FADA: Crafting Feature-Aware Data Augmentation Policies for Enhanced Text Classification

Anonymous ACL submission

Abstract

This paper introduces FADA, a novel data augmentation technique that creates **f**eature-**a**ware **d**ata **a**ugmentation policies. Unlike traditional dataset-level approaches, FADA utilizes the abstract meaning representation of texts to extract high-level concepts, enabling targeted transformations for specific features. It evaluates transformation effectiveness through cheaply computed quality metrics like label alignment, fluency, and grammaticality. Our evaluations on four benchmark datasets show that our learned augmentation policies attain strong performance against baseline techniques and transfer surprisingly well to new domains.

1 Introduction

Most existing automated data augmentation frameworks produce *dataset-level* policies that do not take into account all the relevant features of the input. However, they do generally produce easily interpretable policies that can support data exploration and debugging. On the other hand, there are approaches that learn *sample-level* policies specific to each instance (Niu and Bansal, 2019; Zhou et al., 2020). These require significantly more computation and often produce policies that are uninterpretable because they are implemented as neural networks. Fortunately, there is a middle ground that is both interpretable and efficient to compute.

In this paper, we propose FADA, a novel **F**eature-**A**ware **D**ata **A**ugmentation technique that efficiently learns when to augment by observing transform-feature interactions. Features, in the context of our approach, refer to distinctive high-level concepts extracted from texts, such as `imperatives`, `negations`, and `polarity`. These features are important because they introduce an additional dimension for discerning the optimal timing and context for effective augmentation. For example, in sentiment analysis, the `WordDeletion` transform can invert meanings in texts with `negations` (e.g. "I do ~~not~~ like standup.") without correctly updating the label. FADA is designed to learn how each transform interacts with various features to ensure their effectiveness during the augmentation process.

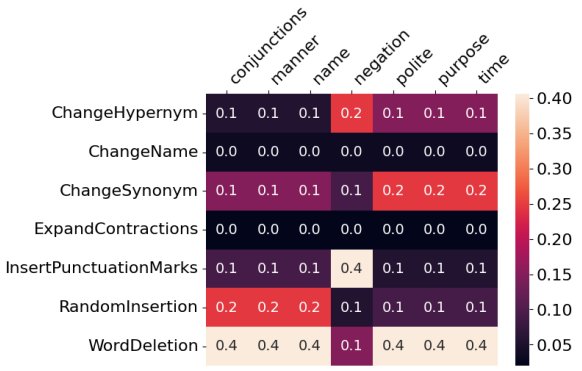

Figure 1: A FADA policy subset, learned for the SST-2 dataset (Socher et al., 2013), determines the likelihood of selecting specific transformations (e.g., WordDeletion) based on given features (e.g., `negation`).

We quantify transformation effectiveness via a set of cheaply computed text quality metrics. The first metric we consider is *label alignment*, i.e. the degree to which the assigned label reflects the semantics of the text, as measured via confident learning (Northcutt et al., 2022). The second is *fluency*, which captures the flow and naturalness of a text represented by an inverted perplexity score. Finally, *grammaticality* is the degree to which the text is correctly structured and is measured by the number of grammatical errors reported by a grammar tool (languagetool, 2023). Each metric is weighted, aggregated, and normalized into a probability of a given transform being effective for input text containing a corresponding feature. Figure 1 shows an example policy where `WordDeletion` is generally effective for most features except negations.

To guide our evaluation of FADA, we formulate two research questions:

RQ1. How does training on FADA augmented

data affect model performance?

RQ2. How well do learned augmentation policies transfer to new datasets and task domains?

Overall, FADA's feature-level policies offer interpretability and insight into the complex interactions between transforms and specific features. They also support targeted instance-level augmentations, allowing for the selection of the most suitable transforms for each text, contingent on its unique features. Notably, learning such a policy is efficiently achievable within just a few hours using a single RTX 2060 GPU. Furthermore, these policies exhibit a remarkable ability to transfer across various datasets and task domains.

2 Approach

FADA efficiently learns augmentation policies at both feature and instance levels, combining targeted transform precision with reduced computational demands. This automated, multi-objective search determines the most effective augmentations by analyzing the effects of transform-feature interactions through heuristic text quality metrics.

