Few-Shot Text Classification with Triplet Networks, Data Augmentation, and Curriculum Learning

Jason Wei[📚] Chengyu Huang[👥] Soroush Vosoughi[◎] Yu Cheng[▨] Shiqi Xu[📚]
[📚]ProtagoLabs [👥]International Monetary Fund
[◎]Dartmouth College [▨]Microsoft AI
{jason,xu}@protagolabs.com huangchengyu24@gmail.com
soroush@dartmouth.edu yu.cheng@microsoft.com

Abstract

Few-shot text classification is a fundamental NLP task in which a model aims to classify text into a large number of categories, given only a few training examples per category.

This paper explores data augmentation—a technique particularly suitable for training with limited data—for this few-shot, highly-multiclass text classification setting. On four diverse text classification tasks, we find that common data augmentation techniques can improve the performance of triplet networks by up to 3.0% on average.

To further boost performance, we present a simple training strategy called *curriculum data augmentation*, which leverages curriculum learning by first training on only original examples and then introducing augmented data as training progresses. We explore a *two-stage* and a *gradual* schedule, and find that, compared with standard single-stage training, curriculum data augmentation trains faster, improves performance, and remains robust to high amounts of noising from augmentation.

1 Introduction

Traditional text classification tasks such as sentiment classification (Socher et al., 2013) typically have few output classes (e.g., in binary classification), each with many training examples. Many practical scenarios such as relation classification (Han et al., 2018), answer selection (Kumar et al., 2019), and sentence clustering (Mnasri et al., 2017), however, have a converse setup characterized by a large number of output classes (Gupta et al., 2014), often with few training examples per class. This scenario, which we henceforth refer to as *few-shot, highly-multiclass text classification*, is a common setting in NLP applications and can be challenging due to the scarcity of training data.

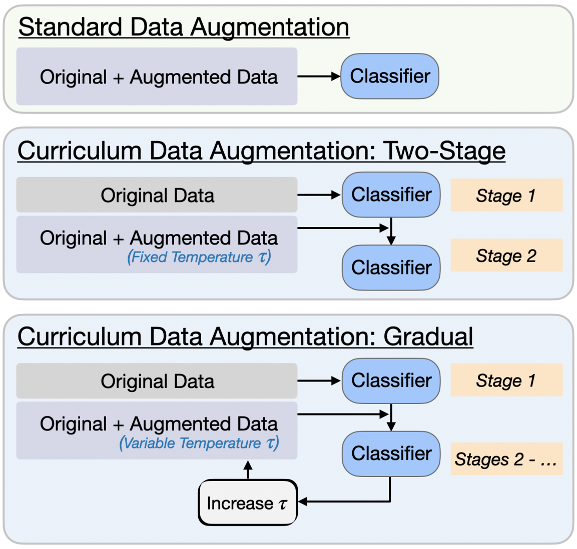
Standard Data Augmentation
Original + Augmented Data
Classifier
Curriculum Data Augmentation: Two-Stage
Original Data
Classifier
Stage 1
Original + Augmented Data
(Fixed Temperature τ)
Classifier
Stage 2
Curriculum Data Augmentation: Gradual
Original Data
Classifier
Stage 1
Original + Augmented Data
(Variable Temperature τ)
Classifier
Stages 2 - ...
Increase τ

Figure 1: Schematic showing the two types of curriculum augmentation that we propose. τ is a parameter that controls augmentation temperature (fraction of perturbed tokens).

Data augmentation for NLP has seen increased interest in recent years (Wei and Zou, 2019; Qiu et al., 2020). In traditional text classification tasks, it has been shown that although performance improvements can be marginal when training data is sufficient, augmentation is especially beneficial in limited data scenarios (Xie et al., 2020). As such, we hypothesize that the few-shot, highly-multiclass text classification scenario is a suitable context for data augmentation.

Based on this motivation, our paper makes two main contributions.

- First, we apply popular data augmentation techniques to the common triplet loss (Schroff et al., 2015) approach for few-shot, highly multiclass classification, finding that out-of-the-box augmentation can improve performance noticeably.
- We then propose a simple curriculum learning strategy called *curriculum data augmentation* and experiment with two schedules, as shown in Figure 1. A *two-stage* curriculum, which first trains on original data and then introduces aug-

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5493–5500
June 6–11, 2021. ©2021 Association for Computational Linguistics

mented data of fixed temperature (amount of noising), achieves slightly better performance than standard augmentation, while training faster and remaining more robust to high temperatures. A *gradual* curriculum, which also first trains on original data only but gradually increases augmentation temperature at each subsequent stage, takes longer to converge but improves more than 1% over standard augmentation.

