AutoAugment Is What You Need: Enhancing Rule-based Augmentation Methods in Low-resource Regimes

Juhwan Choi[1], **Kyohoon Jin**[2], **Junho Lee**[1], **Sangmin Song**[1] and **Youngbin Kim**[1,2]

[1]Department of Artificial Intelligence, Chung-Ang University
[2]Graduate School of Advanced Imaging Sciences, Multimedia and Film, Chung-Ang University
{gold5230,fhzh123,jhjo32,s2022120859,ybkim85}@cau.ac.kr

Abstract

Text data augmentation is a complex problem due to the discrete nature of sentences. Although rule-based augmentation methods are widely adopted in real-world applications because of their simplicity, they suffer from potential semantic damage. Previous researchers have suggested easy data augmentation with soft labels (softEDA), employing label smoothing to mitigate this problem. However, finding the best factor for each model and dataset is challenging; therefore, using softEDA in real-world applications is still difficult. In this paper, we propose adapting AutoAugment to solve this problem. The experimental results suggest that the proposed method can boost existing augmentation methods and that rule-based methods can enhance cutting-edge pre-trained language models. We offer the source code.[1]

1 Introduction

Data augmentation is a regularization strategy that improves model performance expanding the data held in various ways (Hernández-García and König, 2018). In the natural language processing (NLP) field, data augmentation is used in various fields to alleviate data shortages, and various augmentation methods have been proposed accordingly (Feng et al., 2021; Li et al., 2022). For example, image data can be augmented by applying simple rules, such as flipping and rotation, to image data (Yang et al., 2022), and text data can also be augmented, by simple rules such as replacing synonyms and changing the order between words (Zhang et al., 2015; Wei and Zou, 2019). In addition, a method for augmenting data by generating new text using various deep learning models has also been proposed (Sennrich et al., 2016; Wu et al., 2019; Anaby-Tavor et al., 2020; Yoo et al., 2021; Zhou et al., 2022; Dai et al., 2023).

However, as these methods often demand training data for fine-tuning before augmentation (Zhang et al., 2022; Li et al., 2022), it may be challenging to apply them in a low-resource environment (Hu et al., 2019; Bayer et al., 2022; Kim et al., 2021). Rule-based text data augmentation methods are less costly and easy to implement; thus, they are often used in real-world problems. Despite that, the previously proposed rule-based text data augmentation methods risk not maintaining semantic consistency with original data, which is different from image data (Zhao et al., 2022), leading to performance degradation. To relieve this problem, methods that perform data augmentation only through random insertion of punctuation marks have also been proposed (Karimi et al., 2021), but they introduce fewer variations compared to easy data augmentation (EDA). Recently, softEDA (Choi et al., 2023), a method applying label smoothing (Szegedy et al., 2016) to the augmented data, was proposed to alleviate these drawbacks.

In softEDA, a heuristic grid search was performed for the label smoothing factor (a hyperparameter for performing label smoothing). However, the method based on a heuristic search has the following disadvantages. First, a heuristic search is expensive to execute (Bergstra and Bengio, 2012). Second, although we found the best factor value of the grid, it may not be the global optimum. There could be a better value outside the heuristic search grid; thus, revealing the possible performance gain is difficult.

This paper proposes a method to apply AutoAugment (Cubuk et al., 2019), a technique to determine the optimal factors in the data augmentation process to alleviate the limitations of previous softEDA methods. By optimizing various arguments of softEDA, it is shown that stable and effective performance improvement is possible compared to the existing rule-based strategy with static factors.

[1]https://github.com/c-juhwan/soft-text-autoaugment

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 1–8
March 21-22, 2024

In addition, the existing softEDA experiment was conducted on an entire dataset. However, more severe overfitting occurs when the given training data are insufficient (Althnian et al., 2021), and the scope of performance improvement is greater when additional training data are obtained from a small dataset (Prusa et al., 2015; Okimura et al., 2022), so data augmentation becomes increasingly crucial in this low-resource environment. Therefore, this study evaluates the proposed method under a low-resource scenario and demonstrates that the proposed method is effective even under data-scarce conditions. In addition, some existing studies have argued that simple rule-based augmentation strategies are less effective in improving the performance of pre-trained language models (PLMs) (Longpre et al., 2020; Zhang et al., 2022; Pluščec and Šnajder, 2023). In this study, we show that through argument optimization, it is possible to improve the performance of not only BERT (Devlin et al., 2019), the standard PLM, but also DeBERTaV3 (He et al., 2023), a cutting-edge PLM, through rule-based data augmentation.

