HuCurL: Human-Induced Curriculum Learning

Anonymous ACL submission

Abstract

Curriculum Learning (CL) techniques can accelerate training and improve model generalizability by ordering training samples based on their difficulty. This work advocates a re-imagining of CL paradigms so that they can determine if and how samples should be used for training, given *prior knowledge* about their difficulty. We study how knowledge about a sample's difficulty to humans can be used to create a training curriculum, and improve performance on downstream tasks. We quantify difficulty towards using the inter-annotator agreement of data with multiple annotations per sample. The proposed CL framework can control the initial weight of samples and their rate of growth or decay, with respect to the prior knowledge about sample difficulty, allowing to effectively explore the curriculum space and derive curricula that are not predefined. We evaluate our approach on natural language inference and two other text classification tasks. Our model provides a consistent gain in performance across all datasets and configurations.

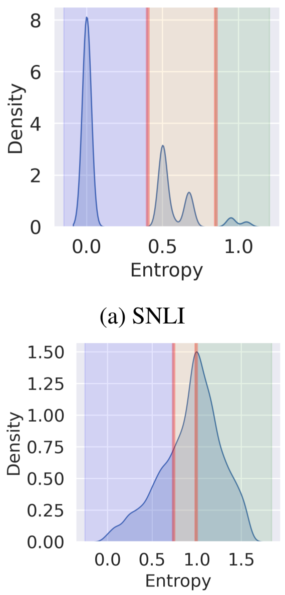
1 Introduction

Curriculum learning (Bengio et al., 2009; Kumar et al., 2010) is a training paradigm inspired by the learning process of humans and animals, called *shaping* (Peterson, 2004), where data samples are scheduled or ordered for training given models, e.g., by starting with easier samples and gradually transitioning to more difficult ones. CL is instrumental in many settings of computational modeling, from limited and noisy data to limited training time to problems with hierarchical complexity (Wu et al., 2021).

A primary challenge in creating effective curricula for machines is to accurately estimate the difficulty of data samples. Current research has addressed this challenge by estimating sample difficulty through model behavior during training, employing signals such as prediction loss (Xu et al., 2020; Wu et al., 2021), consistency in prediction loss (Amiri et al., 2017; Xu et al., 2020), moving average of loss (Zhou et al., 2020), transformations of loss (Jiang et al., 2018; Castells et al., 2020) and weighting of loss (Kumar et al., 2010; Jiang et al., 2014, 2015).

The objective of the above models and the basic principle of CL is to generate an ordering of samples in terms of their relative difficulty, where training often starts with easier samples and gradually proceeds with more difficult ones (Bengio et al., 2009; Kumar et al., 2010). However, training loss can be affected by model initialization, stochastic gradient descent dynamics and random batch sampling (Erhan et al., 2010), which leads to noisy estimations of sample difficulty and unnecessarily limits the curriculum space that can be examined for finding effective curricula. Although these challenges could be partially alleviated through moving average of loss (Zhou et al., 2020) or loss transformation (Castells et al., 2020), we suggest investigating whether *prior knowledge* about latent sample difficulty can be used to create effective and robust CL paradigms for training neural networks.

In this paper, we consider entropy among annotation labels for each data sample as prior knowledge about sample difficulty. Such use of entropy is supported by Nie et al. (2020b), who reported a consistent positive correlation between model accuracy and level of human agreement. We quantify annotation disagreement using Shannon entropy (Shannon, 2001), and split training samples into partitions based on their entropy. For example, three partitions can be considered as *easy*, *medium*, and *hard* classes. Armed with this prior knowledge about sample difficulty, we devise a novel CL framework, named Human-Induced Curriculum Learning (HuCurL), that effectively explores the curriculum space with respect to prior knowledge during training. HuCurL specifies a weight function that controls when and to which extent a sample should contribute to training, given prior knowledge about its difficulty. These capabilities enable our framework to explore the curriculum space and discover effective data-driven curricula, given prior knowledge about sample difficulty. We discuss how the parameters of our model can be optimized to discover the optimal curriculum.



