Curriculum Learning for Neural Machine Translation

Stanford CS224N {Custom} Project

Jo Chuang
Department of Computer Science
Stanford University
jochuang@stanford.edu

Kevin Tan
Department of Computer Science
Stanford University
kevtan@stanford.edu

Abstract

Curriculum learning (CL) is a training framework from Reinforcement Learning (RL) in which easy training examples are initially seen and the difficulty of examples are gradually increased over time. In this project, we aim to (1) reproduce CL for neural machine translation (NMT), and (2) improve CL by leveraging additional sources of self-supervision. For part (1), we have produced our own implementation of curriculum learning and performed initial iteration on a baseline RNN model (BiLSTM with attention); we will be replicating and running Transformer based experiments soon. For part (2), we present concrete experimental ideas for experiments on how to improve CL for NMT with backtranslation, pre-trained language model difficulty scoring and competency control.

1 Approach

1.1 Baselines

Our baseline in this case would be a vanilla model trained without CL. We use the code from the A4 assignment. Two model types are considered for all experiments: a BiLSTM with Attention [1] (which was evaluated for the milestone) as well as a Transformer [2]. The baselines are trained as close to the original baselines in [3].

1.2 Main Approach

We reproduced the experiments in [3] for the *competence-based curriculum learning* framework, the idea that training can perform better if the examples are seen in an ordering that is appropriate for the model's current *competence*. Specifically, we implemented the *difficulty* and *pacing* functions from scratch, and made appropriate modifications to the data loader from the A4 assignment. Apart from the A4 assignment, we implemented everything else ourselves. In the following, we describe in more detail the difficulty functions, pacing functions, and training procedure for curriculum learning.

More details are available in the Appendix, including equations for specific formulations.

2 Experiments

2.1 Dataset

For fair comparison to the original paper, we use the same datasets in from [3], specifically IWSLT-15 $En \rightarrow Vi$ (133k train, 768 dev, 1268 test), IWSLT-16 $Fr \rightarrow En$ (224k train, 1080 dev, 1133 test), and WMT-16 $En \rightarrow De$ (4.5m train, 3003 dev, and 2999 test). We used IWSLT-16 $Fr \rightarrow En$ for our first replication effort.

2.2 Evaluation Method

The evaluation metric we use at test time are the BLEU scores and the time it takes for the models using curriculum learning to obtain a BLEU score that the baseline model attains at convergence. These are the same metrics used in [3]. Also important is the Test set BLEU score over time, which is a key indicator of how quickly the models converge under different settings.

2.3 Experimental Details

At each iteration for epochs $i = 1, ..., \texttt{max\_epoch}$, we retrieve the current curriculum training and validation sets according to the current time $t = \texttt{train\_iteration}$ and difficulty function (e.g. *rarity*). We trained each model for a maximum of 5 epochs, or about 28000 iterations, as the models generally converged quite early. We use default Adam with $LR = 0.001$. For evaluation, we conduct beam search with beam size of 5. We clip gradients at a maximum of 5. The architecture used follows the original Seq2Seq with Attention model from Bahdanau et al[1].

2.4 Results

Our results indicate that, with Rarity scoring and the BiLSTM attention model, we have successfully reproduced similar results compared to the original paper [3]. We note that most of the benefits from using CL were more apparent with training Transformers, and that the original paper saw a limited speedup in convergence for BiLSTM models. Reference the Appendix for the resulting graphs.

3 Future work

First, we would like to match all of the experiments from the paper, including Length difficulty scoring as well as Transformer experiments. We wrote our own implementation of the original Transformer [2], but running it directly with hyperparameters specified in Platanios et al. [3] did not result in a converging model, as Transformers are generally oversentive to learning rate tuning. We will attempt to first train a Transformer with LR scheduling as specified in general folklore, then attempt a fixed LR schedule with CL as the original authors claim that CL reduces the reliance on LR tuning.

We would then like to explore the use of BERT as a difficulty scoring function. Fundamentally, a language model models the probability of a certain utterance, which can be interpreted as a difficulty metric. We would generate a scoring for each sentence by feeding tokens into a pretrained BERT model and obtaining next-word probabilities.

Other various improvements will be attempted, time permitting. Examples include: moving to use BPE encodings, using perplexity as a proxy for competence.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[3] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. Competence-based curriculum learning for neural machine translation. *CoRR*, abs/1903.09848, 2019.