2 Related Work

Data augmentation of text is primarily performed by augmenting data according to predetermined rules (Zhang et al., 2015; Belinkov and Bisk, 2018; Wei and Zou, 2019; Karimi et al., 2021; Choi et al., 2023) or using various deep learning models (Sennrich et al., 2016; Wu et al., 2019; Anaby-Tavor et al., 2020; Yoo et al., 2021; Zhou et al., 2022; Dai et al., 2023). Rule-based data augmentation methods generate new data by performing perturbation in various ways, such as replacing some of the words in a given sentence with synonyms (Zhang et al., 2015) or inserting typos at the character level (Belinkov and Bisk, 2018). The easy data augmentation (EDA) (Wei and Zou, 2019) technique is a representative rule-based data augmentation method consisting of synonym replacement (SR), random insertion (RI), random swap (RS), and random deletion (RD). However, because such random changes can lead to the loss of semantic consistency, the "an easier data augmentation" (AEDA) technique (Karimi et al., 2021) consisting only of the RI of six punctuation marks has also been proposed. The softEDA (Choi et al., 2023) method compensates for the semantic damage caused by EDA by applying label smoothing to the augmented data.

Model-based augmentation methods employ deep learning models to generate new data. Back-translation (Sennrich et al., 2016) is one of the early model-based methods. It first translates the given data into another language and back-translates it to the original language, generating different expressions with the same concept. Methods based on PLM have also been proposed, and C-BERT (Wu et al., 2019), LAMBADA (Anaby-Tavor et al., 2020), and FlipDA (Zhou et al., 2022) generate new data using BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and T5 (Raffel et al., 2020), respectively. In addition, GPT-3 (Brown et al., 2020) and ChatGPT, which are larger than these PLMs, have been proposed to generate new data (Yoo et al., 2021; Dai et al., 2023). Other researchers have introduced Mixup (Zhang et al., 2018) strategy to the NLP field to augment text data (Guo, 2020; Sun, 2020; Yoon et al., 2021).

Moreover, some previous approaches have tried to apply AutoAugment for NLP. Text AutoAugment (Ren et al., 2021), the work closest to the proposed method, suggested applying AutoAugment to optimize hyperparameters for data augmentation. In addition, DND (Kim et al., 2022) incorporated various data augmentation methods and suggested optimizing two reward terms regarding the difficulty and consistency with the original data. While the proposed work uses AutoAugment to optimize augmentation hyperparameters, we also focus on optimizing label smoothing values for the original and augmented data.

3 Method

3.1 Preliminaries

The EDA (Wei and Zou, 2019) method comprises four aforementioned suboperations: SR, RI, RS, and RD. First, SR randomly selects several words in a given sentence and changes them into their synonyms. Second, RI selects a random word in the sentence and inserts its synonym at a random position in the sentence. Third, RS operation randomly selects two words in the sentence and changes their positions. Finally, RD removes each word from the sentence with a predefined probability.

Through these four suboperations, EDA introduces noise to the original data and generates augmented data. Each suboperation has a magnitude of perturbation. For instance, in the case of SR, a higher magnitude leads to the additional replacement of the original words with their synonyms.

For each observed data pair $(\mathbf{x}, \mathbf{y})$ in the original dataset $\mathcal{D}$, where $\mathbf{x}$ denotes an input sentence and $\mathbf{y}$ represents the corresponding label value, the process of EDA can be formulated as follows:

$$\hat{\mathbf{x}} = \texttt{EDA}(\mathbf{x}, p_{\texttt{EDA}}) = \begin{cases} \texttt{SR}(\mathbf{x}, \alpha_{\texttt{SR}}) \\ \texttt{RI}(\mathbf{x}, \alpha_{\texttt{RI}}) \\ \texttt{RS}(\mathbf{x}, \alpha_{\texttt{RS}}) \\ \texttt{RD}(\mathbf{x}, \alpha_{\texttt{RD}}) \end{cases} \quad (1)$$

where $\{\alpha_{\texttt{SR}}, \alpha_{\texttt{RI}}, \alpha_{\texttt{RS}}, \alpha_{\texttt{RD}}\}$ denotes the magnitude of each suboperation, and $p_{\texttt{EDA}} = \{p_{\texttt{SR}}, p_{\texttt{RI}}, p_{\texttt{RS}}, p_{\texttt{RD}}\}$ represents the probability distribution of each suboperation to be selected, which are equal and sum to one. As indicated, EDA only modifies $\mathbf{x}$, and the label of augmented data is the same as for $\mathbf{y}$.

