79.59	40.78/78.64	75.42	4.67
T5-MLM	77.39	83.04/73.71	88.17	81.23	60.73	**82.37**	35.02/74.98	74.69	4.73
MixUP	63.41	71.13/60.83	72.00	68.59	57.70	68.38	39.24/76.88	64.87	16.39
FlipDA	**81.80**	**88.24/87.94**	**90.83**	**83.75**	**65.12**	78.85	**44.18/80.00**	**78.65**	**1.28**

find out the differences among them, we conduct massive ablation studies by feeding the candidate augmented data obtained by these methods into the same classifier (as FlipDA uses) and see if other strategies can also reach similar performance as FlipDA does. Table 6 shows the ablation results on three types of methods of obtaining candidate augmented data. Note that the feature space-based method is not included since this method can not be combined with a classifier.

FlipDA outperforms all the baseline methods with a classifier (i.e., with "FlipDA cls"). FlipDA achieves the largest average effectiveness scores as well as the smallest MaxDrop value (denoted in bold). When combining different strategies of obtaining candidate augmented data with FlipDA classifier still cannot reach similar performance to FlipDA, which proves that our pattern-based data cloze strategy with T5 is effective. There are several possible reasons. Augmentation based on T5 produces samples with less grammatical errors. This will be further discussed in Section **??**. Moreover, T5-style blank filling might produce samples that are more compatible with label flipping.

Effectiveness of FlipDA Classifier Now we compare the performance of different data augmentation methods with and without the FlipDA classifier. According to results in 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 are BT-6 and BT-10, for which the FlipDA classifier does not improve performance. This is reasonable since data augmented by back translation usually lack diversity. Using the FlipDA classifier further decreases diversity and hurts performance.

The improvement brought by the FlipDA classifier is more consistent on BoolQ, RTE, and MultiRC. This may be because these tasks involve predicting one single token with two opposite choices. As a result, 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

As indicated in Table 5 and Table 6, generating both label-flipped and label-preserved data at the same time lead to performance improvement. A follow-up question is how label-flipped data and label-preserved data respectively contribute to the overall performance. To answer this question, we conduct further analysis by running decoupling label-flipped data and label-preserved data. Results are presented in Table 7, where bold text represents the best-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. For more results on ALBERT please refer to Appendix A.6.2.

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.	CB Acc./F1	COPA Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a	Avg.	MD
Baseline	78.30	85.42/79.31	87.67	81.95	58.74	40.40/78.14	74.72	-
FlipDA (both)	**81.80**	**88.24/87.94**	**90.83**	**83.75**	**65.12**	**44.18/80.00**	**78.61**	0.0
Label-Flipped	<u>80.91</u>	<u>84.52/80.99</u>	<u>89.67</u>	<u>83.51</u>	<u>62.34</u>	<u>42.7/79.37</u>	<u>76.70</u>	0.0
Label-Preserved	77.04	83.48/78.68	87.67	80.99	60.08	39.55/78.30	74.30	1.28

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. Along this direction, it is even more intriguing to study

which direction of label flipping is able to benefit the few-shot performance to the maximum extent. To account for this, we experiment with 4 tasks, including RTE, BoolQ, WiC, and MultiRC, all of which are binary classification tasks. 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. Here we focus on using DeBERTa as the base model. For the results on ALBERT please refer to Appendix A.6.3.

Results are shown in Table 8. Different tasks can be roughly categorized into two types, asymmetric and symmetric. For BoolQ, RTE, and WiC, transforming in one direction is more beneficial than the other. This demonstrates an effect of asymmetric label transformation. For example, BoolQ aims to answer yes-no questions. It is relatively easy for a model to generate samples with answer "no" because one small conflict is sufficient for the prediction. However, in the reverse direction, it is difficult to generate sample with answer "yes" as the model has to analyze all details in an entire paragraph to ensure consistency. As a result, "hard-to-easy" might benefit from higher generation quality. The above reasoning is also applicable to RTE (i.e., generating non-entailment samples is easier than generating entailment samples) and WiC (i.e., generating sentences in which a word has the same meaning is easier). We observed similar effects for these three tasks that generating "hard-to-easy" samples is more beneficial. On the other hand, for MultiRC, the difference between the two directions are not significant. This may be because the two directions are similar in terms of generation difficulty.

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.