(a) SNLI

(b) ChaosNLI

Figure 1: Distributions of entropy/difficulty classes for the three datasets used in our experiments (see Section 3.1 for details). Samples of the *easy* class are to the left of the first vertical line, those of the *medium* class are between the two vertical lines, and samples of the *hard* class are to the right of the second line.

Annotation entropy is a generic metric and can be computed for any dataset that carries multiple annotations per sample. Such information is present in most NLP datasets at their creation time. In addition, given the recent works by the NLP community that aims to increase human involvement in dataset development and model evaluation (Rodriguez et al., 2021; Nie et al., 2020a; Parrish et al., 2021), our work is a timely effort in utilizing annotation entropy to inform the technical objective of creating effective curricula for NLP tasks.

We evaluate our approach on natural language inference and two other text classification tasks. HuCurL results in a consistent gain in performance across all datasets and shows higher efficacy in cases of limited and noisy data, shorter training time, and harder test samples.

2 Human-Induced Curriculum

2.1 Inter-annotator Agreement

Consider a dataset $\mathcal{D}$, where each sample consists of a feature vector and a set of categorical annotations $(\boldsymbol{x}_i, \{y_i^{(1)}, .., y_i^{(l_i)}\})$. We define difficulty scores using annotation entropy (Shannon, 2001):

$$H(\boldsymbol{x}_i) = -\sum_{c=1}^{k} p_c * \log p_c, \qquad (1)$$

where k is the number of label categories and p_c is the fraction of annotators who choose label c for the sample. We use the entropy scores to split the training data into $\{$*easy*, *medium*, *hard*$\}$ sets, corresponding to three levels of difficulty. The splits can be obtained by dividing the data based on entropy distribution percentiles, or an empirical cumulative distribution function (Platanios et al., 2019). We also explore the division into more than three partitions.

Figure 1 shows the entropy distribution of four datasets used in this paper and the thresholds for splitting the data into the three difficulty classes. SNLI has a greater density of easier samples and a smaller density of harder ones.

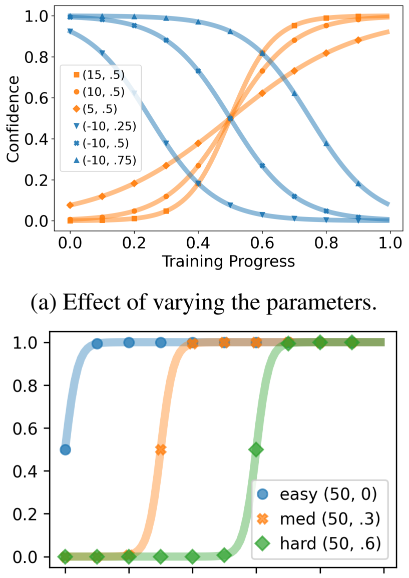
2.2 Curriculum as Weighting Framework

We define a curriculum based on entropy classes using the generalized logistic function (Richards, 1959), which facilitates exploring the curriculum space by dynamically weighting samples:

$$w(t; r, s) = \frac{1}{1 + \exp(-r \times (t - s))}, \qquad (2)$$

where $t \in [0, 1]$ is the training progress (typically, iteration number divided by max iterations), $r \in \mathbb{R}$ is the rate-of-change and specifies how fast the weight of a sample can increase ($r > 0$) or decrease ($r < 0$), and $s \in \mathbb{R}$ is the shift parameter that moves the pivot weight of the logistics function ($f(.) = .5$) to the left or right such that at $t = s$ the weight is 0.5. Figure 2a illustrates the effect of varying these parameters. Greater absolute values for the rate parameter enforce a greater rate of change in sample weights, while greater values of the shift parameter enforce longer delays in reaching the pivot weight of 0.5. These parameters provide flexibility in controlling the importance (weight) and order of samples during training, which are both crucial for deriving effective curricula. We define a weight function for each entropy class using separate hyperparameters (r, s).

The proposed sample weighting framework provides flexibility in ordering samples according to difficulty. It can be learned to approximate existing predefined curricula, e.g. it is possible to begin training with only easy instances or only difficult instances (as an *anti-curriculum*) or a combination of both. Figure 2b shows a specific configuration for the logistic function based on standard CL (Bengio et al., 2009; Kumar et al., 2010), where training starts with easier samples and gradually proceeds with harder ones.