The softEDA (Choi et al., 2023) is a technique that incorporates noise into the label of augmented data through label smoothing (Szegedy et al., 2016). While softEDA follows the previous EDA to augment $\hat{\mathbf{x}}$, the following equation defines the process of softEDA, generating a label for augmented data $\hat{\mathbf{y}}$:

$$\begin{aligned} \hat{\mathbf{y}} &= (1 - \epsilon_{aug})\mathbf{y} + \frac{\epsilon_{aug}}{N_{Class}} \\ &= \begin{cases} (1 - \epsilon_{aug}) + \frac{\epsilon_{aug}}{N_{Class}} & \text{if } y = y_i \\ \frac{\epsilon_{aug}}{N_{Class}} & \text{Otherwise} \end{cases} \end{aligned} \quad (2)$$

where ϵ_{aug} is a smoothing factor for label smoothing.

3.2 Proposed Method

Previous EDA and softEDA have numerous augmentation hyperparameters and were primarily fixed or heuristically searched. This paper proposes a method to optimize these hyperparameters by adapting AutoAugment. First, we defined an augmentation policy $\mathcal{P}$ with various factors:

$$\begin{aligned} \mathcal{P} = \{&p_{aug}, p_{\texttt{SR}}, p_{\texttt{RI}}, p_{\texttt{RS}}, p_{\texttt{RD}}, \\ &\alpha_{\texttt{SR}}, \alpha_{\texttt{RI}}, \alpha_{\texttt{RS}}, \alpha_{\texttt{RD}}, \\ &N_{aug}, \epsilon_{ori}, \epsilon_{aug}\} \end{aligned} \quad (3)$$

where p_{aug} indicates the probability of augmentation, N_{aug} refers to the amount of augmented data per original data point, ϵ_{ori} represents a label smoothing factor for the original data, different from ϵ_{aug}. Following Text AutoAugment (Ren et al., 2021), we optimized the proposed policy based on sequential model-based global optimization (Bergstra et al., 2011). Finding the optimal augmentation parameter for each model and dataset through this adaptation of AutoAugment with softEDA is more beneficial than inefficient grid search.

4 Experiment

4.1 Datasets and Low-resource Setting

Eight text classification datasets were used to evaluate the proposed method. The SST2, SST5 (Socher et al., 2013) and MR (Pang et al., 2002) sentiment classification tasks are from movie reviews. The CoLA (Warstadt et al., 2019) binary classification dataset measures the linguistic acceptability of a given sentence. The SUBJ (Pang and Lee, 2004) binary classification dataset deals with the subjectivity of a sentence. PC (Ganapathibhotla and Liu, 2008), and CR (Hu and Liu, 2004; Liu et al., 2015) are datasets constructed from customer reviews. In addition, the TREC (Li and Roth, 2002) multiclass text classification dataset is about the question type of given text. Dataset specifications can be found in Appendix A.

Data augmentation becomes more important when the given data is deficient than when sufficient data can be accessed (Chen et al., 2023). To simulate a more challenging scenario, we evaluated the proposed method with only 100 and 500 randomly selected original data from each dataset.

4.2 Baselines

To validate the claim that hyperparameter optimization for the augmentation method is effective in enhancing model performance, we compared our approach with previous rule-based data augmentation methods with fixed hyperparameters. We compared the proposed method against the previous EDA, AEDA, and softEDA methods with fixed hyperparameters.

Recent studies suggest that simple rule-based augmentation methods are insufficient to enhance PLM-based models (Longpre et al., 2020; Zhang et al., 2022; Pluščec and Šnajder, 2023). In addition, validating the newly proposed augmentation method using cutting-edge models, not just models like BERT, is necessary (Zhou et al., 2022). Therefore, we adopted BERT and DeBERTaV3 (He et al., 2023), an improvement of DeBERTa (He et al., 2021) as the baseline model for evaluation.