2 Curriculum Data Augmentation

Motivation. Inspired by human and animal learning, curriculum learning (Bengio et al., 2009) posits that neural networks train better when examples are not randomly presented but instead organized in a meaningful order that gradually shows more concepts and complexity. Traditionally, curriculum learning approaches first assume that a range of example difficulty exists in the data and then leverage various heuristics to sort examples by difficulty and train models on progressively harder examples (Bengio et al., 2009; Tsvetkov et al., 2016; Weinshall et al., 2018). A newer school of thought, however, has noted that instead of discovering a curriculum in existing data, data can be intentionally modified to dictate an artificial range of difficulty (Korbar et al., 2018; Ganesh and Corso, 2020)—this is the approach we will take here.

Our approach. Unlike data augmentation in computer vision where augmented data undoubtedly resembles original data, in text, data augmentation techniques might introduce linguistic adversity and therefore can be seen as a form of noising (Li et al., 2017; Wang et al., 2018), where noised data is harder to learn from than unmodified original data. As such, we can create an artificial curriculum in the data by leveraging controlled application of data augmentation, starting by training on only original data and then adding augmented data with a higher levels of noising as training progresses. Specifically, we propose two simple schedules. **(1) Two-stage** curriculum data augmentation calls for one stage of training with only original data, followed by one stage of training with augmented data of fixed temperature. **(2) Gradual** curriculum data augmentation involves one stage of training with only original data, followed by multiple stages of training with augmented data where the temperature of augmented data (i.e., fraction of perturbed tokens) gradually increases each stage.

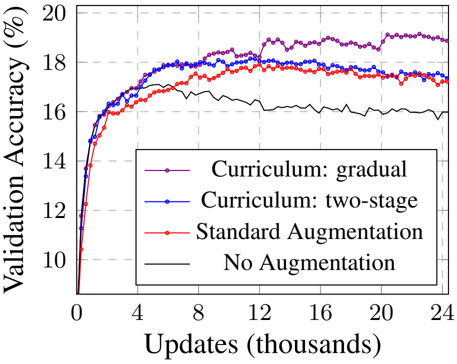

Figure 2: Example training plot for the HUFF dataset (41 classes, with 5 examples per class) using EDA augmentation (Wei and Zou, 2019). Our proposed two-stage curriculum (second stage starts at four-thousand updates) trains faster and achieves slightly higher performance compared with standard augmentation while using the same number of updates. Our proposed gradual curriculum (which here linearly increases augmentation temperature τ by 0.1 at $\{4, 8, 12, 16, 20\}$-thousand updates) outperforms both standard augmentation and the two-stage curriculum, but takes longer to converge. Results shown are averaged over thirteen random seeds.

3 Experimental Setup

We conduct empirical experiments to evaluate curriculum data augmentation on a variety of text classification tasks using a triplet loss model.[1]

3.1 Datasets

We will consider four diverse few-shot, highly-multiclass text classification scenarios:

1. **HUFF** ($c = 41$). The HuffPost Dataset categorizes 200k news headlines from 2012–2018 into 41 categories such as politics, wellness, entertainment, and travel (Misra, 2018). We use all 41 categories and perform a 70%-30% train-test split by class.
2. **FEWREL** ($c = 64$). The FewRel dataset contains sentences categorized by a relationship between its specified head and tail tokens such as ‘capital of,’ ‘member of,’ and ‘birth name’ (Han et al., 2018). We use all 64 classes given in the posted training set, splitting 100 examples per class into a test set, with the remainder of the examples going into the training set.
3. **COV-C** ($c = 87$). The COVID-Q dataset classifies questions into 89 clusters where all ques-

[1] Code is made publicly available at `https://github.com/jasonwei20/triplet-loss`.

tions in a cluster ask about the same thing (Wei et al., 2020). We use the train-test split with three training examples per class as given by the authors. We find that 2 of the 89 classes in the training set actually have only two examples per class instead of the reported three, and so we remove these classes from the training and test sets and use the 87 classes that remain.

