Towards a Task-Agnostic Model of Difficulty Estimation for Supervised Learning Tasks

Antonio Laverghetta Jr., Jamshidbek Mirzakhalov, John Licato
Advanced Machine and Human Reasoning (AMHR) Lab
Dept. of Computer Science and Engineering
University of South Florida
Tampa, FL, USA
{alaverghett,mirzakhalov,licato}@usf.edu

Abstract

Curriculum learning, a training strategy where training data are ordered based on their difficulty, has been shown to improve performance and reduce training time on various NLP tasks. While much work over the years has developed novel approaches for generating curricula, these strategies are typically only suited for the task they were designed for. This work explores developing a task-agnostic model for problem difficulty and applying it to the Stanford Natural Language Inference (SNLI) dataset. Using the human responses that come with the dev set of SNLI, we train both regression and classification models to predict how many annotators will answer a question correctly and then project the difficulty estimates onto the full SNLI train set to create the curriculum. We argue that our curriculum is effectively capturing difficulty for this task through various analyses of both the model and the predicted difficulty scores.

1 Introduction

Recent advances on natural language processing (NLP) benchmarks have been driven by increasingly sophisticated language models, which are pretrained on enormous amounts of data before use. Refinements of this process has led to increasingly powerful language models, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and more recently T5 (Raffel et al., 2019). Such models are quickly saturating even new tasks that have undergone a rigorous adversarial filtering process (Zellers et al., 2018). However, these downstream performance improvements also require more computational resources and data to train the models, which is not always feasible. Curriculum learning (Elman, 1993), a strategy where the model is trained on easier examples before harder ones, has recently been shown to improve performance and reduce training time on a variety of NLP tasks, especially machine translation (Liu et al., 2020; Wang et al., 2020; Zhou et al., 2020; Xu et al., 2020).

The success of this research shows that the order in which training data is presented to a model is important, but how to best apply curriculum learning broadly to NLP and how to design effective measures of difficulty to create curricula remain relatively unexplored. Prior approaches either use a hand-crafted measure of difficulty that works well for a particular task or design an architecture that automatically creates the curriculum during training. In either case, the curriculum is formed using some information-theoretic measure of difficulty (semantic distance, feedback from a separate network, etc.), and it is difficult to interpret why they work well for some tasks and not others. In a practical sense, it is seldom clear how to apply a previously investigated curricula directly to another task.

In this paper, we explore how to address these shortcomings by creating what we call a *task-agnostic* model of difficulty, which we argue can, in principle, be applied to any supervised learning task. We use this model to investigate what makes a good difficulty measure for curricula beyond how it affects downstream performance to better explain why one curriculum should be preferred over another. We use the well-known SNLI dataset for our experiments (Bowman et al., 2015), which provides a high-quality data source for the natural language inference (NLI) task. Given a premise sentence, the goal is to predict whether a hypothesis sentence is entailed by the premise, contradicted by it, or is neither entailed nor contradicted by it (neutral). Unlike many NLI datasets, SNLI includes in its dev set the responses from each crowd-sourced worker who attempted to answer the question, giving us a set of 10,000 annotated questions to create a difficulty model. We release all code related to our

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop, pages 16–23

experiments on GitHub.[1]

2 Related Work

The idea of applying concepts from developmental psychology to artificial intelligence traces as far back as Turing (1950). Although much work over the years has explored variants of this idea, mostly under the subject title "Developmental AI" (Elman, 1993; Shultz, 2003; Shultz and Sirois, 2008; Asada et al., 2009; Bruce, 2010; Guerin, 2011; Licato and Bringsjord, 2016; Vernon et al., 2016), only recently has the target task been defined as that of determining how to organize training data of machine learning models so that they can benefit from the kind of scaffolded learning that a human tutor might provide to a child.

Bengio et al. (2009) demonstrated that following a curriculum can provide a more optimal solution during gradient descent. Other early work includes Krueger and Dayan (2009), which experimented with shaping tasks (Skinner, 1938) learned by a LSTM model. These early results suggest that, for many learning tasks, estimating the difficulty of problems in a training set may indeed be simpler than training on random permutations of the dataset, but determining an effective measure of difficulty remains far from trivial. Previous work has addressed this problem in two fundamentally different ways: by using a separate model from that used on the main task itself to predict difficulty (Kumar et al., 2010; Graves et al., 2017; Jiang et al., 2018; Matiisen et al., 2019; Shen and Feng, 2020); or using a manually designed measure of difficulty not obtained through learning (Platanios et al., 2019; Liu et al., 2020; Xu et al., 2020).

