

How Effective is Task-Agnostic Data Augmentation for Pretrained Transformers?

Shayne Longpre*

Apple Inc.

slongpre@apple.com

Yu Wang*

Apple Inc.

w.y@apple.com

Christopher DuBois

Apple Inc.

cdubois@apple.com

Abstract

Task-agnostic forms of data augmentation have proven widely effective in computer vision, even on pretrained models. In NLP similar results are reported most commonly for low data regimes, non-pretrained models, or situationally for pretrained models. In this paper we ask how effective these techniques really are when applied to pretrained transformers. Using two popular varieties of task-agnostic data augmentation (not tailored to any particular task), Easy Data Augmentation (Wei and Zou, 2019) and Back-Translation (Sennrich et al., 2015), we conduct a systematic examination of their effects across 5 classification tasks, 6 datasets, and 3 variants of modern pretrained transformers, including BERT, XLNET, and ROBERTA. We observe a negative result, finding that techniques which previously reported strong improvements for non-pretrained models fail to consistently improve performance for pretrained transformers, even when training data is limited. We hope this empirical analysis helps inform practitioners where data augmentation techniques may confer improvements.

1 Introduction

“Task-agnostic” data augmentations — those which are not tailored to a task, but are broadly applicable across the visual or textual domain — have long been a staple of machine learning. Task-agnostic data augmentation techniques for computer vision, such as image translation, rotation, shearing, and contrast jittering, have achieved considerable success, given their ease of use, and wide-spread applicability (Cubuk et al., 2018; Perez and Wang, 2017). In natural language processing, benefits of data augmentation have usually been observed where the augmentations are suited to the task: as with back-translation for machine translation (Edunov et al.,

2018; Xia et al., 2019), or negative sampling for question answering and document retrieval (Zhang et al., 2017; Yang et al., 2019a; Xiong et al., 2020). Outside of application-tailored augmentations, improvements are primarily reported on autoregressive models without unsupervised pretraining or contextual embeddings, such as LSTMs and CNNs, and even then in low data regimes (Zhang et al., 2015; Coulombe, 2018; Wei and Zou, 2019; Yu et al., 2018). Additionally, in computer vision task-agnostic augmentations continue to report benefits when applied to pretrained representations (Gu et al., 2019). However, in NLP it is less clear whether these general augmentations benefit modern Transformer architectures with unsupervised pretraining at scale.

We pose the question: to what extent do modern NLP models benefit from task-agnostic data augmentations? In this paper, we provide empirical results across a variety of tasks, datasets, architectures, and popular augmentation strategies. Among data augmentation techniques, we select Easy Data Augmentation (Wei and Zou, 2019) and Back-Translation (Sennrich et al., 2015); EDA and BT respectively. Both are popular task-agnostic options, and report significant gains for LSTMs on a wide variety of datasets. We apply these techniques to 6 classification-oriented datasets, spanning 5 tasks with varying linguistic objectives and complexity. For fair comparison, we tune each of BERT, XLNET, and ROBERTA extensively, allocating an equal budget of trial runs to models trained with and without augmentations. As a separate dimension, we also vary the availability of training data to understand under what specific conditions data augmentation is beneficial.

Our findings demonstrate that these popular task-agnostic data augmentations provide only sparse and inconsistent improvements for modern pretrained transformers on many simple classification

* equal contribution

Dataset	c	l	$ D_{train} $
SST-2 (Socher et al., 2013)	2	19	7.6k
SUBJ (Pang and Lee, 2004)	2	23	8k
RT (Pang and Lee, 2005)	2	21	8.7k
MNLI (Williams et al., 2017)	3	2x17	8k
STS-B (Baudiš et al., 2016)	5	2x12	6.6k
TREC (Li and Roth, 2002)	6	10	3.9k

Table 1: Summary statistics for each dataset. c : The number of classes. l : The average sequence length in word tokens. D_{train} : The training set size after random sampling up to 10k unique examples (if available), and subtracting 2k for dev and test sets.

tasks. They further lend empirical evidence to the hypothesis that task-agnostic data augmentations may be significantly less effective on pretrained transformers for other classification and NLP tasks. Observed patterns suggest that the scale of pretraining may be the critical factor replacing the need for linguistic variety that augmentations confer. We hope our work provides guidance to ML practitioners in deciding when to use data augmentation and encourages further examination of its relationship to unsupervised pretraining.

2 Experimental Methodology

2.1 Datasets

Following Wei and Zou (2019) and Wu et al. (2019) we adopt 4 classification datasets on which general data augmentation techniques demonstrated strong performance gains, and include 2 more from the GLUE benchmark (Wang et al., 2018). As shown in Table 1, a variety of classification sizes, sequence lengths, and vocabulary sizes are represented. Included tasks are sentiment analysis (SST-2, RT), subjectivity detection (SUBJ), question type classification (TREC), semantic similarity (STS-B) and natural language inference (MNLI).

2.2 Augmentation Techniques

Among the many variations of data augmentation two families are widely used in NLP: back translation and text editing.