Table 8: Results of different label transformation on DeBERTa-v2-xxlarge. 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**

4.6 Analysis of Strategies for Augmented Data Selection

We also conduct quantitative analysis of the strategies for choosing the augmented data. There are several coupled attributes of augmented datasets that would drastically influence the few-shot performance simultaneously, including label distribution, diversity, data quality (since augmented data have noises), etc. And unfortunately, we find it hard to decouple each of them to see how each attribute will affect the overall performance.

Therefore, by comprehensively considering the coupled attributes, we propose four plausible strategies for augmented data selection, and quantitatively evaluate them. Our proposed four strategies for augmented data selection are described as follows.

1. **Default Strategy.** This is the strategy we described in Section 3.5. Each sample could have multiple candidate augmented samples. We will divide the candidate augmented samples into different categories by the classifier. For each direction, we will choose the one with the highest probability if there exists. This method is easy to be implemented and doesn't require any hyper-parameters.
2. **Global TopK**. All the candidate augmented data from all the original samples are gathered together and sorted by their predicted probabilities along a certain label transformation direction. The top-K samples with the highest predicted probabilities are selected as augmented data for the direction. This is equivalently implemented as selected top-$r\%$ augmented samples.

3. **Global TopP**. Similar to Global TopK, but with a different selection method. Augmented data with predicted probabilities higher than a threshold P are selected.
4. **Diverse TopK**. Similar to Global TopK 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 our data selection method can be viewed as a self-training algorithm, we also add a popular self-training algorithm Noisy Student [66] as another baseline in this section. We treat the augmented data as unlabeled data and add noise through dropout 0.1. We use spatial dropout in the embedding space.

Table 9 shows the results of different strategies on multiple tasks. For Global TopP, we set the threshold P at 0.9 or 0.95, whichever is better. For Global TopK and Diverse TopK, we select the top 10% or 20% augmented examples, whichever is better. Our strategies outperform Noisy Student because as it leverages the idea of label flipping as discussed in Sections 3.3 and 3.4. Among our four data selection strategies, the default strategy and Diverse TopK 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.

Table 9: 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	BoolQ Acc.	CB Acc./F1	COPA Acc.	RTE Acc.	WiC Acc.	MultiRC EM/F1a	Avg.	MD
Baseline	78.30	85.42/79.31	87.67	81.95	58.74	40.40/78.14	74.72	-
Noisy Student	**82.13**	86.31/82.60	84.33	82.79	64.11	39.99/77.43	76.09	3.34
Default Strategy	81.80	88.24/87.94	**90.83**	83.75	65.12	**44.18/80.00**	**78.61**	**0.00**
Global TopP	81.22	88.10/85.59	89.33	81.11	64.19	42.56/79.16	77.26	0.84
Global TopK	80.71	88.54/85.69	87.83	81.35	**65.13**	41.14/78.52	76.99	0.60
Diverse TopK	**81.99**	**89.73/88.92**	90.0	**84.59**	63.85	42.64/79.13	**78.44**	**0.00**

4.7 Case Study

We provide several selected augmented cases on the RTE dataset by our model in Table 10 to show the augmented sample quality of our method. We provide cases in four scenarios:"entailment" $\rightarrow$ "entailment", "entailment" $\rightarrow$ "not entailment", "not entailment" $\rightarrow$ "entailment", and "not entailment" $\rightarrow$ "not entailment".

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 adds "not" to the premise, and as a result, the label flips. The third case changes "dwindles" to its antonym "increased", and then the label changes from "not entailment" to "entailment". The fourth 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. 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.

More augmented examples please refer to Appendix A.7.

5 Discussion

Limitations for the WSC Task As is illustrated above, 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

Table 10: 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"
"entailment" → "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" → "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"
"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"

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 [54].

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-params that are determined according to practical consideration. The second type is to construct a small dev set (e.g., a 32-sample-dev set). It then performs grid search and uses the small dev set for hyper-params & model selection. Our work uses 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 important research direction for future work, too.

Connection to Contrastive Learning Few-shot learning systems are prone to overfitting for its lack of samples. The core idea of FlipDA is to provide intermediate supervision about what causes a label difference to improve generalization. In this sense, there is a connection between FlipDA and contrastive learning [19, 4], which uses data augmentation to generate positive instances and uses

samples existing in the dataset as negative samples. FlipDA shows that augmented samples can also play the role of negative samples. While previous work on contrastive learning focuses on training with large data, our experiments focus on showing that augmenting a small-sized dataset is able to improve few-shot generalization. It might be intriguing to see whether such a connection might lead to advances in both the fields of contrastive pretraining and few-shot learning; e.g., generating negative samples for large-scale contrastive pretraining.