A crucial question which recent AI work has not sufficiently addressed is how humans design curricula, which was stressed as essential to understand by early research in this area (Bengio et al., 2009; Taylor, 2009). For example, much has been said about the use of *scaffolding* in child education, in which a student is provided examples by a more experienced (typically older) tutor, such that those examples are neither "too easy" nor "too hard" given their current skill level (a space called the zone of proximal development) (Vygotsky, 1978). In reinforcement learning, there has been fruitful research that applies these ideas by using human feedback to manually create curricula (Stanley et al., 2005; Thomaz and Breazeal, 2006; Suay and Chernova, 2011; Loftin et al., 2016). We seek to apply these ideas in the context of supervised learning.

[1] https://github.com/AMHRLab/supervised-CL

3 Difficulty Model Experiments

Using the SNLI dev set's annotation data, our goal is to train a model that can estimate the difficulty of the questions in the train set, hence creating a measure of difficulty. Given the prior success of human-in-the-loop training, we believe this provides a natural way to estimate difficulty for any task where human annotation data can be collected. Each question in the dev set contains 5 different labels assigned by each annotator, along with the gold label deemed correct for the question. The model's specific objective is to predict how many of 5 possible respondents predicted the gold label correctly, and this percentage is the approximate difficulty.

The most intuitive solution to this problem is to treat it as a regression task, and we experiment with several regression models (Section 3.1). However, because the notion of difficulty among humans is a somewhat fuzzy concept, it's unclear to us if a floating-point difficulty score will be meaningful up to arbitrary precision. Therefore, we also experiment with framing the objective as a classification task by mapping the 5 possible percentages into labels and then training various classification models (Section 3.2).

Using features from prior work, we create 5 different feature sets as inputs to both sets of models. The first is sentence length, which has demonstrated to be useful to measure difficulty in machine translation tasks (Platanios et al., 2019). We adapt this to work for SNLI by taking the sentence length of the premise and hypothesis. Second is the RoBERTa sentence embeddings of the premise and the hypothesis. We feed the raw embeddings in by concatenating the entire hypothesis vector to the premise vector, and also by stacking them pairwise such that dimension i of the premise vector is paired with dimension i of the hypothesis vector in the final representation. We refer to this as the flattened embedding. Finally, we use cosine similarity to measure the semantic distance between the input sequences. Table 1 summarizes the various feature sets we have used.

3.1 Regression

Table 2 shows the Spearman's correlation (Spearman, 1904) measured for all regression models.

Feature set
1: cosine similarity + sentence length
2: flattened embeddings
3: concatenated embeddings
4: concatenated embeddings + sentence length
5: concatenated embeddings + cosine similarity

Table 1: Feature sets used for experiments

We experiment with various linear, non-linear, and neural models and perform a grid search over each model's hyperparameters to ensure we have achieved an optimal correlation in each case. For each trial, we shuffle the dataset before training. We use the scikit-learn[2] implementation of all the non-neural models, and scipy[3] to measure correlation. A decision tree using feature set 1 with a maximum depth of 40 achieves the best correlation of all models, with an average observed correlation of 84%. Support vector regression using feature set 2 with C=1 also achieves a strong mean correlation of about 74%. To verify that results for these top-performing models achieve statistically significant results, we also perform 10-fold cross-validation experiments for each of them. The decision tree model achieves a mean correlation of 0.895 across all folds of cross-validation, and the SVR model 0.749.

We conduct two types of neural regression experiments: finetuning a pretrained RoBERTa model and training a multi-layer deep neural network (DNN) from scratch. For the DNN model, we experiment with a simple feed-forward neural network with {2,3,4,5} hidden layers and dimension size of {512, 1024, 2048}. As common with many large language models used for relatively small datasets, RoBERTa quickly overfits to the data and was unable to generalize to the test set. DNN models, on the other hand, failed to capture any significant relationship between the feature sets and the difficulty estimation, as they performed the worst out of all models. We ran an extensive hyperparameter search over the learning rates, batch sizes, the number of layers, and the number of epochs for both RoBERTa and DNN experiments. We use the simpletransformers[4] implementation of RoBERTa and PyTorch[5] for the DNN experiments.