4. **AMZN** ($c = 318$). The Amazon product review dataset aims to categorize a product into a certain class given a review (Yury, 2020). We only consider the 318 'level-3' classes given in this dataset with at least six examples per class.

To balance the class distribution during experiments, we randomly sample N_c examples per class to be used for training, with N_c varying based on the experiment and dataset. Our sampled training sets for COV-C and AMZN have $N_c = 3$ examples per class, and our training sets for HUFF and FEWREL have $N_c = 10$, a common low-resource scenario.[2] For all experiments, we use top-1 accuracy (%) as the evaluation metric.

3.2 Triplet Loss Model

For few-shot, highly-multiclass classification, a common approach is the *triplet loss* classifier (Schroff et al., 2015), first developed for facial recognition and now also used in NLP (dos Santos et al., 2016; Ein Dor et al., 2018; Lauriola and Moschitti, 2020). Specifically addressing few-shot classification, a triplet loss network minimizes distance between examples with the same label and maximizes distance between examples with different labels. During training, given a triplet of (anchor a, positive example p, and negative example n), a triplet loss network minimizes:

$$L = \sum_i d(a, p) - d(a, n) + \alpha \,, \quad (1)$$

where α is a margin enforced between positive and negative pairs, and $d(\cdot)$ computes the distance between the input encodings of two examples. To sample triplets, we will consider two strategies: *random sampling*, which selects triplets randomly, and *hard negative mining* (Schroff et al., 2015), where triplets are sampled such that $d(a, p) + \alpha > d(a, n)$. At evaluation time, a triplet loss classifier returns the class of the example in the training set with the smallest distance to a given test example. Indeed, both triplet loss and data augmentation target training with limited data, and so combining them seems particularly promising for the the few-shot classification scenario.

For our model, we use standard BERT-base with average-pooled encodings and then train a two-layer triplet loss network on top of these encodings. Our triplet loss network architecture contains a linear layer with 200 hidden units, tanh activation, a dropout layer with $p = 0.4$, and a final linear layer with 40 hidden units. We use cosine distance, a margin of $\alpha = 0.4$, a batch size of 64 triplets, and a learning rate of 2×10^{-5}.

3.3 Augmentation Techniques

We implement EDA (Wei and Zou, 2019), a popular combination of token-level augmentation techniques (synonym replacement, random insertion, random swap, random deletion) that defines their temperature parameter $0 \leq \tau \leq 1$ as the fraction of perturbed tokens, in §4.1–4.4, and explore four other techniques in §4.5.

3.4 Schedules

For the two-stage curriculum, we started by training on original data only, and when validation loss converges, we introduce augmented data of fixed temperature at an augmented to original data ratio of 4:1. For the gradual curriculum, we begin with a temperature of $\tau = 0.0$ (equivalent to no augmentation) and then linearly increase the temperature by 0.1 every time validation loss plateaus, up to a final temperature of 0.5. Schedules for each dataset are shown in the Appendix. Figure 2 shows an example training plot with our proposed curriculum schedules.

4 Results

4.1 Curriculum Data Augmentation

This section compares no augmentation, standard augmentation, and curriculum augmentation for triplet loss networks using two different triplet sampling strategies. Table 1 summarizes these results for five random seeds. We also implement a cross-entropy loss classifier for reference.

For triplet loss using random sampling, a model with no augmentation achieved a mean accuracy across our four datasets of 30.2%, and standard augmentation improved performance noticeably by +1.9%. Two-stage curriculum augmentation, which

[2] For COV-C, the given training set size is $N_c = 3$, and for AMZN, $N_c = 3$ is the largest possible such that the training set is balanced by class.