	SST2	SST5	CoLA	SUBJ	TREC	MR	CR	PC
BERT w/o Aug	$80.46_{1.84}$	$35.13_{0.74}$	$71.49_{1.40}$	$92.85_{0.44}$	$78.42_{1.30}$	$72.11_{1.39}$	$79.88_{0.82}$	$88.12_{0.58}$
	$86.08_{1.03}$	$43.64_{0.50}$	$75.50_{0.58}$	$95.07_{\mathbf{0.22}}$	$93.27_{0.42}$	$81.29_{0.52}$	$87.53_{0.60}$	$91.15_{0.21}$
w/ EDA	$80.76_{1.39}$	$36.63_{1.33}$	$70.70_{0.98}$	$93.39_{\mathbf{0.25}}$	$81.56_{1.71}$	$73.18_{1.36}$	$79.54_{1.15}$	$89.64_{0.80}$
	$86.71_{0.63}$	$45.08_{1.16}$	$73.18_{0.52}$	$94.69_{0.33}$	$93.99_{1.05}$	$80.41_{0.29}$	$87.71_{0.57}$	$90.81_{0.40}$
w/ AEDA	$80.96_{1.63}$	$36.54_{0.97}$	$72.24_{1.85}$	$93.29_{0.23}$	$81.27_{2.19}$	$74.37_{2.84}$	$80.67_{1.64}$	$88.75_{0.90}$
	$86.66_{0.63}$	$44.53_{1.02}$	$74.44_{0.41}$	$94.60_{0.48}$	$93.87_{0.75}$	$81.57_{\mathbf{0.15}}$	$87.66_{0.55}$	$91.03_{0.31}$
w/ softEDA	$80.80_{3.22}$	$37.13_{1.60}$	$72.41_{0.95}$	$93.24_{0.40}$	$82.92_{1.70}$	$74.40_{1.27}$	$78.95_{2.65}$	$88.82_{1.63}$
	$87.84_{0.65}$	$45.04_{1.28}$	$74.16_{0.99}$	$94.85_{0.39}$	$94.68_{0.51}$	$81.16_{0.88}$	$87.94_{0.85}$	$91.12_{0.63}$
w/ Ours	$\mathbf{85.48}_{0.57}$	$\mathbf{39.88}_{0.41}$	$\mathbf{74.63}_{\mathbf{0.33}}$	$\mathbf{94.10}_{0.35}$	$\mathbf{85.88}_{\mathbf{1.06}}$	$\mathbf{79.32}_{\mathbf{0.37}}$	$\mathbf{86.49}_{\mathbf{0.22}}$	$\mathbf{91.54}_{\mathbf{0.11}}$
	$\mathbf{88.53}_{\mathbf{0.27}}$	$\mathbf{46.16}_{0.63}$	$\mathbf{76.66}_{0.81}$	$\mathbf{95.54}_{0.33}$	$\mathbf{95.17}_{0.54}$	$\mathbf{83.10}_{0.34}$	$\mathbf{89.98}_{\mathbf{0.25}}$	$\mathbf{92.16}_{0.19}$
w/ Ours w/o LS	$84.71_{\mathbf{0.44}}$	$39.22_{\mathbf{0.38}}$	$73.80_{0.79}$	$93.71_{0.35}$	$84.85_{1.40}$	$77.86_{0.53}$	$85.70_{0.88}$	$91.13_{0.19}$
	$88.13_{0.48}$	$45.45_{\mathbf{0.39}}$	$76.30_{\mathbf{0.34}}$	$95.15_{\mathbf{0.22}}$	$94.70_{\mathbf{0.46}}$	$82.19_{0.60}$	$89.66_{0.35}$	$91.98_{\mathbf{0.18}}$