(a) Effect of varying the parameters.

(b) Easy to Hard Curriculum.

Figure 2: Generalized logistic function for learning curricula. The rate of growth (r) and shift (s) parameters, see (2), are reported in brackets. (a) shows the effect of changing the two parameters. (b) is a parameter configuration for a curriculum that start training with easier samples and gradually introduce medium and hard samples at 30% and 60% of training.

2.3 The HuCurL Model

The Human-Induced Curriculum Learning (HuCurL) model consists of the following components: entropy computation, data partitioning, and the CL framework described above. The model architecture is shown in Figure 3. Each instance is weighted according to the weight function of its corresponding entropy class and the current training progress. In HuCurL, all instances of the same entropy class (*easy*, *medium*, *hard*) carry the same weight, but they can contribute differently to the objective function depending on their respective loss. Weighted loss for each sample i is calculated as $\hat{l}_i = w(t; r_c, s_c) * l_i$ where l_i is the instantaneous loss of i and w is the weight from (2).

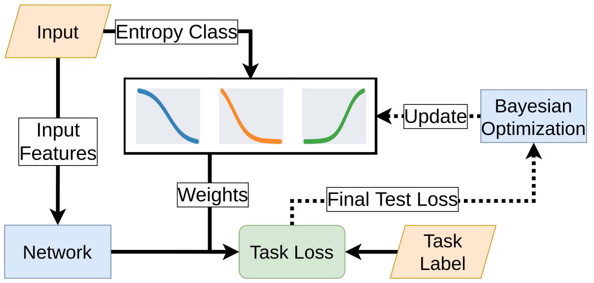


Figure 3: The architecture of our HuCurL model. HuCurL defines a difficulty score based on the entropy among annotation labels for each sample and assigns samples to different entropy classes: *easy*, *medium*, and *hard* before training. A curriculum/weight function is defined for each entropy class. During training, the model weights each training instance according to the weight function of its corresponding entropy class. Each weight curve is defined by a pair of parameters (r, s) that can be adjusted using hyperparameter optimization methods to discover the optimal curriculum based on sample difficulty and the dataset.

We consider three difficulty levels as it will be feasible to tune the learning strategy for each class and find the optimal curriculum within the HuCurL framework. In addition, it is sufficient to schedule training in terms of stages of difficulty levels (Bengio et al., 2009; Guo et al., 2018). Nonetheless, our approach can be extended to a larger number of difficulty classes. We note that current CL approaches split their data into *easy* and *hard* difficulty groups only. However, three classes including *medium* provide finer-grained difficulty information and can filter out too easy or hard samples.

3 Experiments

3.1 Datasets

We evaluate the methods on three datasets that contain multiple annotations per sample.

SNLI: the task of natural language inference to determine the semantic relationship between a given pair of sentences (a premise and a hypothesis), as {*entailment*, *neutral*, *contradiction* }. The task requires linguistic reasoning for successful prediction. The Stanford natural language inference (SNLI) dataset contains 550k training samples, 10k development samples, and 10k test samples. We consider samples validated by at least 4 annotators, a total of 39k samples.[1] In the scope of this work,

[1] We ignore samples labeled by 2-3 annotators, <100 cases.

the full SNLI dataset consists of 39k samples. The data is imbalanced in terms of entropy classes, see Figure 1. In order to remove the biases imposed by imbalanced data, we create another training set with balanced entropy classes. The training set is downsampled such that each entropy class has an equal number of samples as the least frequent class. *snli balanced* contains a total of 2.3k samples, equally divided across entropy classes.

ChaosNLI: ChaosNLI (Nie et al., 2020b) collects 100 annotations per sample for about 1500 samples of the development set SNLI and 1500 of the development set of MNLI (Williams et al., 2017). We devise a new task using ChaosNLI as a training dataset, consisting of the 3000 samples, using the remaining 8500 sample of the development set of SNLI as a development set, and the test set of SNLI as a test set. Using this dataset with 100 annotations per sample, we can obtain a much more accuracte estimate of the difficulty of training samples. Therefore, we expect HuCurL to be more effective.