Back Translation (BT): We use an English to German machine translation model (Ott et al., 2018) and a German to English model (Ng et al., 2019).¹ We selected German due to its strong results as a pairing with English for back translation, as reported in Yu et al. (2018); Sennrich et al.

¹Adapted from <https://ai.facebook.com/tools/fairseq/>.

(2015). We translate each English sentence to one German sentence and back to six candidate English sentences. From these sentence candidates we obtain the best results sampling the most distant sentence from the original English sentence, measured by word edit distance. From manual inspection this approach produced the most diverse paraphrases, though this strategy needs to be tailored to the machine translation systems employed. The overall aim of this strategy is to maximize linguistic variety while retaining sentence coherency.

Easy Data Augmentation (EDA): Following Wei and Zou (2019) we employ a combination of popular text editing techniques that have shown strong performance on LSTMs.² Text edits include synonym replacement, random swap, random insertion, and random deletion. To improve upon EDA further, we enforce part-of-speech consistency for synonym selection. As an example, the verb “back” in the phrase “to back the government” will not be replaced by “rear”, which is a synonym of the noun “back”.

2.3 Experimental Setup

To conduct a fair assessment of each data augmentation technique, we ensure three properties of our experimental setup: (I) our tuning procedure mimics that of a machine learning practitioner; (II) the selected hyperparameters cannot be significantly improved as to change our conclusions; and (III) each strategy is evaluated with an equal number of trial runs.³

We experiment with 3 types of Transformers (Vaswani et al., 2017): BERT-BASE (Devlin et al., 2019), XLNET-BASE (Yang et al., 2019b), and ROBERTA-BASE (Liu et al., 2019). These models each use slightly different pretraining strategies. BERT and ROBERTA are both pretrained with Masked Language Modeling, but with different auxiliary objectives and number of training steps. XLNET is pretrained with its own “Permutation Language Modeling”. For each model and dataset 1k examples are randomly selected for each of the validation and test sets. Separately from these fixed sets, we iterate over five training data sizes $N \in \{500, 1000, 2000, 3000, \text{Full}\}$ to simulate data scarcity.

²We use the implementation at https://github.com/jasonwei20/eda_nlp.

³We verify property (II), that the tuning budget is sufficient, by doubling the allocated trials and observing the magnitude of changes (see Appendix B).

Algorithm 1 EXPERIMENTAL DESIGN

Input: Model M , Dataset D , Train size N **Output:** Mean and standard deviation for test accuracies

```
( $\mu_{NoDA}, \sigma_{NoDA}$ ), ( $\mu_{BT}, \sigma_{BT}$ ), ( $\mu_{EDA}, \sigma_{EDA}$ )
1:  $T_1, T_2, K \leftarrow 3, 20, 30$ 
2:  $D_{train} \leftarrow \text{sample}(\text{shuffle}(D), N)$ 
3: for each Augmentation  $\alpha$  in  $[NoDA, BT, EDA]$  do
4:   // Find best hyperparameters  $H_\alpha$  for augmentation  $\alpha$ 
5:    $H_\alpha \leftarrow \text{RANDOMSEARCH}(M, D_{train}, K, T_1)$ 
6:    $M_\alpha \leftarrow M.\text{USE}(H_\alpha)$ 
7:   // Compute validation scores for augmentation  $\alpha$ 
8:   for  $s = 1$  to  $T_2$  do
9:      $\text{SCORES} \leftarrow \text{TRAIN}(M_\alpha, D_{train}, \text{seed}=s)$ 
10:  end for
11:  // Select test scores using best validation scores
12:   $\mu_\alpha, \sigma_\alpha \leftarrow \text{SELECTBEST}(\text{Scores}, 10)$ 
13: end for
14: return ( $\mu_{NoDA}, \sigma_{NoDA}$ ), ( $\mu_{BT}, \sigma_{BT}$ ), ( $\mu_{EDA}, \sigma_{EDA}$ )
```

Given a model M , dataset D , and training set size N , we allocate an equal number of training runs to No Augmentation (NO DA), EDA, and BT. For each setting, we define continuous ranges for the learning rate, dropout, and number of epochs. EDA and BT settings also tune a “dosage” parameter governing augmentation $\tau \in \{0.5, 1, 1.5, 2\}$. $N \times \tau$ is the quantity of augmented examples added to the original training set.

First, we conduct a RANDOMSEARCH for $K = 30$ parameter choices, each repeated for $T_1 = 3$ trials with differing random training seed. As shown in Algorithm 1 this stage returns the optimal hyperparameter choices H_α for each augmentation type $\alpha \in \{NoDA, BT, EDA\}$. The best hyperparameters are selected by mean validation accuracy over random seed trials. In the second stage, a model with these best hyperparameters (Algorithm 1 line 6) is trained over $T_2 = 20$ random seed trials.⁴ Finally, the 10 best trials by validation accuracy are selected for each per setting (line 12). We report the mean and 95% confidence intervals of their test results. The bottom 10 trials are discarded to account for the high accuracy variance of pretrained language models with respect to weight initialization, and data order (Dodge et al., 2020). This procedure closely mimics that of an ML practitioner looking to select the best model.⁵

3 Empirical Results

Figure 1 shows both the baseline NO DA test accuracies as a reference point, and the mean relative

⁴Note that we cache the top performing trials from T_1 to reduce total trial runs.