DeBERTaV3 w/o Aug	$88.36_{0.36}$	$35.95_{1.69}$	$72.62_{4.24}$	$92.23_{0.24}$	$80.19_{3.23}$	$82.84_{0.39}$	$85.61_{1.20}$	$91.22_{0.43}$
	$92.59_{0.73}$	$48.77_{1.52}$	$82.21_{0.82}$	$94.66_{0.22}$	$94.06_{0.43}$	$86.22_{0.37}$	$91.40_{0.36}$	$91.85_{0.26}$
w/ EDA	$86.61_{0.70}$	$37.64_{1.23}$	$74.83_{1.10}$	$92.85_{0.48}$	$83.65_{1.84}$	$83.18_{\mathbf{0.32}}$	$84.86_{0.73}$	$90.51_{0.47}$
	$93.25_{0.55}$	$49.04_{0.78}$	$79.24_{0.66}$	$94.81_{0.53}$	$94.33_{0.99}$	$86.71_{0.65}$	$91.24_{0.39}$	$92.3_{0.15}$
w/ AEDA	$88.44_{0.80}$	$36.87_{2.88}$	$79.29_{0.65}$	$92.81_{0.47}$	$84.17_{0.79}$	$82.87_{0.75}$	$85.76_{1.37}$	$90.61_{0.49}$
	$92.54_{0.78}$	$49.16_{0.83}$	$82.78_{\mathbf{0.40}}$	$94.92_{0.58}$	$94.45_{0.80}$	$85.77_{1.63}$	$91.09_{0.49}$	$92.29_{\mathbf{0.11}}$
w/ softEDA	$88.94_{1.03}$	$38.37_{1.65}$	$79.40_{1.51}$	$92.90_{1.08}$	$84.58_{1.29}$	$83.50_{0.65}$	$86.33_{1.65}$	$91.28_{0.82}$
	$93.12_{1.05}$	$50.34_{1.44}$	$78.97_{1.16}$	$94.77_{0.21}$	$94.71_{0.69}$	$87.02_{0.50}$	$91.81_{0.76}$	$92.16_{0.20}$
w/ Ours	$\mathbf{91.38}_{0.32}$	$\mathbf{42.92}_{0.52}$	$\mathbf{82.56}_{0.51}$	$\mathbf{94.47}_{0.26}$	$\mathbf{87.70}_{0.90}$	$\mathbf{85.31}_{0.79}$	$\mathbf{89.95}_{\mathbf{0.51}}$	$\mathbf{92.32}_{\mathbf{0.19}}$
	$\mathbf{93.94}_{\mathbf{0.30}}$	$\mathbf{52.77}_{\mathbf{0.62}}$	$\mathbf{84.32}_{0.49}$	$\mathbf{95.29}_{0.31}$	$\mathbf{94.92}_{0.62}$	$\mathbf{87.96}_{\mathbf{0.17}}$	$\mathbf{92.46}_{\mathbf{0.18}}$	$\mathbf{92.72}_{0.40}$
w/ Ours w/o LS	$90.47_{\mathbf{0.26}}$	$42.44_{\mathbf{0.49}}$	$82.10_{\mathbf{0.43}}$	$94.22_{\mathbf{0.15}}$	$86.57_{\mathbf{0.61}}$	$85.07_{0.58}$	$89.47_{0.67}$	$92.22_{0.21}$
	$93.40_{0.58}$	$52.54_{0.66}$	$83.67_{0.86}$	$95.15_{\mathbf{0.12}}$	$94.92_{\mathbf{0.18}}$	$87.41_{0.37}$	$92.28_{0.27}$	$92.49_{0.33}$