3.2 Baselines

We use the following baselines:

No-CL: standard training without re-weighting.

Difficulty Prediction (DP): (Yang et al., 2019) defines a difficulty score based on multi-annotator labels as follows:

$$d_i = \frac{\sum_{j=1}^{l_i} f(y_i^{(j)}, \hat{y}_i)}{l_i}, \qquad (3)$$

where $\hat{y}_i$ is the ground truth label and f is a scoring function that measures the quality of an annotation. They define f as the Spearman's rank correlation coefficient and compute the correlation between labels produced by experts and non-experts. They propose a pruning and a re-weighting method to use this information for model improvement. The re-weighting method uses the following formula:

$$1 - \alpha \frac{d_i - \tau}{1 - \tau}, \qquad (4)$$

where τ is a predefined threshold. This is the best performing method in (Yang et al., 2019), which we refer to as Difficulty Prediction (DP).

Mentornet: (Jiang et al., 2018) uses an auxiliary network to weight samples at every iteration. The auxiliary network is trained at the same time as the main model. It takes as an input the training loss of the last k steps, the running mean of the loss, the current epoch number (to account for training progress for learning curricula), and the target labels. It consists of an LSTM layer to encode the k steps of loss, embedding matrices for the target label and epoch number, a fully connected layer, and a final sigmoid layer. The sigmoid layer outputs weights of samples for training.

Self-paced Learning (SPL): (Kumar et al., 2010) weights instances based on their difficulty to the model by optimizing the following objective:

$$\mathcal{L}(\mathcal{D}; \theta) = \arg\min_{\boldsymbol{v}} \sum_{i}^{n} v_i l_i + f(\boldsymbol{v}; \lambda), \qquad (5)$$

where l_i is the loss of instance i parameterized by θ, v_i is a trainable weight parameter assigned to each instance, and a regularization function of the weights is added to the learning objective. The model chooses $\mathbf{v}$, under the constraint of $f(\mathbf{v}; \lambda)$, that minimizes its loss. The binary scheme SPL is defined by the regularization function $f(\mathbf{v}; \lambda) = -\lambda \|\mathbf{v}\|_1$. In the binary scheme, if $l_i < \lambda$, $v_i = 1$, otherwise $v_i = 0$, indicating that only samples that are easy to the model are selected at each step. λ can be increased over time to allow difficult instances to be seen during training.

SuperLoss (SL): (Castells et al., 2020) is the state-of-the-art CL approach that uses the following objective function with a closed form solution to infer sample weights:

$$\mathcal{L}_\lambda = (l_i - \tau)\, \sigma_i + \lambda\, (\log \sigma_i)^2, \qquad (6)$$

which has the solution

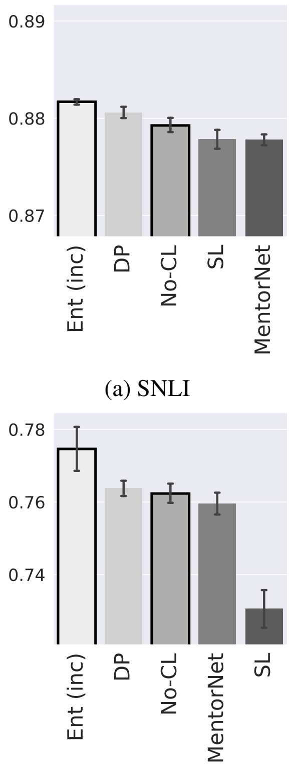
$$\sigma_\lambda^* = \exp(-W(\frac{1}{2}\max(-\frac{2}{e}, \beta))), \ \beta = \frac{l_i - \tau}{\lambda}, \qquad (7)$$

where σ is the sample confidence and W is the Lambert W function (Euler, 1783). At each training iteration, the weight of an instance is computed as a transformation of its loss and is used to re-weight the loss. Instances with a small loss are emphasized, and instances with a larger loss are de-emphasized.

We compare the above baselines against **Ent (inc)**: HuCurL with gradually increasing difficulty as illustrated in Figure 2b.