[2]https://scikit-learn.org/stable/index.html
[3]https://www.scipy.org
[4]https://github.com/ThilinaRajapakse/simpletransformers
[5]https://pytorch.org/

Model	Mean	Min	Max
SVR	0.737	0.714	0.755
Decision Tree	0.84	0.801	0.891
Linear	0.299	0.277	0.334
KNN	0.361	0.333	0.383
DNN	0.151	0.09	0.166
RoBERTa-large	0.264	0.215	0.35

Table 2: Spearman's correlation for all regression models.

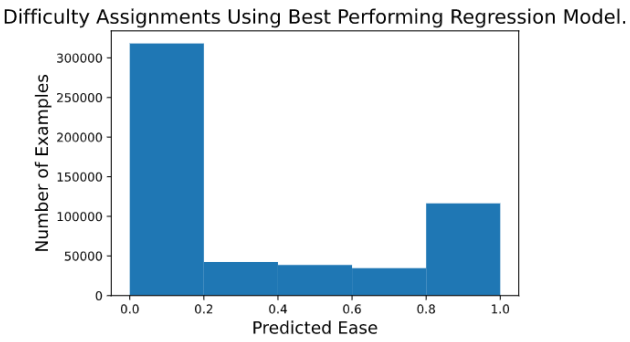

Figure 1: Histogram of difficulty scores for SNLI using decision tree regression, which was the best performing regression model.

3.2 Classification

To perform classification, we map the five possible regression scores onto discrete labels. However, after mapping, we found that the resulting classes were imbalanced. For instance, the number of examples where all annotators correctly predicted the gold label was 10 times more than the examples where no one predicted the label correctly. To address this issue, we oversampled the minority classes using imblearn[6] before training our models. Table 3 shows performance statistics for each model. We again perform a grid search to optimize the feature set choice and hyperparameters for each model. Similar to regression, a decision tree classifier with feature set 1 and a depth of 40 achieves the best performance by a large margin. We found that using either Gini impurity (Breiman et al., 1984) or entropy to measure the quality of the splits achieved about the same performance. We additionally evaluate the same decision tree model using 10 fold cross-validation and achieve the same mean accuracy.

3.3 Analysis of Models

The previous results demonstrate that we can create a model that significantly correlates with the actual

[6]https://github.com/scikit-learn-contrib/imbalanced-learn

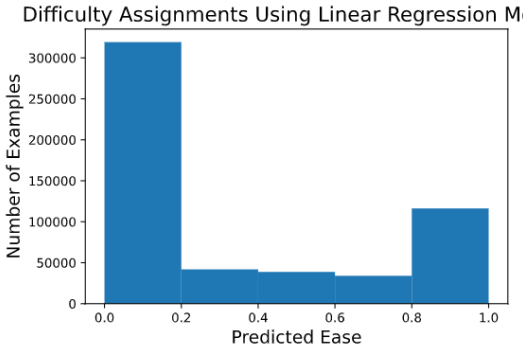

Figure 2: Histogram of difficulty scores for SNLI using the linear regression model.

Model	Mean	Min	Max
SVC	0.304	0.291	0.312
Decision Tree	0.834	0.821	0.854

Table 3: Accuracy for all classification models.

human responses to the SNLI dev set. We now perform additional quantitative and qualitative analyses to gain a better understanding of how well these models actually predict difficulty, and by extension how good any curriculum resulting from it will be. Previous work has tended to evaluate the quality of the curriculum primarily in terms of downstream performance; a curriculum is considered good if using it improves performance or reduces training time. However, many factors will ultimately affect model performance beyond the use of any curricula (hyperparameters, training strategy, etc). In the context of human education, it has long been appreciated that curricula are complex and cannot be evaluated using any single metric (Macdonald, 1971; Kliebard, 1989; Kelly, 2009; Pinar, 2012). Therefore, we believe it is important to attempt to evaluate the quality of our model separate from its application to any task.