	HUFF $c=41$	FEWREL $c=64$	COV-C $c=87$	AMZN $c=318$	Average	Δ
Cross-entropy loss	13.3 ± 2.1	32.4 ± 2.3	26.1 ± 0.8	2.0 ± 0.3	18.5	-
+ standard data augmentation	16.3 ± 2.4	33.0 ± 1.1	24.0 ± 1.6	2.2 ± 0.4	18.9	+0.4
Triplet loss with random sampling	20.9 ± 1.0	43.6 ± 1.2	39.7 ± 1.0	16.4 ± 1.3	30.2	-
+ standard data augmentation	22.2 ± 1.4	44.2 ± 1.6	45.4 ± 1.8	16.5 ± 1.7	32.1	+1.9
+ **curriculum data augmentation: two-stage**	22.3 ± 1.6	44.2 ± 1.8	46.5 ± 1.7	17.2 ± 1.3	32.6	+2.4
+ **curriculum data augmentation: gradual**	23.7 ± 1.2	46.1 ± 0.9	47.1 ± 1.3	17.6 ± 1.0	33.6	+3.4
Triplet loss with hard negative mining	21.0 ± 1.2	44.6 ± 1.2	39.5 ± 1.0	16.2 ± 0.9	30.3	-
+ standard data augmentation	22.6 ± 1.8	45.0 ± 1.6	48.2 ± 0.9	17.4 ± 1.7	33.3	+3.0
+ **curriculum data augmentation: two-stage**	22.6 ± 1.8	45.7 ± 1.4	47.6 ± 1.3	17.9 ± 1.1	33.5	+3.2
+ **curriculum data augmentation: gradual**	23.8 ± 0.9	47.1 ± 1.4	48.9 ± 0.9	18.9 ± 0.9	34.7	+4.4

Table 1: Accuracy (%) on four diverse highly multiclass classification tasks for no augmentation, standard augmentation, and curriculum augmentation. c: number of classes; Δ: improvement compared with no augmentation.

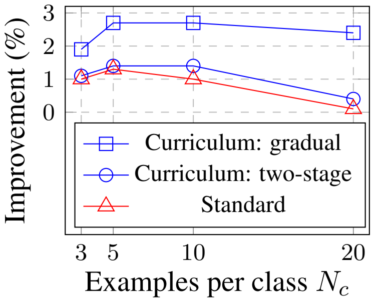

Figure 3: Improvement from data augmentation for different dataset sizes (results averaged over HUFF and FEWREL datasets).

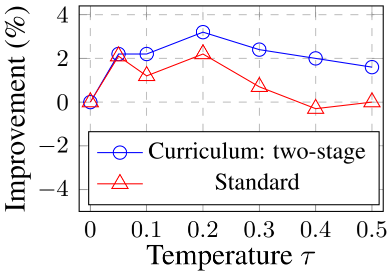

Figure 4: Curriculum data augmentation outperforms standard data augmentation for a range of different augmentation temperatures τ. Whereas standard augmentation performs better at lower τ, curriculum data augmentation helps even for higher τ (e.g., $\tau \geq 0.2$).

trains for the same number of updates as standard augmentation, achieved a mean accuracy of 32.4%, outperforming standard augmentation by +0.5%. The gradual curriculum further improved +1.0% over the two-stage curriculum.

For triplet loss with hard negative mining, standard augmentation substantially improved +3.0% over no augmentation, as adding in augmented data, which is more difficult to classify, likely helped generate a more diverse set of hard negatives. The two-stage curriculum still maintained small improvement over standard augmentation here, and the gradual curriculum provided an even-stronger boost of +4.4% over no augmentation, possibly because increasing the temperature of augmented data over time facilitated hard-negative mining more so than using a constant temperature.

Notably, the largest gains for all augmentation types were on COV-C (up to +9.4%). We hypothesize that this occurred not necessarily because of COV-C's smaller data size; rather, there was likely more overfitting to be mitigated by data augmentation as a result of the greater semantic difference between COV-C and the corpus used to pre-train BERT, compared with the other three datasets.

4.2 Ablation: Dataset Size

This ablation investigates how data augmentation performs for different dataset sizes. Figure 3 shows these results for hard negative mining averaged over HUFF and FEWREL, our two datasets where sufficient data is available. The two-stage curriculum outperformed standard augmentation by a small margin, although both dropped in performance at $N_c = 20$, consistent with prior findings on the diminished effect of data augmentation for larger datasets (Xie et al., 2020; Andreas, 2020). The gradual curriculum, on the other hand, maintained relatively robust improvement for all dataset sizes explored.