As seen in Figure 2, we begin by extracting abstract meaning representation (AMR) graphs (Banarescu et al., 2013) from each text in the training dataset and create a binary feature matrix to identify features in each text. In the search phase, we pair transformations with features and select text instances from the dataset that contain a specific feature. We apply a transformation to these instances and evaluate the changes in text quality based on metrics like alignment, fluency, and grammaticality. Transformations are assessed based on their impact on these quality metrics, with penalties applied for any worsening of the text's original state. We then compute an aggregated quality score for each transformation-feature pair, averaging these scores to update our policy matrix. This matrix, represented as a probability distribution, guides the selection of transformations for texts with specific features. We can also derive an instance-level policy by combining the policy matrix with the feature matrix, allowing for automatic, targeted text augmentations. We continue to sample transform-feature pairs until the policy converges.

Text Transforms. A text transform is simply a function that intakes texts alters them in some way. We focus on edit-based transforms (Wei and Zou, 2019; Xie et al., 2020) that employ simple, label-invariant editing operations like word swaps, deletes, and typo insertion because they are widely

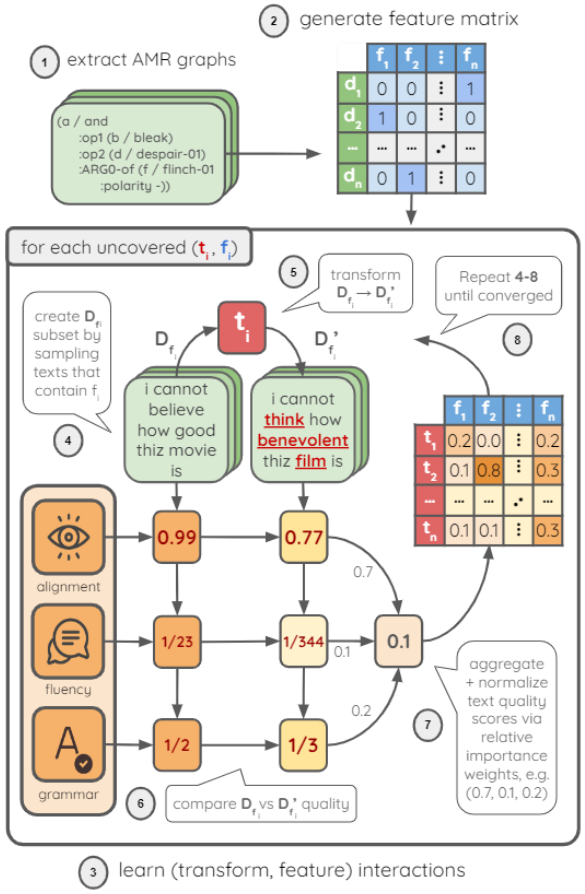

Figure 2: Overview of the FADA search procedure. In this sentiment analysis example, applying transform t_i to a data subset D_{f_i} containing feature f_i, results in adverse impacts to average alignment, fluency, and grammaticality. The transform-feature interaction is aggregated into the augmentation policy with a score of 0.1, which indicates a relatively low probability of sampling t_i for any texts containing f_i.

used and cheap to compute. A detailed list of the 20 edit-based transforms used in this work is available in Appendix C.

Text Features. Our approach utilizes features from AMR graphs (Banarescu et al., 2013; bjascob, 2023) because of their ability to capture both semantic and syntactic text properties. AMR features are also sparse enough to permit sufficient differentiation between texts, which in turn allows FADA to target augmentations with greater nuance.

Quality Metrics. In contrast to previous approaches that rely on compute intensive model training to approximate augmentation effectiveness, we directly evaluate the impact of a transformation text quality. To this end, we chose alignment, fluency, and grammaticality to reflect the intuition that classification performance is improved by intelligible, natural, and well-formed inputs.