⁵Further details for our model tuning procedure are available in Appendix A.

improvement from applying EDA and BT. Empirically, improvements are marginal for 5 of the 6 datasets, only exceeding 1% for BERT-B in a couple of instances where $N \leq 1000$. XLNET-B and ROBERTA-B see no discernible improvements at almost any data level and just as frequently observe regressions in mean accuracy from EDA or BT. MNLI presents a clear outlier, with augmentations yielding relative improvements in excess of 2%, but only for BERT. In contrast, the other pretrained transformers experience unpredictable, and mostly negative results.

In terms of augmentation preferences for BERT, BT confers superior results to EDA in 60% of cases, averaging 0.18% absolute difference. This advantage is muted for both XLNET and ROBERTA, with only 53% of cases preferring BT, and at smaller margins.

Table 2 shows the improvement of **either** EDA or BT over NO DA, averaged across all 6 datasets. We compare against Wei and Zou (2019)’s experiments, measuring the impact of EDA on LSTM and CNN models over 5 classification datasets.⁶ They observe consistent improvements for non-pretrained models. LSTMs and CNNs improve 3.8% and 2.1% on average at $N = 500$ training points, and 0.9% and 0.5% on average with full data (approximately equivalent to our own FULL setting). As compared to these, BERT observes muted benefits. To exclude MNLI from this average (not present in Wei and Zou (2019)’s experiments) would reduce all of BERT’s improvements well below 1%. ROBERTA and XLNET again show no signs of improvement, frequently yielding worse results than the baseline, even with the best data augmentation.

Finally, we examine the claim that data augmentation confers an advantage with any statistical significance. We use a one-sided t-test with null hypothesis that data augmentation confers a greater mean performance than without, using p-value of .05. Examining BT and EDA vs NO DA over all dataset and data sizes we reject the null hypothesis in 43%, 85%, and 87% of cases for BERT, XLNET, and ROBERTA respectively. Moreover, for ROBERTA, the inverse hypothesis (that NO DA is statistically better than DA) is true in 28% of cases.

We believe these results are surprising due to two advantages given to data augmentation in this ex-

⁶Their experimental setup is directly comparable to our own, comprising similar training sizes, datasets, and tuning procedures.

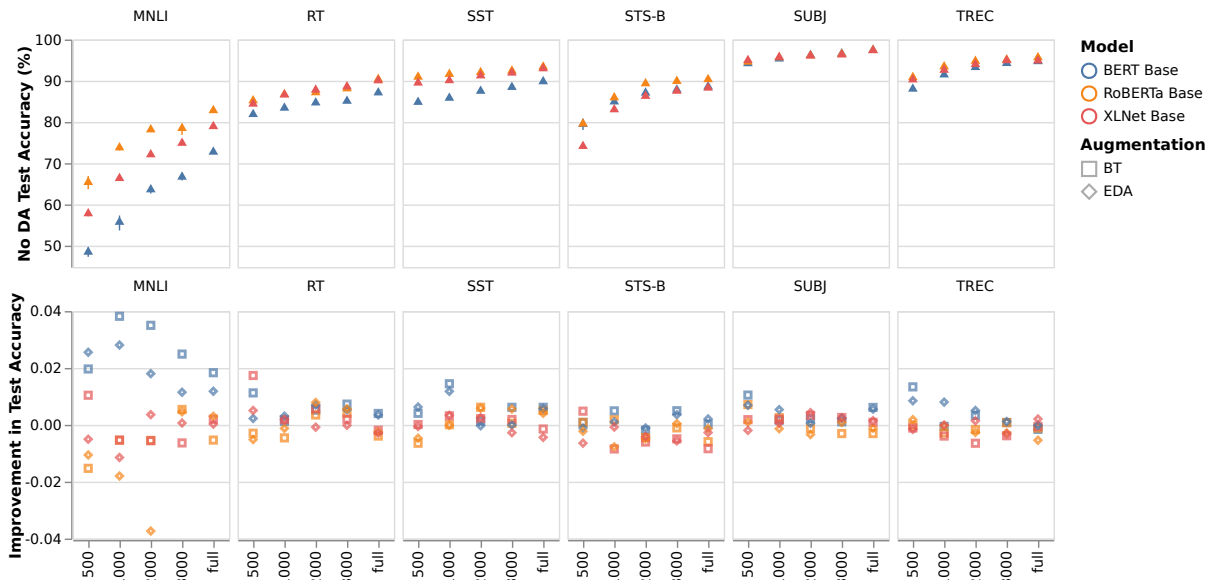


Figure 1: The upper row plots the mean test accuracies, with 95% confidence intervals, across model types, datasets, and training sizes for the NO DA baseline setting (displayed with triangle points). The lower row plots the mean relative improvement in test accuracy over the No DA setting, for each augmentation type.