Using the best regression and classification models, we predict the difficulty of each problem in the SNLI train set. We additionally do the same using the linear regression model, which achieves only weak correlation, as a point of comparison. Figures 1 and 2 show the distributions of difficulty assignments for the best performing regression model (the decision tree) and the linear regression model, respectively. Higher scores represent easier problems in both figures. Table 4 shows examples rated with both maximum and minimum difficulty from both models. Interestingly, both models predict the same general trend in the difficulty distributions. However, as Table 4 shows there are cases where the models make substantially different difficulty predictions about the same question. The question in the first row of the easy column is rated with minimum difficulty by the best performing model, whereas the same question is rated with maximum difficulty by the linear model. To better understand the differences between each model's predictions, we computed the difference between the predicted difficulty scores for each question in the train set and normalized them by taking the absolute value. About 27% of the questions have the same prediction from both models (difference of estimated score is 0), and about 15.8% have polar opposite predictions (the difference is 1). Furthermore, the mean difference between the scores from both models is 0.41, suggesting that they make substantially different predictions even though they report the same overall distribution.

Figure 3 shows the distributions of difficulty scores from the classification models. We similarly observed wide variation in the assigned difficulty, however in this case most problems are assigned as being very easy. This is probably the most accurate model, given that in the majority of cases the annotators were able to correctly predict the gold label (Figure 4).

These results give us some confidence that our curriculum may lead to a performance gain on SNLI, given a suitable model to train it with. Designing a curriculum for both humans and machine learning models requires that there be some difference in difficulty among the questions in the task, otherwise any random permutation would be essentially the same curriculum. That all models are reporting variance in the difficulty distribution indicates that there may be enough difference in difficulty for curriculum learning to help in this case. We hypothesize that the linear model, despite having only very weak correlation with the human responses, is still able to capture the high level structure of the difficulty distribution, which would explain why the two histograms are identical. However, when we examine the predictions at a finer level it becomes obvious the models are making fundamentally different regression decisions.

4 Conclusion and Future Work

In humans, teaching, especially according to a curriculum, reduces the difficulty of learning complex skills by providing a simpler path for learners to

Easy	Hard
Decision Tree Model: *Premise:* A white dog runs through a field. *Hypothesis:* A white dog is running back to his master. neutral	*Premise:* A man is surfing in a bodysuit in beautiful blue water. *Hypothesis:* On the beautiful blue water there is a man in a bodysuit surfing. entailment
Premise: A group of workers are posing for a picture. *Hypothesis:* A group of workers are playing baseball. contradiction	*Premise:* A man is kneeling in the top step while many people are behind him sitting in chairs. *Hypothesis:* A man sleeps comfortably at home. contradiction
Linear Regression Model: *Premise:* A small child is running on the shore of the beach surrounded by birds. *Hypothesis:* Birds surround a child looking for food. neutral	*Premise:* A white dog is running through a field. *Hypothesis:* A white dog is running back to its master. neutral
Premise: A group of people are hiking in the forest. *Hypothesis:* People are hiking in the forest. entailment	*Premise:* An old man is standing before a crowd to perform a feat. *Hypothesis:* An oldtimer about to perform a group of people. entailment

Table 4: SNLI questions predicted as being very easy and as very challenging by both the decision tree model and the linear regression model. The first two rows are predictions from decision tree model and the last two rows are from the linear regression model. The left column are questions rated as **Easy** by the respective model, while the right column are questions rated as **Hard**. Gold labels are shown in blue

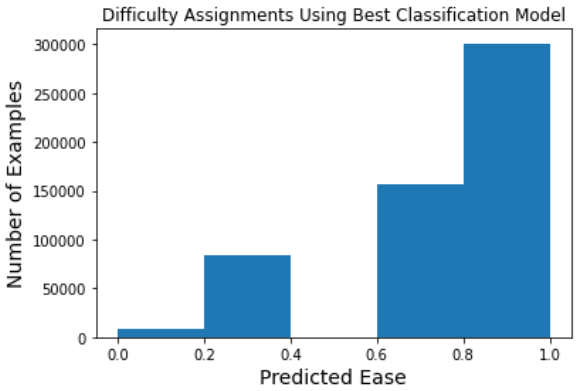

Figure 3: Histogram of difficulty scores for SNLI using the best classification model.