3.3 Setting

We optimize the parameters α of DP in (4) and λ of SL in (6) using the optimization framework



(a) SNLI

(b) ChaosNLI

Figure 4: Accuracy of different CL approaches on the four *full* datasets.

Optuna (Akiba et al., 2019). The search for λ converged in 31 trials, each trial is the average of five random seeds, and the optimal value found is $\lambda = 1.2$ across the datasets. Following (Castells et al., 2020), we set τ in (7) to the moving average of the loss in all experiments. The search for α converged in 21 trials, each trial is averaged over five seeds and the optimal value found is 0.9. A value of 0.9 for α in (4) means that difficult samples are down weighted more strongly throughout training. In the original paper, authors set $\alpha = 0.5$. We dynamically set τ in (4) upon loading the dataset as the 50th percentile of difficulty scores of the training set. This setting of τ improves performance compared to $\tau = 0.8$ reported in the original paper. In (4), we define $\hat{y}_i$ to be the majority-vote label by annotators, and define the scoring function $f = \mathbb{1}[y_i^{(j)} == \hat{y}_i]$. So, the difficulty in (4) is simply the percentage of annotators who agree on the majority-vote label. We follow the implementations of MentorNet and SPL from (Jiang et al., 2018) and use the same hyperparameters. Finally, we treat the number of partitions of difficulty classes in HuCurL as a hyper-paramter, and report the result using the optimal number of partitions. The optimal number of difficulty classes are the following. SNLI: 3 classes, ChaosNLI: 12 classes.

We use `roberta-base` for SNLI and ChaosNLI from (Wolf et al., 2020; Devlin et al., 2019; Liu et al., 2019). We set learning rate to $1e-5$, batch size to 16, number of training epochs to 10 (we confirm that this is sufficient for all models to convergence), and use Adam (Kingma and Ba, 2017) as model optimizer. For each experiment, we train the models with twenty random seeds (except SNLI, which is trained with five seeds) applied to both `pytorch` and `numpy`. Additionally, during all data pre-processing, splitting, and sub-sampling, random seed is set to 0. A single NVIDIA A100 40GB GPU is used for training. The development set is used to determine the best training step which is used for the final evaluation.

3.4 Results and Insights

3.5 HuCurL Improves Performance

Figure 4 reports the main results of our experiments. Ent (inc) is the HuCurL model with a curriculum of gradually increasing difficulty, and it achieve the top performance in all cases. Ent (inc) proves to be a generalizable curriculum, which uses the entropy signal and the weighting scheme (2) and performs better than the current state-of-the-art CL methods.

HuCurL significantly improves the performance on the *medium* samples, which we believe are the most important data points in training and key to performance. This is because *easy* samples may be predictable through learning high-level linguistic features and semantics or through artifacts unrelated to the task (Liu et al., 2020; Poliak et al., 2018; Gururangan et al., 2018; Sinha et al., 2021), and *hard* samples may be too difficult for a model or noisy.

4 Related Work

Use of Human Annotations: Annotation information has been extensively used by previous research in natural language processing (NLP) to devise better strategies for (a): further data collection, e.g., through estimating sample difficulty and collecting a larger number of annotations for more difficult samples or particularly collecting expert labels for those samples (Yang et al., 2019; Dligach et al., 2010), (b): for model improvement, e.g., by learning rationales of human annotators for better model training (Zaidan and Eisner, 2008) or pruning and weighting samples based on their corresponding inter-annotator agreement (Yang et al., 2019), or (c): for efficient use of monetary funds, in terms of collecting the minimal number of annotations per sample for the highest model accuracy gain (Dligach et al., 2010).