Label alignment measures how well labels match

Model / Approach	SST-2	IMDB	AG News	Yahoo! Answers
BERT-tiny	39.9 ± 7.2	35.7 ± 4.7	31.5 ± 1.7	16.6 ± 0.2
+ EDA	37.6 ± 5.7	36.4 ± 4.1	**59.8** ± 4.8	30.6 ± 2.1
+ CheckList	36.9 ± 4.1	40.6 ± 3.7	59.6 ± 0.1	29.2 ± 1.7
+ TAA	37.3 ± 9.2	40.4 ± 8.8	–	–
+ Uniform20	36.6 ± 9.2	45.2 ± 13.2	57.5 ± 0.6	30.1 ± 1.7
+ FADA (Ours)	**42.3** ± 6.0	**50.8** ± 6.8	**59.8** ± 2.8	**31.1** ± 1.5
BERT-base	42.9 ± 9.4	53.7 ± 3.3	71.1 ± 5.2	56.9 ± 1.1
+ EDA	49.6 ± 8.2	53.5 ± 7.5	77.6 ± 2.8	59.6 ± 2.1
+ CheckList	45.3 ± 8.2	52.5 ± 7.5	77.6 ± 3.8	60.9 ± 1.5
+ TAA	57.4 ± 2.2	57.7 ± 4.6	–	–
+ Uniform20	54.2 ± 3.4	56.9 ± 9.3	**79.9** ± 0.3	**61.5** ± 1.4
+ FADA (Ours)	**58.1** ± 9.9	**58.6** ± 5.4	77.4 ± 2.3	61.3 ± 1.2
BERT-large	46.5 ± 0.03	56.9 ± 0.02	74.6 ± 0.02	58.2 ± 0.02
+ EDA	49.1 ± 0.18	59.6 ± 0.01	75.3 ± 0.01	61.4 ± 0.01
+ CheckList	55.6 ± 0.05	59.1 ± 0.04	78.1 ± 0.00	61.7 ± 0.01
+ TAA	62.7 ± 0.07	49.1 ± 0.16	–	–
+ Uniform20	61.4 ± 0.02	**61.7** ± 0.02	79.0 ± 0.02	60.4 ± 0.03
+ FADA (Ours)	**63.5** ± 0.02	60.0 ± 0.02	**79.4** ± 0.01	**61.8** ± 0.02

Table 1: Test F1-score (%) with standard deviation of different augmentation approaches in a low-resource regime. Results are averaged across three runs. All approaches use a 3× augmentation multiplier for fair comparison.

text semantics, as measured by `cleanlab`'s Confident Learning (CL) (cleanlab.ai, 2023; Northcutt et al., 2022). CL detects label errors by comparing noisy and trusted labels. It involves a surrogate model that assesses label confidence, identifying misalignments on a scale from 0 to 1. Fluency in a text is assessed by its naturalness and can be quantified using a language model's perplexity score, like GPT-2 (Radford et al., 2019). Lower perplexity scores indicate more expected, natural text, while higher scores suggest implausible text. Grammaticality assesses a text's adherence to grammar rules, focusing on syntax and word usage. We quantify it using `language-tool` (languagetool, 2023) to count the number of grammatical errors.

3 Experimental Setup

Datasets. We study four benchmark datasets: SST-2 (Socher et al., 2013) and IMDB (Maas et al., 2011) for sentiment analysis, and AG News and Yahoo! Answers (Zhang et al., 2016) for topic detection. Dataset statistics are in Appendix B. Following previous work (Ren et al., 2021; Wei and Zou, 2019; Chen et al., 2020), we focus on low-resource settings, using only 10 examples per class from each dataset, expanded by 3× for both baseline techniques and our FADA approach.

Baselines. We compare FADA with four dataset-level augmentation policies: EDA (Wei and Zou, 2019); CheckList (Ribeiro et al., 2020)[1]; Text AutoAugment (TAA) (Ren et al., 2021)[2]; and Uniform20, which uses the same 20 transforms as FADA, but ignores text features. Details of each transform are in Appendix C.

Text Classification Models. Our experiments utilize three text classification models: BERT-large, BERT-base (Devlin et al., 2019) and BERT-tiny (Turc et al., 2019) as implemented in HuggingFace (Wolf et al., 2020). The selection of these models is motivated by their widespread use and varying sizes, enabling a thorough analysis of how model scale affects performance. Model and training details are in Appendix D.