Model	Train Size (N)				
	500	1000	2000	3000	Full
LSTM [†]	+3.8	-	+0.7	-	+0.9
CNN [†]	+2.1	-	+0.8	-	+0.5
BERT-B	+1.13	+1.23	+0.82	+0.78	+0.61
XLNET-B	+0.56	+0.02	+0.22	-0.02	-0.01
ROBERTA-B	-0.14	-0.04	+0.02	+0.27	+0.03

Table 2: The absolute improvement in test accuracy (%) by **either** data augmentation technique over NO DA. Results are averaged over all 6 datasets.

[†] We include Wei and Zou (2019)’s results for comparison, though their setup differs slightly: they show the improvement only for EDA (not the best of EDA and BT), and they average over 5 classification datasets, of which we have SST-2, SUBJ, and TREC in common.

perimental setup: (A) these are relatively low data regimes compared to what is available for most tasks, and (B) in total, the data augmentation techniques receive twice the number of tuning trials as the NO DA baseline. Even if EDA and BT confer no advantage over NO DA, we would expect to see a minor positive increase from tuning over twice as many trials.⁷

4 Discussion & Limitations

Our empirical results verify that popular data augmentation techniques fail to consistently improve performance for modern pretrained transformers — even when training data is limited. A single excep-

⁷Per dataset metrics, with confidence intervals, are available in Appendix C.

tion (BERT-B on MNLI) sees significant benefits from data augmentation. We speculate the outlier results could pertain to the inherent difficulty of natural language inference in low data regimes. Alternatively, Gururangan et al. (2018) discuss “annotation artifacts” in MNLI that lead models to rely on simple heuristics, such as the presence of the word “not” in order to make classifications. EDA or BT could mitigate these spurious cues by distributing artifacts more evenly across labels.

4.1 Why can Data Augmentation be ineffective?

Our results consistently demonstrate that augmentation provides more benefits to BERT than to ROBERTA and XLNET. The key distinguishing factor between these models is the scale of unsupervised pretraining; therefore, we hypothesize that pretraining provides the same benefits targeted by common augmentation techniques. Arora et al. (2020) characterize the benefits of contextual embeddings, showing a boost on tasks containing complex linguistic structures, ambiguous word usage, and unseen words. Text editing and translation techniques vary linguistic structure and word usage to address these same issues. Under this hypothesis, we would expect new data augmentation techniques to help only when they provide linguistic patterns that are relevant to the task but not seen during pretraining.

Manually inspecting RT examples for which an

LSTM requires augmentation to classify correctly, but ROBERTA does not, we observe rare word choice, atypical sentence structure and generally off-beat reviews. This set contains reviews such as “suffers from over-familiarity since hit-hungry british filmmakers have strip-mined the monty formula mercilessly since 1997”, “wishy-washy”, or “wanker goths are on the loose! run for your lives!”, as compared to “exceptionally well acted by diane lane and richard gere”, more representative of examples outside this set. We verify this quantitatively: for 100 examples in this set there are 206 (rare) words which **only** appear in this set, whereas for 100 random samples we see an average of 116 rare words. Interestingly, we also notice label skew in this set (34% of examples are positive instead of the overall mean of 50%). While we leave deeper analysis to future work, we believe these results suggest data augmentation and pretraining both improve a model’s ability to handle complex linguistic structure, ambiguous word usage, and unseen words within a category of label.

4.2 When can Data Augmentation be useful?

Given these observations, where might task-agnostic data augmentation be useful (with pretrained models)? One candidate application is out-of-domain generalization. However, we believe the target domain must **not** be well represented in the pretraining corpus. For instance, Longpre et al. (2019) did not find standard BT useful for improving generalization of question answering models. While their training domains are diverse, they are mostly based in Wikipedia and other common sources well represented in the BERT pretraining corpus. Additionally, we suspect it is not enough to vary/modify examples in ways already seen in pretraining. Our results motivate more sophisticated (read: targeted) augmentation techniques rather than generic, task (and domain)-agnostic strategies which unsupervised pretraining may capture more effectively.

Another candidate application of task-agnostic data augmentation is semi-supervised learning. Xie et al. (2019) illustrate a use for generic data augmentations as a noising agent for their consistency training method, assuming large quantities of unlabeled, in-domain data are available. While task-agnostic data augmentations are effective in this particular task setup, they are not the critical factor in the success of the method, nor is it clear

that more tailored or alternative noising techniques might not achieve even greater success.

To our knowledge, our experiments provide the most extensive examination of task-agnostic data augmentation for pretrained transformers. Nonetheless, our scope has been limited to classification tasks, and to the more common models and augmentations techniques.