4.3 Ablation: Augmentation Temperature

Effective curriculum learning necessitates a range of difficulty in training data. In our case, this range is controlled by *augmentation temperature*, a parameter that dictates how perturbed augmented examples are and therefore affects the distribution of difficulty in training examples. When the distribution of difficulty in data is larger, we should expect

Schedule	HUFF	FEWREL	COV-C	AMZN	Avg.
Curriculum	23.7	46.1	48.1	17.6	**33.63**
Control	23.5	45.3	46.3	17.1	33.05
Anti	23.3	44.8	46.2	17.5	32.95

Table 2: Gradual curriculum augmentation with three schedules. Curriculum: temperature τ increases. Control: τ is randomly selected every fifty updates. Anti: decreasing τ. Results are shown for ten seeds.

a greater improvement from curriculum learning.

Figure 4 compares standard and two-stage curriculum augmentation for various temperatures, with results averaged over all four datasets. At low temperature, augmented examples remained pretty similar to original examples, and so the range of difficulty in examples was small and therefore curriculum learning showed little improvement. At higher temperatures, however, augmented examples became quite different from original examples, and so the range of difficulty in examples was much larger and therefore curriculum data augmentation improved over standard augmentation more. Whereas Wei and Zou (2019) recommend $\tau \in \{0.05, 0.1\}$, our curriculum framework liberates us to use much larger τ and maintain relatively robust improvements even at $\tau \in \{0.4, 0.5\}$ when standard augmentation is no longer useful.

4.4 Ablation: Curriculum Schedules

The gradual curriculum linearly increases temperature τ from 0.0 to 0.5 in six stages, and so to isolate the effect of this curriculum, in this section we compare it with a control schedule (where the τ in each stage is decided randomly) and an anti-curriculum schedule (where τ linearly decreases from 0.5 to 0.0 in six stages). As expected, these results, shown in Table 2, indicate that the curriculum contributes substantively over the control schedule.

4.5 For Various Augmentation Techniques

As our experiments so far have focused on EDA augmentation (Wei and Zou, 2019), this section explores other common techniques in the curriculum framework: **(1) Token Substitution** replaces words with WordNet synonyms (Zhang et al., 2015); **(2) Pervasive Dropout** applies word-level dropout with probability p=0.1 (Sennrich et al., 2016a); **(3) SwitchOut** replaces a token with a randomly token uniformly sampled from the vocabulary (Wang et al., 2018); and **(4) Round-Trip Translation** translates text into another language and then back into the original language (Sennrich et al., 2016b). Figure 5 compares standard and two-stage curriculum results averaged over all datasets. EDA improved performance the most, perhaps because it combines four token perturbation functions, creating more diverse noise compared with using a single operation.

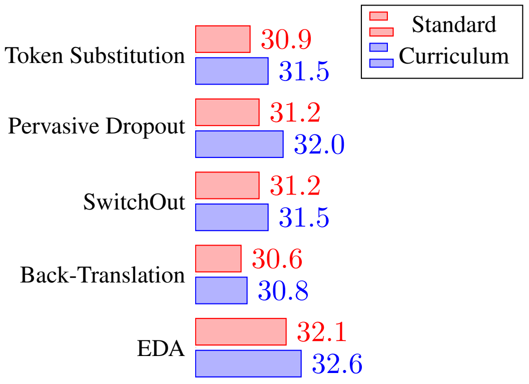

Figure 5: Common text data augmentation techniques work better in the curriculum framework (two-stage) than standard single-stage training. A model with no data augmentation achieved a performance of 30.2%.

5 Related Work and Conclusions

Our work combines curriculum learning, data augmentation, and triplet loss, and is inspired by prior work in these areas. In vision, several papers have proposed reinforcement learning policies for data augmentation (Cubuk et al., 2019; Ho et al., 2019), and hard negative mining (Schroff et al., 2015; Song et al., 2016) itself can be seen as a form of curriculum learning. In NLP, the work of Kumar et al. (2019) is perhaps most similar to ours—they show that sampling strategies are key for improving performance with triplet loss networks. We see our work as the first to explicitly analyze curriculum learning for data augmentation in text.

In closing, we have proposed a curriculum data augmentation framework that is simple yet provides empirical performance improvements, a compelling case for the combination of ideas explored. Our approach exemplifies how data augmentation can create an artificial range of example difficulty that is helpful for curriculum learning, a direction that potentially warrants future research.

Acknowledgements

We thank Ruibo Liu, Weicheng Ma, Jerry Wei, and Chunxiao Zhou for feedback on the manuscript. We also thank Kai Zou for organizational support.

References

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48. Association for Computing Machinery.