Curriculum Learning: The idea of introducing samples in an order of increasing difficulty during training guides the optimization process through a better path in the parameter space and can lead to better solutions in terms of final test error and speed up the optimization process (Bengio et al., 2009; Kumar et al., 2010). Research in CL have reported performance gain through ordering samples for training (Wang et al., 2019; Platanios et al., 2019; Pentina et al., 2015). Other works have shown that *anti-curricula* are useful for certain tasks (Kocmi and Bojar, 2017; Zhang et al., 2018, 2019). CurriculumNet (Guo et al., 2018) is a curriculum learning approach that works by splitting the training data into three groups of increasing complexity. The model extracts deep features from the training data, cluster the data into three groups, and rank the groups based on intra-group coherence. It considers less coherent groups to be noisy and defines three classes of *clean*, *noisy*, and *highly noisy* samples. CurriculumNet achieves better performance through pruning the noisy data and focuses the training on clean data. Wu et al. (2021) have performed a large-scale study over multiple datasets, architectures, and difficulty scores and found that curricula are generally beneficial when there are limited training resources or noisy data.

5 Conclusion and Future Work

The Human-induced Curriculum Learning (HuCurL) approach is an effective approach that can encode prior knowledge in learning data-driven curricula. A subject of future work is to study the ability of HuCurL to systematically explore the curricula space, learn existing predefined curricula, and discover effective data-driven curricula with respect to given prior knowledge about sample difficulty by optimizing the parameters of the weighting function (Eq 2). As prior knowledge, it utilizes entropy in human annotations per sample, which is available in most NLP datasets at creation time, while typically overlooked. Different versions of HuCurl outperform the state-of-the-art CL methods. HuCurL adds no computational cost. Data annotations are readily available, entropy is computed as a pre-processing step with little cost, and during training, the weighting of samples incurs no additional cost. In the future, we aim to investigate other types of prior knowledge about sample difficulty, focusing on specific linguistic metrics that might be useful for learning effective curricula. In addition, we plan to develop CL approaches that are effective for other natural language tasks and text properties, e.g., short vs. long text.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Hadi Amiri, Timothy Miller, and Guergana Savova. 2017. Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2401–2410, Copenhagen, Denmark. 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)*, pages 41–48.

Thibault Castells, Philippe Weinzaepfel, and Jerome Revaud. 2020. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Dmitriy Dligach, Rodney Nielsen, and Martha Palmer. 2010. To annotate more accurately or to annotate more. In *Proceedings of the Fourth Linguistic Annotation Workshop (LAW)*, pages 64–72.

Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. 2010. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208. JMLR Workshop and Conference Proceedings.

Leonhard Euler. 1783. De serie lambertine plurimisque eius insignibus proprietatibus. *Acta Academiae scientiarum imperialis petropolitanae*, pages 29–51.

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 107–112. Association for Computational Linguistics (ACL).

Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. 2014. Self-paced learning with diversity. *Advances in Neural Information Processing Systems*, 27:2078–2086.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning (ICML)*, pages 2304–2313. PMLR.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386.

M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in Neural Information Processing Systems (NIPS)*, 23:1189–1197.

Tianyu Liu, Zheng Xin, Baobao Chang, and Zhifang Sui. 2020. HypoNLI: Exploring the artificial patterns of hypothesis-only bias in natural language inference. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, pages 6852–6860, Marseille, France. European Language Resources Association.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020a. Adversarial NLI: A new benchmark for natural language understanding. In *ACL*, pages 4885–4901, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020b. What can we learn from collective human opinions on natural language inference data? In *EMNLP*, pages 9131–9143.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *EMNLP*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. 2015. Curriculum learning of multiple tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5492–5500.

Gail B Peterson. 2004. A day of great illumination: Bf skinner's discovery of shaping. *Journal of the experimental analysis of behavior*, 82(3):317–328.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 1162–1172.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, page 180.

FJ Richards. 1959. A flexible growth function for empirical use. *Journal of experimental Botany (JXB)*, 10(2):290–301.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *ACL-IJCNLP*, pages 4486–4503, Online. Association for Computational Linguistics.

Claude Elwood Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online. Association for Computational Linguistics.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. 2019. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5017–5026.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2021. When do curricula work? In *International Conference on Learning Representations (ICLR)*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6095–6104.

Yinfei Yang, Oshin Agarwal, Chris Tar, Byron C Wallace, and Ani Nenkova. 2019. Predicting annotation difficulty to improve task routing and model performance for biomedical information extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 1471–1480.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing (EMNLP)*, pages 31–40.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1903–1915.

Tianyi Zhou, Shengjie Wang, and Jeff A Bilmes. 2020. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.