6 Conclusions

We propose to study few-shot natural language understanding based on large-scale pretrained models. We identify two key desiderata—effectiveness and robustness. Based on the empirical insight that label flipping improves few-shot generalization, we propose FlipDA that utilizes a classifier for 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 understand why and how generating label-flipped data in the neighborhood of existing data points improves generalization. Moreover, increasing the diversity and quality of augmented data generation is also an important long-term goal.

Acknowledgments and Disclosure of Funding

Zhou and Li are supported in part by the National Natural Science Foundation of China Grant 61822203, 61772297, 61632016 and the Zhongguancun Haihua Institute for Frontier Information Technology, Turing AI Institute of Nanjing and Xi'an Institute for Interdisciplinary Information Core Technology.

References

[1] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling. Do not have enough data? deep learning to the rescue! In AAAI, pages 7383–7390. AAAI Press, 2020.

[2] J. Chen, Z. Wang, R. Tian, Z. Yang, and D. Yang. Local additivity based data augmentation for semi-supervised NER. In EMNLP (1), pages 1241–1251. Association for Computational Linguistics, 2020.

[3] J. Chen, Z. Yang, and D. Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, 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, 2020.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In ICML, volume 119 of Proceedings of Machine Learning Research, pages 1597–1607. PMLR, 2020.

[5] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. ArXiv, abs/1905.10044, 2019.

[6] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, 2019.

[7] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In MLCW, 2005.

[8] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In Machine Learning Challenges Workshop, pages 177–190. Springer, 2005.

[9] M.-C. de Marneffe, M. Simons, and J. Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. 2019.

[10] M.-C. De Marneffe, M. Simons, and J. Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In proceedings of Sinn und Bedeutung, volume 23, pages 107–124, 2019.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT, 2019.

[12] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. A. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. CoRR, abs/2002.06305, 2020.

[13] M. Fadaee, A. Bisazza, and C. Monz. Data augmentation for low-resource neural machine translation. In ACL (2), pages 567–573. Association for Computational Linguistics, 2017.

[14] M. Gardner, Y. Artzi, V. Basmova, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala, N. Gupta, H. Hajishirzi, G. Ilharco, D. Khashabi, K. Lin, J. Liu, N. F. Liu, P. Mulcaire, Q. Ning, S. Singh, N. A. Smith, S. Subramanian, R. Tsarfaty, E. Wallace, A. Zhang, and B. Zhou. 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, 2020.

[15] S. Garg and G. Ramakrishnan. BAE: bert-based adversarial examples for text classification. In EMNLP (1), pages 6174–6181. Association for Computational Linguistics, 2020.

[16] A. Gordon, Z. Kozareva, and M. Roemmele. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In SemEval@NAACL-HLT, 2012.

[17] D. Guo, Y. Kim, and A. M. Rush. Sequence-level mixed sample data augmentation. In EMNLP (1), pages 5547–5552. Association for Computational Linguistics, 2020.

[18] J. He, J. Gu, J. Shen, and M. Ranzato. Revisiting self-training for neural sequence generation. In ICLR. OpenReview.net, 2020.

[19] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick. Momentum contrast for unsupervised visual representation learning. In CVPR, pages 9726–9735. IEEE, 2020.

[20] P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention. ArXiv, abs/2006.03654, 2020.

[21] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. ArXiv, abs/1503.02531, 2015.

[22] Y. Hou, Y. Liu, W. Che, and T. Liu. Sequence-to-sequence data augmentation for dialogue language understanding. In COLING, pages 1234–1245. Association for Computational Linguistics, 2018.

[23] Z. Huang and M. P. Harper. Self-training PCFG grammars with latent annotations across languages. In EMNLP, pages 832–841. ACL, 2009.

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

[25] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling BERT for natural language understanding. CoRR, abs/1909.10351, 2019.

[26] J. Kahn, A. Lee, and A. Hannun. Self-training for end-to-end speech recognition. In ICASSP, pages 7084–7088. IEEE, 2020.

[27] D. Kaushik, E. H. Hovy, and Z. C. Lipton. Learning the difference that makes A difference with counterfactually-augmented data. In ICLR. OpenReview.net, 2020.

[28] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In NAACL-HLT, 2018.

[29] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262, 2018.

[30] S. Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In M. A. Walker, H. Ji, and A. Stent, editors, 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, 2018.