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 (CVPR)*.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. 2018. Learning thematic similarity metric from article sections using triplet networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Melbourne, Australia. Association for Computational Linguistics.

Madan Ravi Ganesh and Jason J. Corso. 2020. Rethinking curriculum learning with incremental labels and adaptive compensation.

Maya R. Gupta, Samy Bengio, and Jason Weston. 2014. Training highly multiclass classifiers. *J. Mach. Learn. Res.*, 15(1):1461–1492.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. 2019. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR.

Bruno Korbar, Du Tran, and Lorenzo Torresani. 2018. Cooperative learning of audio and video models from self-supervised synchronization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 7774–7785.

Sawan Kumar, Shweta Garg, Kartik Mehta, and Nikhil Rasiwasia. 2019. Improving answer selection and answer triggering using hard negatives. 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 5911–5917, Hong Kong, China. Association for Computational Linguistics.

Ivano Lauriola and Alessandro Moschitti. 2020. Context-based transformer models for answer sentence selection. *arXiv preprint arXiv:2006.01285*.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 21–27, Valencia, Spain. Association for Computational Linguistics.

Rishabh Misra. 2018. HuffPost news category dataset.

Maâli Mnasri, Gaël de Chalendar, and Olivier Ferret. 2017. Taking into account inter-sentence similarity for update summarization. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–209, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Siyuan Qiu, Binxia Xu, Jie Zhang, Yafang Wang, Xiaoyu Shen, Gerard de Melo, Chong Long, and Xiaolong Li. 2020. Easyaug: An automatic textual data augmentation platform for classification tasks. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 249–252, New York, NY, USA. Association for Computing Machinery.

F. Schroff, D. Kalenichenko, and J. Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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, Berlin, Germany. 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.

H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with Bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, 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.

Jerry Wei, Chengyu Huang, Soroush Vosoughi, and Jason Wei. 2020. What Are People Asking About COVID-19? A Question Classification Dataset. *arXiv preprint arXiv:2005.12522*.

Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5238–5246. PMLR.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training.

Kashnitsky Yury. 2020. Hierarchical text classification of Amazon product reviews.

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.

	HUFF $c = 41$	FEWREL $c = 64$	COV-C $c = 87$	AMZN $c = 318$
SINGLE-STAGE TRAINING				
Updates until convergence, no aug. (approx)	4,000	8,000	4,000	15,000
Update until convergence, aug. (approx)	10,000	10,000	8,000	20,000
Total updates	15,000	15,000	15,000	25,000
CURRICULUM: TWO-STAGE				
Stage 1 updates	4,000	6,000	4,000	8,000
Stage 2 updates	11,000	9,000	11,000	17,000
Total updates	15,000	15,000	15,000	25,000
CURRICULUM: GRADUAL				
Stage 1 updates	6,000	6,000	6,000	10,000
Updates per stage in stages 2-6	6,000	6,000	4,000	8,000
Total updates	36,000	36,000	26,000	50,000

Table 3: Training schedules for single-stage training, two-stage curriculum training, and gradual curriculum training.

A Appendix

Table 3 shows the training schedules for single-stage, two-stage curriculum, and gradual curriculum training.

All models in standard single-stage training (with and without augmentation) for the same dataset trained for the same number of updates; convergence typically took longer with augmentation compared to without augmentation.

Curriculum two-stage training employs a first stage of only original data and a second stage of augmented data, using the same number of updates as single-stage training in total. We determined the number of updates in the first stage based on when training loss plateaued in the training plot for training with no augmentation.

The gradual curriculum starts with one stage of training with original data only and then increases the augmentation temperature by 0.1 in each of the following five stages. To determine the number of updates in each stage, we examined training plots in preliminary experiments and increased the augmentation temperature (i.e., begun the next stage) whenever training loss plateaued. Since our preliminary experiments already showed relatively strong performance improvements, we did not perform an extensive hyperparameter search or experiment with automatic scheduling, which could further improve performance. As the gradual curriculum trains on more diverse set of augmented data, more updates are needed than in the single-stage and two-stage schedules.

For evaluation, we evaluate our models every 200 updates for COV-C and every 300 updates for HUFF, FEWREL, and AMZN, reporting the highest validation accuracy achieved during training.

In all models, we include 20% original data whenever augmented data is used, in order to prevent catastrophic forgetting.