4 Experimental Results

4.1 Generalization Performance

Table 1 presents the post-training performance of the augmentation techniques we studied to address **RQ1**. Across all models and datasets, FADA consistently achieves the highest or near-highest F1 scores, indicating its effectiveness regardless of the underlying model size. On average, FADA boosted F1 performance by **11.7%**, **7.7%**, and **7.1%** for

[1] As implemented in TextAttack (Morris et al., 2020)

[2] We use the authors' pre-searched policies for SST-2 and IMDB; constructing new policies was time prohibitive.

for each BERT variant (tiny, base, and large), respectively. While other techniques sometimes degraded performance below the no-augmentation baseline, FADA demonstrated its reliability by *always* improving performance by as much as 17% for BERT-large and 28% for BERT-tiny.

4.2 Transferability of FADA Policies

For **RQ2**, we also conducted experiments to study whether the policies learned on one source dataset could be successfully applied to other datasets. As illustrated in Figure 3, policies learned for larger datasets with longer texts, such as Yahoo! Answers and IMDB, were able to make significant improvements when applied to datasets with shorter texts. These improvements even generalized across task domains (i.e. topic classification to sentiment analysis), with the most significant boost of 12.2% observed for SST-2 datasets augmented using the policies learned from Yahoo! Answers.

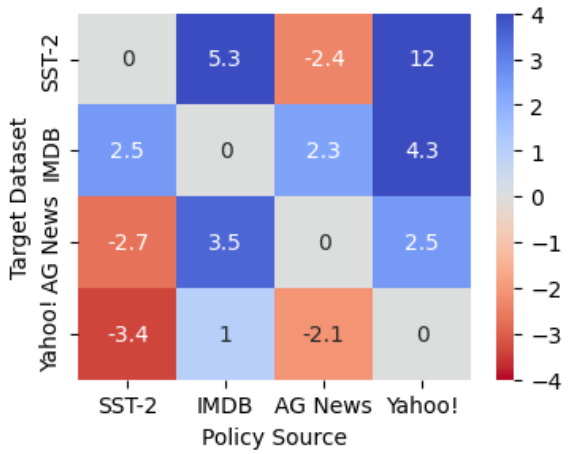

Figure 3: Transferability of FADA policies learned from a source dataset and applied to other target datasets. The numbers denote changes in F1-scores, averaged across all model architectures.

5 Related Work & Discussion

Automated Data Augmentation. In recent years, there has been a growing interest in learning automated data augmentation policies for NLP tasks (Yang et al., 2022). Such policies are a probability distribution over transforms according to which training samples are altered. Especially effective transforms are assigned higher probabilities and harmful transforms are assigned values near zero.

In an early work, Niu and Bansal (2019) adapted AutoAugment (Cubuk et al., 2019) to discover effective augmentation policies for NLP tasks like dialogue generation. While their approach inherits the computational complexities of their predecessor, they are among the first to introduce instance-level augmentation policies conditioned on the text. However, we diverge from previous work in prioritizing interpretability and computational efficiency while learning effective augmentation policies.

Text Quality and Generalization. Several related works applying augmentation to machine translation (Pham et al., 2021; Edunov et al., 2018) have also noted that better data quality did not necessarily lead to stronger models. Pham et al. (2021) suggested that "lower-quality but *more diverse* data often yielded stronger results." Optimal trade-offs between quality and diversity in data metrics indicate "sweet spots" that don't require maximum scores for effective training. For instance, models trained on perfect grammar might underperform on grammatically inconsistent test data. This concept, supported by Fast AutoAugment (Lim et al., 2019) through density matching, advocates for augmentations that make training data more closely mirror validation data.

To better understand the beneficial trade-off between data quality and diversity, we conducted a supplementary analysis of the two aspects. As seen in Table 2 in Appendix A, it is evident that the majority of text augmentations tend to compromise data quality. Notably, the two most effective augmentation strategies, Uniform20 and our proposed FADA approach, demonstrate substantial enhancements to diversity. This finding suggests that the improvements in generalization associated with FADA could be attributed to its effective management of the quality-diversity balance.

6 Conclusion

In this research, we introduced a novel, feature-aware data augmentation framework, tailored to enhance text classification performance. Our approach is designed to optimize the interaction between text transforms and distinct text features, guided by cheaply computed quality metrics. This method maintains the interpretability of dataset-level augmentation policies while facilitating bespoke instance-level transformations tailored to individual training texts. Our empirical results demonstrate that FADA not only consistently enhances performance but also that the learned policies exhibit notable effectiveness when applied to unfamiliar datasets and domains.