[31] V. Kumar, A. Choudhary, and E. Cho. Data augmentation using pre-trained transformer models. CoRR, abs/2003.02245, 2020.

[32] V. Kumar, H. Glaude, C. de Lichy, and W. Campbell. A closer look at feature space data augmentation for few-shot intent classification. In DeepLo@EMNLP-IJCNLP, pages 1–10. Association for Computational Linguistics, 2019.

[33] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. ArXiv, abs/1909.11942, 2020.

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

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

[36] H. Levesque, E. Davis, and L. Morgenstern. The winograd schema challenge. In Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning. Citeseer, 2012.

[37] D. Li, Y. Zhang, H. Peng, L. Chen, C. Brockett, M. Sun, and B. Dolan. Contextualized perturbation for textual adversarial attack. In NAACL-HLT, pages 5053–5069. Association for Computational Linguistics, 2021.

[38] J. Li, L. Qiu, B. Tang, D. Chen, D. Zhao, and R. Yan. Insufficient data can also rock! learning to converse using smaller data with augmentation. In AAAI, pages 6698–6705. AAAI Press, 2019.

[39] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu. BERT-ATTACK: adversarial attack against BERT using BERT. In EMNLP (1), pages 6193–6202. Association for Computational Linguistics, 2020.

[40] D. Liu, Y. Gong, J. Fu, Y. Yan, J. Chen, J. Lv, N. Duan, and M. Zhou. Tell me how to ask again: Question data augmentation with controllable rewriting in continuous space. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, 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, 2020.

[41] S. Longpre, Y. Wang, and C. DuBois. How effective is task-agnostic data augmentation for pretrained transformers? In T. Cohn, Y. He, and Y. Liu, editors, 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, 2020.

[42] D. McClosky, E. Charniak, and M. Johnson. Effective self-training for parsing. In HLT-NAACL. The Association for Computational Linguistics, 2006.

[43] Z. Miao, Y. Li, X. Wang, and W. Tan. Snippext: Semi-supervised opinion mining with augmented data. In Y. Huang, I. King, T. Liu, and M. van Steen, editors, WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pages 617–628. ACM / IW3C2, 2020.

[44] N. Ng, K. Cho, and M. Ghassemi. SSMBA: self-supervised manifold based data augmentation for improving out-of-domain robustness. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, 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, 2020.

[45] H. Pham, Q. Xie, Z. Dai, and Q. V. Le. Meta pseudo labels. CoRR, abs/2003.10580, 2020.

[46] M. T. Pilehvar and J. Camacho-Collados. Wic: 10, 000 example pairs for evaluating context-sensitive representations. CoRR, abs/1808.09121, 2018.

[47] M. T. Pilehvar and J. Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In NAACL-HLT, 2019.

[48] C. Raffel, N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. ArXiv, abs/1910.10683, 2020.

[49] J. Raiman and J. Miller. Globally normalized reader. In EMNLP, pages 1059–1069. Association for Computational Linguistics, 2017.

[50] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In ICLR (Workshop), 2015.

[51] R. Reichart and A. Rappoport. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In ACL. The Association for Computational Linguistics, 2007.

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

[53] M. Roemmele, C. A. Bejan, and A. S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning, pages 90–95, 2011.

[54] C. Rosset, C. Xiong, M. Phan, X. Song, P. N. Bennett, and S. Tiwary. Knowledge-aware language model pretraining. ArXiv, abs/2007.00655, 2020.

[55] G. Russo, N. Hollenstein, C. C. Musat, and C. Zhang. Control, generate, augment: A scalable framework for multi-attribute text generation. In EMNLP (Findings), volume EMNLP 2020 of Findings of ACL, pages 351–366. Association for Computational Linguistics, 2020.

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

[57] R. Sennrich, B. Haddow, and A. Birch. 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, 2016.

[58] P. Vijayaraghavan, I. Sysoev, S. Vosoughi, and D. Roy. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns. In S. Bethard, D. M. Cer, M. Carpuat, D. Jurgens, P. Nakov, and T. Zesch, editors, 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, 2016.

[59] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In NeurIPS, 2019.

[60] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. arXiv preprint arXiv:1905.00537, 2019.

[61] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In ICLR (Poster). OpenReview.net, 2019.

[62] W. Y. Wang and D. Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, editors, 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, 2015.

[63] J. W. Wei and K. Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, 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, 2019.

[64] C. Xia, C. Zhang, H. Nguyen, J. Zhang, and P. S. Yu. CG-BERT: conditional text generation with BERT for generalized few-shot intent detection. CoRR, abs/2004.01881, 2020.