5 Conclusion

We examine the effect of task-agnostic data augmentation in modern pretrained transformers. Isolating low data regimes ($< 10k$ training data points) across a range of factors, we observe a negative result: popular augmentation techniques fail to consistently improve performance for modern pretrained transformers. Further, we provide empirical evidence that suggests the scale of pretraining may be the primary factor in the diminished efficacy of textual augmentations. We hope our work provides guidance to ML practitioners in deciding when to use data augmentation and encourages further examination of its relationship to unsupervised pretraining.

Acknowledgments

We would like to acknowledge Stephen Pulman, Andrew Fandrianto, Drew Frank, Leon Gatys, Silvana Ciurea-Ilcus, Hang Zhao, and Kanit Wongsuphasawat for their guiding insights and helpful discussion.

References

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. [TensorFlow: Large-scale machine learning on heterogeneous systems](#). Software available from tensorflow.org.
- Simran Arora, Avner May, Jian Zhang, and Christopher Ré. 2020. Contextual embeddings: When are they worth it? *arXiv preprint arXiv:2005.09117*.
- Petr Baudiš, Jan Pichl, Tomáš Vyskočil, and Jan Šedivý. 2016. Sentence pair scoring: Towards

- unified framework for text comprehension. *arXiv preprint arXiv:1603.06127*.
- Claude Coulombe. 2018. [Text data augmentation made simple by leveraging nlp cloud apis](#).
- Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. 2018. [Autoaugment: Learning augmentation policies from data](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Shanqing Gu, Manisha Pednekar, and Robert Slater. 2019. Improve image classification using data augmentation and neural networks. *SMU Data Science Review*, 2(2):1.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Xin Li and Dan Roth. 2002. [Learning question classifiers](#). In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, page 1–7, USA. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. 2019. [An exploration of data augmentation and sampling techniques for domain-agnostic question answering](#).
- Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair’s wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9.
- Bo Pang and Lillian Lee. 2004. [A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts](#). In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 271–es, USA. Association for Computational Linguistics.
- Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Luis Perez and Jason Wang. 2017. [The effectiveness of data augmentation in image classification using deep learning](#).
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. [Improving neural machine translation models with monolingual data](#).
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform

- for natural language understanding. *EMNLP 2018*, page 353.
- Jason Wei and Kai Zou. 2019. [Eda: Easy data augmentation techniques for boosting performance on text classification tasks](#).
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pages 84–95. Springer.
- Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, volume 57.
- Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. [Data augmentation for bert fine-tuning in open-domain question answering](#).
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.
- Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. [Qanet: Combining local convolution with global self-attention for reading comprehension](#).
- Haotian Zhang, Jinfeng Rao, Jimmy Lin, and Mark D. Smucker. 2017. [Automatically extracting high-quality negative examples for answer selection in question answering](#). In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17*, page 797–800, New York, NY, USA. Association for Computing Machinery.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. [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*, pages 649–657. Curran Associates, Inc.

A Reproducibility

A.1 Transformer Models and Training

We share the details of our hyper-parameter selection, for easy reproducibility. For each of BERT, XLNET, and ROBERTA we use configurations mostly consistent with their original releases’ recommendations.

In all cases code is adapted with minimal changes from open source repositories. The majority of changes to each repository pertain to supporting all 6 datasets, their augmentations, as well as better metrics reporting. All models were trained on 1 NVIDIA Tesla V100 GPU.

For each model we tune over 4 hyperparameters to which the final performance was particularly sensitive. The “augmentation dose” parameter, as described in the paper, only applies to models trained with either EDA or BT. We verify in Appendix Section B that the addition of this tuning dimension did not alter our conclusions with respect to the impact of data augmentation when fully tuned. Lastly, we would note that the final model size varies slightly depending on the size of the classification head — dictated by the number of classes in the task.

A.2 Bert-Base

For BERT (Devlin et al., 2019) we use the original implementation in TensorFlow (Abadi et al., 2015).⁸ See Table 3 for details in our training setup and hyperparameter tuning ranges.

A.3 XLNet-Base

For XLNET (Yang et al., 2019b) we also use the original implementation in TensorFlow.⁹ See Table 4 for details in our training setup and hyperparameter tuning ranges.

A.4 RoBERTa-Base

For ROBERTA (Liu et al., 2019) we use a standard PyTorch (Paszke et al., 2019) implementation as provided by HuggingFace.¹⁰ See Table 5 for details in our training setup and hyperparameter tuning ranges.

⁸Code adapted from <https://github.com/google-research/bert>.

⁹Code adapted from <https://github.com/zihangdai/xlnet>.

¹⁰Code adapted from <https://github.com/huggingface/transformers>.

MODEL PARAMETERS	VALUE/RANGE
Fixed Parameters	
Batch Size	50
Optimizer	Adam
Learning Rate Schedule	Exponential Decay
Lower Case	True
Max Sequence Length	100
Tuned Parameters	
Num Epochs	[2, 100]
Dropout	[0.05, 0.15]
Learning Rate	[$1e - 5$, $5e - 5$]
Augmentation Dose	[0.5, 2.0]
Extra Info	
Model Size (# params)	108.3M
Vocab Size	30,522
Avg. Runtime (Full data)	46m

Table 3: Hyperparameter selection and tuning ranges for BERT-BASE.