7 Limitations

There are several limitations in this work, especially relating to the selection of particular features, transforms, and quality metrics. First, we acknowledge that, like most NLP research, FADA is heavily biased towards the English language. The transforms we study generally expect English inputs, as do the AMR models we use to extract linguistic features. However, the quality metrics can be altered to support multilingual analysis by swapping the base model for perplexity and using a different language argument when initializing the grammar checker. Additionally, label alignment requires the existence of an already fine-tuned surrogate model which may not exist for all tasks. However, with the massive growth of model repositories, such as HuggingFace with over 120k models, it becomes increasingly likely that a useful surrogate exists for most tasks.

Second, the quality metrics we selected represent a relatively small cross section of available options. For example, we could have also explored the use of text diversity, coherence, factuality, informativeness, and so on. We ultimately decided against exploring other quality metrics to minimize policy construction time.

Lastly, scoping the experiments to a low-resource setting potentially limits the generalization of our main findings. Optimistically, the difficulty of training models in a low-resource setting is likely to be the biggest motivation to use any kind of data augmentation in the first place.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

bjascob. 2023. amrlib.

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

cleanlab.ai. 2023. cleanlab.

Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. 2019. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 113–123. Computer Vision Foundation / IEEE.

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

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale.

Fabrice Harel-Canada, Muhammad Ali Gulzar, Nanyun Peng, and Miryung Kim. 2022. Sibylvariant transformations for robust text classification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1771–1788, Dublin, Ireland. Association for Computational Linguistics.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

languagetool. 2023. languagetool.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. 2019. Fast autoaugment.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1317–1323, Hong Kong, China. Association for Computational Linguistics.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2022. Confident learning: Estimating uncertainty in dataset labels.

Hieu Pham, Xinyi Wang, Yiming Yang, and Graham Neubig. 2021. Meta back-translation.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shuhuai Ren, Jinchao Zhang, Lei Li, Xu Sun, and Jie Zhou. 2021. Text AutoAugment: Learning compositional augmentation policy for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9029–9043, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc.

Zihan Yang, Richard O. Sinnott, James Bailey, and Qiuhong Ke. 2022. A survey of automated data augmentation algorithms for deep learning-based image classification tasks.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification.

Fengwei Zhou, Jiawei Li, Chuanlong Xie, Fei Chen, Lanqing Hong, Rui Sun, and Zhenguo Li. 2020. Metaaugment: Sample-aware data augmentation policy learning.

A Augmentation's Impact on Text Quality

We studied the effect different augmentation approaches have on improving or degrading the quality of text data. For comparison purposes, we add a Dist-2 (Li et al., 2016) diversity metric, which measures the number of distinct bi-grams. For each baseline and dataset, we transform the texts and then compute alignment, fluency, grammaticality impact scores. These scores are normalized into ratios where values greater than 1 indicate improvements to that quality dimension relative to the original dataset and vice versa. For FADA, we performed a grid-search over quality weights — w_a, w_f, w_g — to generate 10 different augmentation policies (and datasets) and average the resulting scores.

Table 2 shows the relative impact each augmentation approach had our studied quality metrics across all datasets. With few exceptions, all augmentation frameworks tended to decrease text quality. For alignment and fluency, EDA and CheckList score more highly, indicating that they better preserve the original meaning and naturalness of the text. This result may be explained by the fact that their underlying transforms are limited to making smaller edits less capable of injecting as much lexical diversity into the datasets. In contrast, FADA and Uniform20 exhibit significant diversity-quality tradeoffs. These results represent a starting point for understanding the relationship between data quality and model generalization.

Approach	Align.	Fluency	Gram.	Dist-2
Original	1	1	1	3372
EDA	1	1.02	0.83	3990
Checklist	1	0.95	0.82	3850
TAA	0.93	0.15	1.12	3996
Uniform20	0.94	0.44	0.55	5111
FADA(Ours)	0.91	0.46	0.58	5089

Table 2: Relative impact different augmentations have on our studied text quality metrics & Dist-2 as a comparative diversity metric, averaged across all datasets. Scores larger than 1 indicate that the metric had increased after augmentation. This is possible, for example, if a transform like `RandomDeletion` removed a word that caused a grammar / fluency issue in the original text.