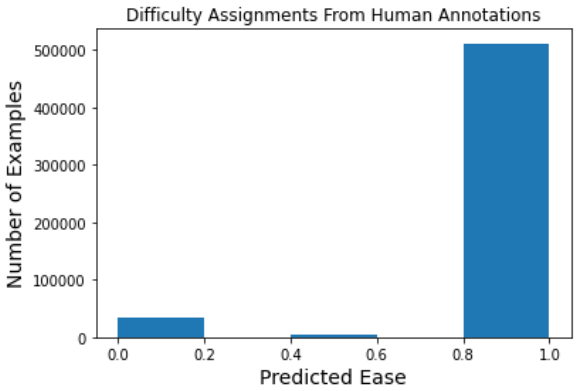

Figure 4: Histogram of difficulty scores for SNLI using the human annotations.

follow. While much work has explored various applications of curriculum learning over the years, the larger problem of what makes a good difficulty measure remains an important research question. In this work, we have presented initial studies on designing a task-agnostic measure of difficulty for curricula. While for practical reasons, we chose to focus specifically on an NLP task, in principle our approach can be extended to any supervised learning task where it is possible to gather human annotations. Our analysis suggests that the resulting curriculum is suitable for use in a curriculum learning algorithm, firstly because it correlates well with human difficulty estimates, and secondly because it displays wide differences in the estimated difficulties across all the problems in the train set.

An important question to answer next is how to actually apply the curriculum to the training of a model. While this may appear trivial, the wide array of factors that affect downstream performance means that it cannot be taken lightly. The model choice is an especially important factor; using a re-

cent model which already does very well on SNLI (Zhang et al., 2020) is unlikely to have any effect because performance on the task has saturated at this point. However, if we apply a model which is too simplistic, for instance, the classic n-gram language model, then it's unlikely that even the best curriculum will help such a model to learn a task which is probably far too challenging for it. A study examining how our curriculum affects downstream performance for models ranging from the state-of-the-art to the baseline would help answer this question.

Another important factor is the sampling protocol. Feeding examples to the model in order of difficulty, with easier ones preceding harder ones, is the most straightforward solution. However, there is no guarantee that this is the optimal way to feed examples into the model. In fact, there have been cases where feeding examples in *reverse* order, with hard examples preceding easier ones, has lead to optimal performance improvement (McCann et al., 2018). Future work will also investigate the effect of various sampling strategies.

Finally, something which should be considered is the quality of the gold labels which our models are trying to predict. Since NLI is an inherently ambiguous task, determining the ground truth of a given NLI question is challenging. Recent work by Nie et al. (2020b) has shown that many of the gold labels for both SNLI and the MNLI dataset (Williams et al., 2018), which also includes the responses of individual annotators, will change when more annotators are used with stricter quality control protocols, though they only found this to be true when agreement on the correct label was low to begin with. Our choice for using SNLI specifically is that, unlike more recent datasets (Bhagavatula et al., 2020; Nie et al., 2020a), it also includes the label predicted by each individual annotator and not just the final gold label. We might be able to account for the fuzziness of the ground truth in the final difficulty prediction of our model. For instance, if the agreement of the gold label for a question is low, we could take this into consideration and predict the question as being even more difficult. This avenue will be explored as a way to improve the quality of the difficulty model, using both SNLI and similar datasets, such as MNLI, as part of a more comprehensive study.

Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under award numbers FA9550-17-1-0191 and FA9550-18-1-0052. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

References

Minoru Asada, Koh Hosoda, Yasuo Kuniyoshi, Hiroshi Ishiguro, Toshio Inui, Yuichiro Yoshikawa, Masaki Ogino, and Chisato Yoshida. 2009. Cognitive Developmental Robotics: A Survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive Commonsense Reasoning. In *ICLR 2020 : Eighth International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees*. CRC press.

David Bruce. 2010. Cognitive Architecture and Testbed for a Developmental AI System. Undergraduate Honours Thesis.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1311–1320. JMLR. org.