MODEL PARAMETERS	VALUE/RANGE
Fixed Parameters	
Batch Size	12
Optimizer	Adam
Learning Rate Schedule	Exponential Decay
Lower Case	True
Max Sequence Length	100
Tuned Parameters	
Num Epochs	[2, 20]
Dropout	[0.05, 0.15]
Learning Rate	[$1e - 5$, $5e - 5$]
Augmentation Dose	[0.5, 2.0]
Extra Info	
Model Size (# params)	117.3M
Vocab Size	32,000
Avg. Runtime (Full data)	37m

Table 4: Hyperparameter selection and tuning ranges for XLNET-BASE.

A.5 Datasets

We experimnt with 6 classification datasets. These are SST-2 (Socher et al., 2013)¹¹ and RT (Pang and Lee, 2005)¹² for sentiment analysis, SUBJ (Pang and Lee, 2004)¹³ for subjectivity detection, TREC (Li and Roth, 2002)¹⁴ for question type

¹¹Available at <https://nlp.stanford.edu/sentiment/>

¹²Available at <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

¹³Available at <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

¹⁴Available at <https://cogcomp.seas.upenn.edu/Data/QA/QC/>

MODEL PARAMETERS	VALUE/RANGE
Fixed Parameters	
Batch Size	50
Optimizer	Adam
Learning Rate Schedule	Exponential Decay
Lower Case	False
Max Sequence Length	100
Tuned Parameters	
Num Epochs	[2, 20]
Dropout	[0.05, 0.15]
Learning Rate	[$1e-5$, $5e-5$]
Augmentation Dose	[0.5, 2.0]
Extra Info	
Model Size (# params)	125.2M
Vocab Size	50,265
Avg. Runtime (Full data)	32m

Table 5: Hyperparameter selection and tuning ranges for ROBERTA-BASE.

classification, STS-B (Baudiš et al., 2016)¹⁵ for semantic similarity, and MNLI (Williams et al., 2017)¹⁶ for natural language inference. For each of these we randomly sample up to $10k$ data points (if available) from the training sets, and separate out $1k$ for each of validation and testing. Additional statistics are available in the main paper.

B Verifying Tuning Sufficiency

To ensure our conclusions are reliable we must verify that our tuning is sufficient to capture all the benefits of data augmentation. Accordingly, we double the number of hyperparameter configurations (K) and see if any of the conclusions change. As this experiment is computationally expensive, we benchmark the results only for BERT on SST-2. Full results are shown in Table 6.

We observe that on average, doubling the number of configuration trials from $K_A = 30$ to $K_B = 60$ results in minor accuracy improvements at lower training set sizes (e.g. $+0.38$ at $N = 500$), and negligible variations at higher training set sizes (e.g. -0.04 at $N = Full$). We also measure the resulting change in the difference between using and not using any data augmentation ($\Delta(DA - No DA)$). While improvements are reported in favour of data augmentation over NO DA, they are all $< 0.15\%$, indicating at $K = 30$ our conclusions are robust.

¹⁵Available at <https://gluebenchmark.com/tasks>

¹⁶Available at <https://www.nyu.edu/projects/bowman/multinli/>

C Empirical Results

Detailed results are provided for analysis. In each results table we include the mean accuracy and 95% confidence interval, for every dataset, augmentation type, and training data size. These are the outputs of the second stage of tuning “SELECTBEST” that use the best hyperparameters per setting in the first RANDOMSEARCH stage. We select only the top 10 trials of 20 (by validation accuracy) to compute these test statistics, due to the observed volatility in fine-tuning Transformers with different seeds (Dodge et al., 2020).

The full results are shown below for BERT-BASE (Table 7), for XLNET-BASE (see Table 8), and for ROBERTA-BASE (see Table 9).

DATASET	D. AUG.	N=500	N=1000	N=2000	N=3000	FULL
$K_A = 30$	No DA.	84.29 ± 0.02	85.50 ± 0.00	87.46 ± 0.01	88.40 ± 0.01	89.79 ± 0.00
	BT	85.38 ± 0.01	87.35 ± 0.02	87.37 ± 0.02	89.29 ± 0.01	90.36 ± 0.00
	EDA	85.06 ± 0.01	86.70 ± 0.02	87.83 ± 0.01	88.55 ± 0.01	90.57 ± 0.01
$K_B = 60$	No DA.	84.64 ± 0.02	85.72 ± 0.01	87.80 ± 0.01	88.66 ± 0.01	89.77 ± 0.01
	BT	85.30 ± 0.01	86.78 ± 0.01	87.69 ± 0.01	89.36 ± 0.01	90.25 ± 0.00
	EDA	85.90 ± 0.01	87.22 ± 0.00	87.48 ± 0.01	88.52 ± 0.01	90.25 ± 0.01
$K_B - K_A$	No DA.	+0.38	+0.08	+0.03	+0.11	-0.06
	BT	+0.25	-0.12	+0.20	+0.18	+0.01
	EDA	+0.51	+0.20	+0.17	+0.03	-0.06
MEAN ($K_B - K_A$)		+0.38	+0.05	+0.13	+0.11	-0.04
$\Delta(\text{DA} - \text{No DA})$		+0.13	-0.15	+0.17	+0.06	+0.07