Table 1: Experimental results. Each experiment has been repeated five times and the statistics are presented in $mean_{std}$ format. The upper side of each column denotes the results when $N_{Train} = 100$, and the lower side shows the results when $N_{Train} = 500$. The best mean and standard deviation values for each model and dataset are boldfaced. Results that reported a lower mean value than the baseline are gray.

4.3 Main Results

Table 1 reports the experimental results. Previously proposed augmentation methods have faced marginal gain, or even performance degradation. Especially, softEDA has a high standard deviation compared to other methods, indicating that softEDA has difficulty being effective within a single fixed hyperparameter and requires optimization for hyperparameters. Whereas, the proposed method exhibits a stable and remarkable performance improvement within every setting, including those where other methods had performance degradation or marginal gains. This finding suggests enhancing extensive and cutting-edge PLMs with simple augmentation methods is achievable under the carefully designed data augmentation policy and hyperparameter optimization strategy. Furthermore, it is shown that our strategy has remarkably low standard deviation values compared to other techniques, showcasing that our approach is robust against statistical differences and valuable for practical application in low-resource text classification problems.

4.4 Ablation Study

One may wonder whether the performance improvement reported in Table 1 is solely caused by the adaptation of AutoAugment, rather than the label smoothing of softEDA. To validate the effectiveness of label smoothing, we conducted an ablation study where label smoothing is not applied (i.e., $\epsilon_{ori} = \epsilon_{aug} = 0$). This setting is equal to optimizing only factors of EDA. "w/ Ours w/o LS" row of Table 1 presents the experimental results, revealing that the proposed method without label smoothing is less effective than the proposed method. This finding supports that the label smoothing optimization introduced by softEDA plays a crucial role in enhancing the model.

5 Conclusion

This paper proposed a method to optimize various hyperparameters of rule-based text augmentation methods. The experimental results suggest that the proposed method is effective and stable, and that rule-based augmentation methods can improve cutting-edge PLMs with proper hyperparameter optimization. Future work may extend this approach to other tasks, such as natural language inference, which is more complex than the single-sentence classification conducted in this paper.

Limitations

This paper used AutoAugment to optimize the rule-based data augmentation method. The primary weakness of AutoAugment is the computational overhead from the searching process (Zhang and Ma, 2022). However, under low-resource situations, where the necessity of data augmentation is emphasized, this problem can be diminished as the time consumption of the search process decreases.

Ethics Statement

This paper proposes an optimized rule-based augmentation method. These rule-based methods are more ethically stable than model-based approaches, as the modification is performed under predefined rules. For example, back-translation can be easily exposed to the potential bias of the translation model. Methods based on PLMs also share this concern. However, rule-based augmentation methods, including the proposed method, perform modifications within a given sentence and are less likely to be exposed to unintentional bias.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2022R1C1C1008534), and Institute for Information & communications Technology Planning & Evaluation (IITP) through the Korea government (MSIT) under Grant No. 2021-0-01341 (Artificial Intelligence Graduate School Program, Chung-Ang University).

References

Alhanoof Althnian, Duaa AlSaeed, Heyam Al-Baity, Amani Samha, Alanoud Bin Dris, Najla Alzakari, Afnan Abou Elwafa, and Heba Kurdi. 2021. Impact of dataset size on classification performance: an empirical evaluation in the medical domain. *Applied Sciences*, 11(2):796.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Juhwan Choi, Kyohoon Jin, Junho Lee, Sangmin Song, and YoungBin Kim. 2023. Softeda: Rethinking rule-based data augmentation with soft labels. In *ICLR 2023 Tiny Papers*.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

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.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 241–248.

Hongyu Guo. 2020. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4044–4051.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Alex Hernández-García and Peter König. 2018. Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 168–177.

Zhiting Hu, Bowen Tan, Russ R Salakhutdinov, Tom M Mitchell, and Eric P Xing. 2019. Learning data manipulation for augmentation and weighting. *Advances in Neural Information Processing Systems*, 32.

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.

Jaehyung Kim, Dongyeop Kang, Sungsoo Ahn, and Jinwoo Shin. 2022. What makes better augmentation strategies? augment difficult but not too different. In *International Conference on Learning Representations*.

Yekyung Kim, Seohyeong Jeong, and Kyunghyun Cho. 2021. Linda: Unsupervised learning to interpolate in natural language processing. *arXiv preprint arXiv:2112.13969*.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. *AI Open*, 3:71–90.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. 2018. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated rule selection for aspect extraction in opinion mining. In *Proceedings of the 24th International Conference on Artificial Intelligence*, page 1291–1297.

Shayne Longpre, Yu Wang, and Chris DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4401–4411.

Vukosi Marivate and Tshephisho Sefara. 2020. Improving short text classification through global augmentation methods. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 385–399. Springer.

Itsuki Okimura, Machel Reid, Makoto Kawano, and Yutaka Matsuo. 2022. On the impact of data augmentation on downstream performance in natural language processing. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 88–93.

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 of the Association for Computational Linguistics (ACL-04)*, pages 271–278.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Domagoj Pluščec and Jan Šnajder. 2023. Data augmentation for neural nlp. *arXiv preprint arXiv:2302.11412*.