Frank Guerin. 2011. Learning Like a Baby: A Survey of Artificial Intelligence Approaches. *The Knowledge Engineering Review*, 26(2):209–236.

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*, pages 2304–2313.

Albert Victor Kelly. 2009. *The curriculum: Theory and practice*. Sage.

Herbert M Kliebard. 1989. Problems of definition in curriculum. *Journal of Curriculum and Supervision*, 5(1):1–5.

Kai A Krueger and Peter Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197.

John Licato and Selmer Bringsjord. 2016. A Physically Realistic, General-Purpose Simulation Environment for Developmental AI Systems. In *Proceedings of the ECAI 2016 Workshop on Evaluating General-Purpose AI (EGPAI 2016)*.

Xuebo Liu, Houtim Lai, Derek F Wong, and Lidia S Chao. 2020. Norm-Based Curriculum Learning for Neural Machine Translation. *arXiv preprint arXiv:2006.02014*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Robert Loftin, Bei Peng, James Macglashan, Michael L. Littman, Matthew E. Taylor, Jeff Huang, and David L. Roberts. 2016. Learning behaviors via human-delivered discrete feedback: Modeling implicit feedback strategies to speed up learning. *Autonomous Agents and Multi-Agent Systems*, 30(1):30–59.

James B Macdonald. 1971. Curriculum theory. *The Journal of Educational Research*, 64(5):196–200.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2019. Teacher-Student Curriculum Learning. *IEEE Transactions on Neural Networks*, pages 1–9.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The Natural Language Decathlon: Multitask Learning as Question Answering. *arXiv preprint arXiv:1806.08730*.

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 2020: 58th annual meeting of the Association for Computational Linguistics*, pages 4885–4901.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020b. What Can We Learn from Collective Human Opinions on Natural Language Inference Data. *arXiv preprint arXiv:2010.03532*.

William F Pinar. 2012. *What is curriculum theory?* Routledge.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based Curriculum Learning for Neural Machine Translation. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1162–1172.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683*.

Lei Shen and Yang Feng. 2020. CDL: Curriculum Dual Learning for Emotion-Controllable Response Generation. In *ACL 2020: 58th annual meeting of the Association for Computational Linguistics*, pages 556–566.

Thomas R. Shultz. 2003. *Computational Developmental Psychology*. The MIT Press, Cambridge, Massachusetts.

Thomas R. Shultz and Sylvain Sirois. 2008. Computational Models of Developmental Psychology. In Ron Sun, editor, *The Cambridge Handbook of Computational Psychology*, chapter 16, pages 451–476. Cambridge Univ Press, New York, New York, USA.

BF Skinner. 1938. The behavior of organisms: an experimental analysis.

Charles Spearman. 1904. The proof and measurement of association between two things. *American journal of Psychology*, 15(1):72–101.

Kenneth O Stanley, Bobby D Bryant, and Risto Miikkulainen. 2005. Evolving neural network agents in the NERO video game. *Proceedings of the IEEE*, pages 182–189.

Halit Bener Suay and Sonia Chernova. 2011. Effect of human guidance and state space size on Interactive Reinforcement Learning. In *2011 RO-MAN*, pages 1–6.

Matthew E. Taylor. 2009. Assisting Transfer-Enabled Machine Learning Algorithms: Leveraging Human Knowledge for Curriculum Design. In *AAAI Spring Symposium: Agents that Learn from Human Teachers*, pages 141–143.

Andrea L. Thomaz and Cynthia Breazeal. 2006. Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance. In *AAAI'06 Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, pages 1000–1005.

Alan M. Turing. 1950. Computing Machinery and Intelligence. *Mind*, pages 433–460.

David Vernon, Claes van Hofsten, and Luciano Fadiga. 2016. Desiderata for Developmental Cognitive Architectures. *Biologically Inspired Cognitive Architectures*, 18:116–127.

Lev Vygotsky. 1978. *Mind in Society: The Development of Higher Psychological Processes*.

Wei Wang, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2020. Learning a Multi-Domain Curriculum for Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7711–7723.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

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*, pages 6095–6104.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In *EMNLP 2018: 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuiliang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for Language Understanding. *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 34(5):9628–9635.

Yikai Zhou, Baosong Yang, Derek F Wong, Yu Wan, and Lidia S Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944.