Table 6: Here we verify that $K = 30$ hyperparameter trials is sufficient to accurately estimate the benefit of data augmentation with full tuning. For BERT-B on SST-2 we compare the results of $K_A = 30$ (used in the paper) and $K_B = 60$. MEAN $K_B - K_A$ compares the average difference in mean performances by doubling the number of trials. $\Delta(\text{DA} - \text{No DA})$ measures the difference between the accuracies of the best data augmentation technique and No DA.

DATASET	D. AUG.	N=500	N=1000	N=2000	N=3000	FULL
RT	No DA.	81.82 ± 0.02	83.23 ± 0.01	84.44 ± 0.01	85.08 ± 0.01	86.98 ± 0.00
	BT	82.84 ± 0.01	83.43 ± 0.01	85.15 ± 0.01	85.64 ± 0.01	87.62 ± 0.01
	EDA	81.26 ± 0.01	83.52 ± 0.01	85.17 ± 0.01	85.41 ± 0.01	87.57 ± 0.00
SUBJ	No DA.	94.07 ± 0.00	95.16 ± 0.01	96.34 ± 0.00	96.41 ± 0.01	97.40 ± 0.00
	BT	95.09 ± 0.01	95.27 ± 0.01	96.28 ± 0.00	96.43 ± 0.00	98.04 ± 0.00
	EDA	94.81 ± 0.01	95.68 ± 0.01	96.13 ± 0.00	96.77 ± 0.00	98.05 ± 0.00
SST-2	No DA.	84.29 ± 0.02	85.50 ± 0.00	87.46 ± 0.01	88.40 ± 0.01	89.79 ± 0.00
	BT	85.38 ± 0.01	87.35 ± 0.02	87.37 ± 0.02	89.29 ± 0.01	90.36 ± 0.00
	EDA	85.06 ± 0.01	86.70 ± 0.02	87.83 ± 0.01	88.55 ± 0.01	90.57 ± 0.01
TREC	No DA.	87.51 ± 0.01	91.25 ± 0.01	93.00 ± 0.00	94.23 ± 0.00	94.67 ± 0.00
	BT	88.95 ± 0.01	91.24 ± 0.01	93.42 ± 0.01	94.31 ± 0.00	94.59 ± 0.00
	EDA	88.75 ± 0.01	92.25 ± 0.01	93.76 ± 0.00	94.31 ± 0.00	94.51 ± 0.00
MNLI	No DA.	47.29 ± 0.04	54.18 ± 0.07	62.50 ± 0.03	65.92 ± 0.03	72.90 ± 0.01
	BT	49.23 ± 0.02	58.15 ± 0.03	66.46 ± 0.03	68.84 ± 0.03	74.25 ± 0.02
	EDA	50.03 ± 0.03	56.92 ± 0.03	64.88 ± 0.02	67.04 ± 0.03	73.85 ± 0.02
STS-B	No DA.	77.93 ± 0.04	84.61 ± 0.01	87.26 ± 0.00	87.73 ± 0.01	88.40 ± 0.01
	BT	77.94 ± 0.05	84.97 ± 0.01	86.62 ± 0.01	88.13 ± 0.01	88.35 ± 0.01
	EDA	78.27 ± 0.06	84.83 ± 0.02	86.89 ± 0.00	88.09 ± 0.00	88.56 ± 0.00

Table 7: BERT-BASE mean test accuracy and the 95% confidence interval for each task, augmentation, and data size, computed over the top 10 best trials, by validation score.