Joseph Prusa, Taghi M Khoshgoftaar, and Naeem Seliya. 2015. The effect of dataset size on training tweet sentiment classifiers. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 96–102. IEEE.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

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.

Congyingand Yin Wenpengand Liang Tingtingand Yu Philipand He Lifang Sun, Lichaoand Xia. 2020. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV 19*, pages 84–95. Springer.

Suorong Yang, Weikang Xiao, Mengcheng Zhang, Suhan Guo, Jian Zhao, and Furao Shen. 2022. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239.

Soyoung Yoon, Gyuwan Kim, and Kyumin Park. 2021. Ssmix: Saliency-based span mixup for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3225–3234.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Le Zhang, Zichao Yang, and Diyi Yang. 2022. Treemix: Compositional constituency-based data augmentation for natural language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5243–5258.

Linfeng Zhang and Kaisheng Ma. 2022. A good data augmentation policy is not all you need: A multi-task learning perspective. *IEEE Transactions on Circuits and Systems for Video Technology*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Minyi Zhao, Lu Zhang, Yi Xu, Jiandong Ding, Jihong Guan, and Shuigeng Zhou. 2022. Epida: An easy plug-in data augmentation framework for high performance text classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4742–4752.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022. Flipda: Effective and robust data augmentation for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8646–8665.

A Dataset Specification

Dataset	N_{Class}	N_{Train}	N_{Test}
SST2	2	6.9K	1.8K
SST5	5	8.5K	2.2K
CoLA	2	8.5K	0.5K
SUBJ	6	8K	2K
TREC	2	5.5K	0.5K
MR	2	9.5K	1.1K
CR	2	3.0K	0.8K
PC	2	39K	4.5K

Table 2: Specification of each dataset used for the experiment.

	SST2	CR	MR	TREC	SUBJ	PC	CoLA
BERT w/o Aug	89.74	89.08	84.28	95.47	96.18	93.44	75.38
w/ EDA	+0.71	-0.41	-0.92	+0.51	-0.35	+0.58	-0.45
w/ AEDA	+0.22	+1.84	**+0.19**	-0.67	-0.30	-0.15	-0.34
w/ softEDA 0.1	-0.11	+0.29	-1.10	-1.45	**+0.15**	+0.43	+1.34
w/ softEDA 0.15	-0.22	+0.66	-0.46	-0.47	-0.50	-0.01	+0.02
w/ softEDA 0.2	-0.12	**+2.10**	**+0.19**	-0.27	+0.05	+0.43	+0.81
w/ softEDA 0.25	-0.23	**+2.10**	-0.92	**+1.17**	-0.10	**+0.67**	**+1.50**
w/ softEDA 0.3	**+0.83**	-0.90	-1.80	-0.78	+0.00	**+0.67**	+0.23

Table 3: Results of softEDA for the BERT model reported in the softEDA paper. The best scores for each dataset are boldfaced. Scores lower than the baseline are gray.

B Implementation Details

We used PyTorch (Paszke et al., 2019) and Huggingface Transformers (Wolf et al., 2020) to implement the model and evaluation process. We used `bert-base-cased` and `microsoft/deberta-v3-base` for the BERT and DeBERTaV3 models. Every model was trained using the Adam optimizer with a batch size of 32 and a learning rate of 5e-5 for ten epochs, with early stopping with a patience value of 5, conditioned on best validation accuracy. The training procedure was performed on a single Nvidia RTX 3090 GPU.

For the baseline method implementation, we used TextAugment library (Marivate and Sefara, 2020) for EDA, and softEDA was built on it. The library did not have an implementation for AEDA; thus, we implemented it separately. We used ray tune (Liaw et al., 2018) to implement the proposed method. Please refer to the attached code for more information.

C Analysis of softEDA

We investigated the experimental results of the softEDA paper. Table 3 presents the experimental results reported in the appendix of the softEDA paper. The results suggest that, although softEDA can potentially enhance model performance, it is problematic to determine the optimal label smoothing factor for each model and dataset. Performance degradation compared to the baseline was also observed where the factor is improper for each setup. This finding motivated us to determine a better solution for finding optimal factors than a heuristic search. Furthermore, the authors performed the experiment on the full dataset. In contrast, we conducted the experiment through low-resource scenarios, which is more challenging for model.