B Benchmark Dataset Statistics

Table 3 shows various statistics for the datasets used in our experiments. Note that for SST-2, the test labels are officially hidden and scores can only be attained by submitting to the GLUE (Wang et al., 2018) benchmark. As a workaround, we evenly split the validation dataset and use the latter half for testing.

C Studied Transform Descriptions

We conducted our policy search over 20 transforms implemented in the SIBYL augmentation tool (Harel-Canada et al., 2022). Table 4 shows brief descriptions of each transform.

D Model and Training Settings

BERT-large (`bert-large-uncased`), BERT-base (`bert-base-uncased`), and BERT-tiny (`prajjwal1/bert-tiny`) are different sizes of the same encoder-only BERT architecture. BERT-large has 24 transformer layers, a hidden representation size of 1024, 16 attention heads, and 336M parameters. BERT-base has 12 transformer layers, a hidden representation size of 768, and 12 attention heads. Lastly, BERT-Tiny has 2 transformer layers, a hidden representation size of 128, and an unspecified number of attention heads. Smaller BERT models like BERT-Tiny are intended for environments with restricted computational resources, like cellular devices. They can be fine-tuned in the same manner as the original BERT models.

We used HuggingFace (Wolf et al., 2020) to fine-tune our models on the training dataset. We made heavy use of the best practices encoded in the defaults of their `Trainer` class, though with several customized settings. We set the initial learning rate to 0.001 with a weight decay of 0.01. The batch size for training and evaluation were set to 4 and 16, respectively. The training process was run for a maximum of 10 epochs, with evaluation performed at the end of each epoch and early stopping if validation loss could not be improved after 5 epochs. The best performing model checkpoint was saved and used for evaluation on the test dataset.

Dataset	Source	Task	Subject	Classes	# Train	# Test	Avg Len
AG News	(Zhang et al., 2016)	Topic	News Articles	4	25,000	3,800	38
Yahoo! Answers	(Zhang et al., 2016)	Topic	QA Posts	10	1,400,000	60,000	92
SST-2	(Socher et al., 2013)	Sentiment	Movies Reviews	2	70,000	436	18
IMDB	(Maas et al., 2011)	Sentiment	Movies Reviews	2	25,000	25,000	234

Table 3: Dataset statistics.

Transform Name	Description	Source
AddNeutralEmoji	Appends a random emoji with neutral sentiment	1
RemoveNeutralEmoji	Removes all emojies judged to exhibit neutral sentiment	1
ChangeHypernym	Randomly replace words with less specific words (e.g. turban → headwear)	2
ChangeHyponym	Randomly replace words with more specific words (e.g. fruit → apple)	2
ChangeLocation	Randomly change city and country names	2
ChangeName	Randomly change integers to other integers within 20% of the original	2
ChangeNumber	Randomly change names with some other names	2
ChangeSynonym	Randomly replaces words with approximate equivalents	2
ContractContractions	Contracts expanded contractions in a sentence (if any)	2
ExpandContractions	Expands contractions in a sentence (if any)	2
HomoglyphSwap	Replaces English characters with visually similar homoglyphs	3
RandomCharDel	Randomly deletes characters	3
RandomCharInsert	Randomly inserts characters	3
RandomCharSubst	Randomly substitutes characters	3
RandomCharSwap	Randomly swaps two adjacent characters	3
RandomInsertion	Randomly inserts a synonym of some word to a new position	3
RandomSwapQwerty	Randomly swaps charcters with others adjacent on a QWERTY keyboard	3
InsertPunctuationMarks	Randomly inserts various punctuation marks	4
RandomSwap	Randomly swaps two adjacent words	5
WordDeletion	Randomly deletes words	5

Table 4: Descriptions of all the text transforms in the FADA search space. Sources: 1. SIBYL (Harel-Canada et al., 2022) 2. CHECKLIST (Ribeiro et al., 2020) 3. TEXTATTACK (Morris et al., 2020) 4. AEDA (Karimi et al., 2021) 5. EDA (Wei and Zou, 2019). Note that source attributions are based on implementation details, not necessarily where the transformation was initially proposed.