DATASET	D. AUG.	N=500	N=1000	N=2000	N=3000	FULL
RT	No DA.	83.85 ± 0.01	86.58 ± 0.01	87.69 ± 0.01	88.59 ± 0.00	89.97 ± 0.01
	BT	86.34 ± 0.01	86.77 ± 0.01	88.23 ± 0.00	88.80 ± 0.00	89.97 ± 0.01
	EDA	84.71 ± 0.01	86.68 ± 0.01	87.54 ± 0.00	88.61 ± 0.00	89.94 ± 0.00
SUBJ	No DA.	94.88 ± 0.00	95.65 ± 0.00	95.99 ± 0.00	96.29 ± 0.00	97.28 ± 0.00
	BT	95.23 ± 0.00	96.07 ± 0.00	96.52 ± 0.00	96.65 ± 0.00	97.40 ± 0.00
	EDA	94.69 ± 0.01	95.75 ± 0.01	96.41 ± 0.00	96.62 ± 0.00	97.5 ± 0.00
SST-2	No DA.	89.44 ± 0.01	90.10 ± 0.00	91.20 ± 0.01	91.87 ± 0.01	92.98 ± 0.00
	BT	89.43 ± 0.01	90.36 ± 0.01	91.39 ± 0.00	92.00 ± 0.00	92.88 ± 0.00
	EDA	89.07 ± 0.01	90.45 ± 0.00	91.59 ± 0.00	91.49 ± 0.00	92.5 ± 0.00
TREC	No DA.	90.36 ± 0.00	92.46 ± 0.01	93.94 ± 0.00	95.07 ± 0.00	94.85 ± 0.00
	BT	90.03 ± 0.01	92.16 ± 0.00	93.14 ± 0.00	94.56 ± 0.00	94.74 ± 0.00
	EDA	90.11 ± 0.01	92.51 ± 0.00	94.14 ± 0.00	94.64 ± 0.00	95.16 ± 0.00
MNLI	No DA.	57.32 ± 0.02	65.80 ± 0.01	72.07 ± 0.01	74.97 ± 0.01	78.75 ± 0.01
	BT	58.88 ± 0.02	65.49 ± 0.02	71.67 ± 0.01	74.16 ± 0.02	79.08 ± 0.01
	EDA	56.90 ± 0.02	64.65 ± 0.02	72.48 ± 0.01	74.81 ± 0.00	78.87 ± 0.01
STS-B	No DA.	73.76 ± 0.02	82.88 ± 0.00	86.29 ± 0.00	87.52 ± 0.00	88.35 ± 0.00
	BT	74.82 ± 0.01	82.17 ± 0.01	85.73 ± 0.00	86.96 ± 0.00	87.52 ± 0.00
	EDA	73.18 ± 0.01	82.87 ± 0.00	85.90 ± 0.00	87.05 ± 0.00	87.98 ± 0.00

Table 8: XLNET-BASE mean test accuracy and the 95% confidence interval for each task, augmentation, and data size, computed over the top 10 best trials, by validation score.

DATASET	D. AUG.	N=500	N=1000	N=2000	N=3000	FULL
RT	No DA.	84.84 ± 0.01	86.71 ± 0.00	87.05 ± 0.01	87.99 ± 0.01	90.10 ± 0.00
	BT	84.66 ± 0.01	86.00 ± 0.01	87.48 ± 0.01	88.44 ± 0.01	90.08 ± 0.00
	EDA	84.26 ± 0.01	86.53 ± 0.01	87.89 ± 0.01	88.40 ± 0.00	90.19 ± 0.01
SUBJ	No DA.	94.27 ± 0.00	95.50 ± 0.01	96.22 ± 0.00	96.48 ± 0.00	97.36 ± 0.00
	BT	95.14 ± 0.00	95.74 ± 0.00	96.11 ± 0.00	96.28 ± 0.00	97.05 ± 0.00
	EDA	94.55 ± 0.00	95.42 ± 0.01	95.87 ± 0.00	96.50 ± 0.00	97.31 ± 0.00
SST-2	No DA.	90.80 ± 0.00	91.51 ± 0.00	91.95 ± 0.01	92.28 ± 0.01	93.52 ± 0.00
	BT	90.13 ± 0.01	91.64 ± 0.00	92.75 ± 0.00	92.45 ± 0.00	93.96 ± 0.00
	EDA	90.16 ± 0.01	91.58 ± 0.00	92.55 ± 0.00	92.92 ± 0.01	93.85 ± 0.00
TREC	No DA.	90.77 ± 0.00	93.41 ± 0.00	94.80 ± 0.01	95.03 ± 0.01	95.66 ± 0.00
	BT	90.65 ± 0.00	93.04 ± 0.00	94.66 ± 0.00	94.99 ± 0.00	95.46 ± 0.00
	EDA	90.97 ± 0.01	93.44 ± 0.00	94.70 ± 0.00	94.80 ± 0.00	95.14 ± 0.00
MNLI	No DA.	63.3 ± 0.04	73.18 ± 0.02	77.94 ± 0.01	77.69 ± 0.07	83.04 ± 0.00
	BT	60.9 ± 0.17	72.04 ± 0.03	77.42 ± 0.01	79.14 ± 0.01	82.28 ± 0.01
	EDA	61.15 ± 0.08	71.09 ± 0.03	72.59 ± 0.23	78.6 ± 0.02	83.5 ± 0.00
STS-B	No DA.	79.49 ± 0.02	85.77 ± 0.01	89.32 ± 0.00	89.94 ± 0.00	90.29 ± 0.00
	BT	79.24 ± 0.01	85.76 ± 0.01	88.80 ± 0.00	89.80 ± 0.00	89.79 ± 0.00
	EDA	78.87 ± 0.01	84.95 ± 0.02	88.82 ± 0.00	89.92 ± 0.00	90.15 ± 0.00

Table 9: ROBERTA-BASE mean test accuracy and the 95% confidence interval for each task, augmentation, and data size, computed over the top 10 best trials, by validation score.