[65] Q. Xie, Z. Dai, E. H. Hovy, T. Luong, and Q. Le. Unsupervised data augmentation for consistency training. In NeurIPS, 2020.

[66] Q. Xie, M. Luong, E. H. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In CVPR, pages 10684–10695. IEEE, 2020.

[67] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan. Billion-scale semi-supervised learning for image classification. CoRR, abs/1905.00546, 2019.

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

[69] K. M. Yoo, D. Park, J. Kang, S. Lee, and W. Park. Gpt3mix: Leveraging large-scale language models for text augmentation. CoRR, abs/2104.08826, 2021.

[70] K. M. Yoo, Y. Shin, and S. Lee. Data augmentation for spoken language understanding via joint variational generation. In AAAI, pages 7402–7409. AAAI Press, 2019.

[71] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. ArXiv, abs/1804.09541, 2018.

[72] X. Yue and S. Zhou. PHICON: improving generalization of clinical text de-identification models via data augmentation. In A. Rumshisky, K. Roberts, S. Bethard, and T. Naumann, editors, Proceedings of the 3rd Clinical Natural Language Processing Workshop, ClinicalNLP@EMNLP 2020, Online, November 19, 2020, pages 209–214. Association for Computational Linguistics, 2020.

[73] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In ICLR (Poster). OpenReview.net, 2018.

[74] S. Zhang, X. Liu, J. Liu, J. Gao, K. Duh, and B. V. Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. ArXiv, abs/1810.12885, 2018.

[75] X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, 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, 2015.

[76] B. Zoph, G. Ghiasi, T. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le. Rethinking pre-training and self-training. In NeurIPS, 2020.

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 to save Video Memory.

On ALBERT, all the parameters and patterns are kept the same as PET/iPET[56]. 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"). We also reduce the learning rate from 1e-5 to 5e-6 on RTE and WiC, which will affect the results 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 Implementation Details of Baseline Augmentation Methods

Synonym Replacement. Our implementation is based on parts of the code of EDA [10]. 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. Our implementation is based on parts of the code of TinyBert [11]. 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. Our implementation is based on the code of EDA [10], 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 [63].

Back Translation. We implement it with the help of Google Translator.

TinyBERT. Our implementation is based on the code of TinyBert [11]. 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. This is the same as FlipDA 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. For each batch, we first sample $\lambda = Beta(0.5, 0.5)$, just as the author [73] 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 Fewvary a lot, it is necessary for us to adjust the details for data augmentation fo r 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 [48] 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 in 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 in 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,

[10] http://github.com/jasonwei20/eda_nlp

[11] https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/TinyBERT

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.

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 in 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.

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 fill in k blanks at a time iteratively. From our experiments, 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 some 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.) For BoolQ and MultiRC, we choose the mask ratio of 0.3 or 0.5. For COPA and WiC, we choose the mask ratio of 0.8. We also find that if the sentence length is too large, such as MultiRC, it is impossible to fill in all the blanks at a time. (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, 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. Above all, for most of the datasets, we

only search 6 hyper-parameter combination, we think this will not lead to severe overfitting, and our algorithm is stable.

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

A.6 More Results on ALBERT

In the body part, we only report the ablation results on DeBERTa because the model is larger, and it seems more stable during our experiments. Here we report ablation results on ALBERT. Most of the conclusions are the same, but some details are different. 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.

A.6.1 Effectiveness of Pattern-based Data Cloze and FlipDA Classifier

From Table 12 we can see that our method 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 12, we can see that the classification is much more useful for DeBERTa than ALBERT. In DeBERTa, almost all augmentation methods will improve their performance with the classifier. In ALBERT, only some augmentation methods will improve its performance on some tasks. We think it is normal because a better classifier will lead to better classification results, i.e., better-selected augmentation data. In RTE, all methods will contribute to better results with the classifier. We think this may be because the label is prone to be flipped in this task.

Table 12: 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+cls	74.32	84.52/79.32	82.17	63.93	49.56	34.53/74.52	67.74	6.16
KNN+cls	71.88	84.52/76.83	83.17	67.39	53.10	31.62/73.92	68.16	5.16
EDA+cls	74.16	84.52/78.92	83.00	60.41	50.49	34.22/75.52	67.44	5.33
BT-10+cls	73.37	83.04/74.19	85.00	63.12	-	34.60/74.69	70.95	3.33
BT-6+cls	73.26	80.06/68.59	86.83	61.46	-	34.49/76.05	70.23	4.46
T-BERT+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**

A.6.2 Analysis of Label-Flipping v.s. Label-Preservation

From Table 13, we can see that FlipDA is still the best, i.e., augmentation with both directions is better than with only one direction. And augmentation only with label-flipped data is better than with only label-preserved data in most tasks. This phenomenon is more obvious in DeBERTa than ALBERT, which may because the classifier quality of DeBERTa is better than ALBERT. DeBERTa has learned better representations of similar phrases, so the label-kept examples will contribute less.

Table 13: 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

A.6.3 Analysis of Label Transformation

We took a closer at the effect of label transformation direction in Table 14. In BoolQ and RTE, the two flipped directions are better than the kept directions. In all datasets, adding data with more directions is better than with only one direction, even some direction seems extremely bad. This is because only adding data with one direction will largely change the data distribution of the new dataset.

Table 14: 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

A.6.4 Analysis of Strategies for Augmented Data Selection

From Table 15, we can see that Noisy Student performs well with the ALBERT model. It achieves good results in almost all the datasets except COPA. While in DeBERTa (see Table 9), the Noisy Student is somewhat weaker. This may be because the DeBERTa model fixes the bottom 1/3 layers' parameters, and thus not suitable for the perturbation on the embedding space, such as dropout. We think a better self-training policy can further improve the performance of data augmentation.

From Table 15, the observation of the effectiveness of different strategies is somewhat similar to that on DeBERTa. Diverse TopK is better than Global TopK, which demonstrates the diversity is important. The best result can appear in Default Strategy, Global TopP, or Diverse TopK, and the gap between these choice strategies is smaller.

A.7 Case Study

We have provided some augmented examples on RTE in Table 10. Here we provide more augmented examples on other tasks. To be specific, BoolQ, WiC, and COPA. The four datasets cover tasks with different targets and sentence lengths.

WiC is a task to tell whether the word w in the two sentences has the same meaning. From Table 16, 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 can still provide good information for this task.

Table 15: 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 16: 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"	**"sentence1"**: "We vaccinate against scarlet fever." **"sentence2"**: "The nurse vaccinated the children in the school." **"sentence1"**: "We vaccinate the children against fever and malaria" **"sentence2"**: "The nurse vaccinated the children against fever and malaria"
"True" → "False"	**"sentence1"**: "You make me feel naked." **"sentence2"**: "She felt small and insignificant." **"sentence1"**: "You can feel the gravity" **"sentence2"**: "She felt so insignificant and useless"
"False" → "True"	**"sentence1"**: "Can you back up your claims?" **"sentence2"**: "I can't back this plan." **"sentence1"**: "Can you please back to your home" **"sentence2"**: "I can't back from your house"
"False" → "False"	**"sentence1"**: "Turn and face your partner now." **"sentence2"**: "The bunkers faced north and east, toward Germany." **"sentence1"**: "Get up and face it now" **"sentence2"**: "The ship faced north and south from the coast"

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 17. 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 is against common sense. The third case copies some parts of the passage into the question, and then the label flips. The last case changed 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 18. 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 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. From the examples, we can see that the T5-model has human knowledge and it is a good augmenter.

Table 17: 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 18: 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	**"choice1"**: "She excavated ancient artifacts." **"choice2"**: "She read about the site's history." **"premise"**: "The archeologist dug up the site." **"question"**: "effect" **"label"**: 0 **"premise"**: "She dug up the earth." **"question"**: "effect" **"label"**: 0
Flip-label	Keep-question	**"choice1"**: "She began going to church." **"choice2"**: "She began travelling abroad." **"premise"**: "The woman had a religious awakening." **"question"**: "effect" **"label"**: 0 **"premise"**: "She had a lot of money." **"question"**: "effect" **"label"**: 1
	Flip-question (effect → cause)	**"choice1"**: "Her friend sent her a greeting card." **"choice2"**: "Her friend cut off contact with her." **"premise"**: "The woman betrayed her friend." **"question"**: "effect" **"label"**: 1 **"premise"**: "A woman is happy." **"question"**: "cause" **"label"**: 0
	Flip-question (cause → effect)	**"choice1"**: "The cafe reopened in a new location." **"choice2"**: "They wanted to catch up with each other." **"premise"**: "The women met for coffee." **"question"**: "cause" **"label"**: 1 **"premise"**: "The cafe closed." **"question"**: "effect" **"label"**: 0