A Appendix

Difficulty Functions. The difficulty is a value that represents the difficulty of a particular training example. The resultant ordered dataset is an sorting of the dataset based on each example's difficulty from easy to hard. See Table 1 for samples of easy, medium, and hard examples, alongside their difficulty values.

The intuition behind the *rarity* difficulty is that – in early stage of training – the model will benefit from examples containing words (in both the source and target) that occur with high frequency compared to examples with low word frequencies. This can be viewed as a way of bootstrapping the model to learn quickly during early training.

More formally, given a corpus of sentences $\{s_i\}_{i=1}^{M}$, the word frequencies are defined as (following [3]):

$$\hat{p}(w_j) = \frac{1}{N_{total}} \sum_{i=1}^{M} \sum_{k=1}^{N_i} \mathbf{1}_{w_k^i = w_j} \tag{1}$$

where $j = 1, ..., n_{unique}$ and $\mathbf{1}$ is the indicator function, and N_{total} is the sum of all of the word occurrences in the corpus.

Given the relative word frequencies $\hat{p}(w_j)$, the sentence rarity difficulty is then:

$$d_{rarity}(s_i) = -\sum_{k=1}^{N_i} \log \hat{p}(w_k^i) \tag{2}$$

Pacing Functions. The pacing function returns a value between 0 and 1 that represents the progress of the model along the curriculum during training. More specifically, $c(t)$ at time t (measured in training iterations) is the fraction of training data it is allowed to see at the current iteration.

We implemented the linear and *root* pacing functions (see Fig 2 for a visualization).

The linear pacing function defines $c(t)$ as follows. Given a bias $c_0 > 0$, and a slope $r = (1 - c_0)/T$ where T is the time after which the learner is fully competent:

$$c_{linear}(t) = \min(1, tr + c_0) \tag{3}$$

Similarly, the root pacing function is:

$$c_{sqrt}(t) = \min\left(1, \sqrt{t\frac{1 - c_0^2}{T} + c_0^2}\right) \tag{4}$$

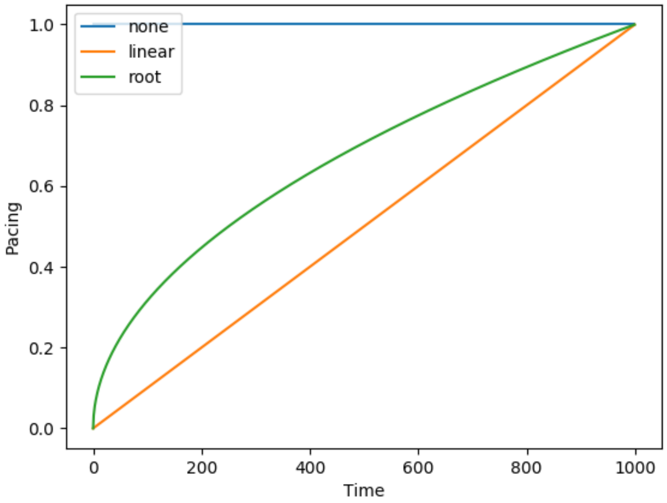

Figure 1: Overview of proposed pacing functions with $c_0 = 0.01$ (initial competence value) and max time $T = 1000$ (total duration of the curriculum)

French-English translations

	Easy example	**Rarity**
src	Merci	7.52
tgt	Thank you	21.50
	Medium example	
src	Pourtant, seulement cent ans plus tard, les 3/4 d'entre nous se font incinérer	115.06
tgt	And yet, only a hundred years later, three quarters of us get cremated	114.31
	Hard example	
src	De la collectivisation radicale des terres à la campagne du Grand Bon en avant, puis la privatisation des terres, puis la Révolution Culturelle, puis la réforme du marché mise en œuvre par Deng Xiaoping, puis son successeur Jiang Zemin a pris l'énorme initiative politique d'ouvrir l'adhésion au Parti, aux entrepreneurs du secteur privé, quelque chose d'inimaginable quand Mao était aux commandes	556.69
tgt	From radical land collectivization to the Great Leap Forward, then privatization of farmland , then the Cultural Revolution, then Deng Xiaoping 's market reform, then successor Jiang Zemin took the giant political step of opening up Party membership to private business people, something unimaginable during Mao's rule	454.23

Table 1: Examples of sentences according to the *rarity* difficulty. For each example, we show the source (*src*) and target (*tgt*) translation from the ordered dataset, as well as the *rarity* score.

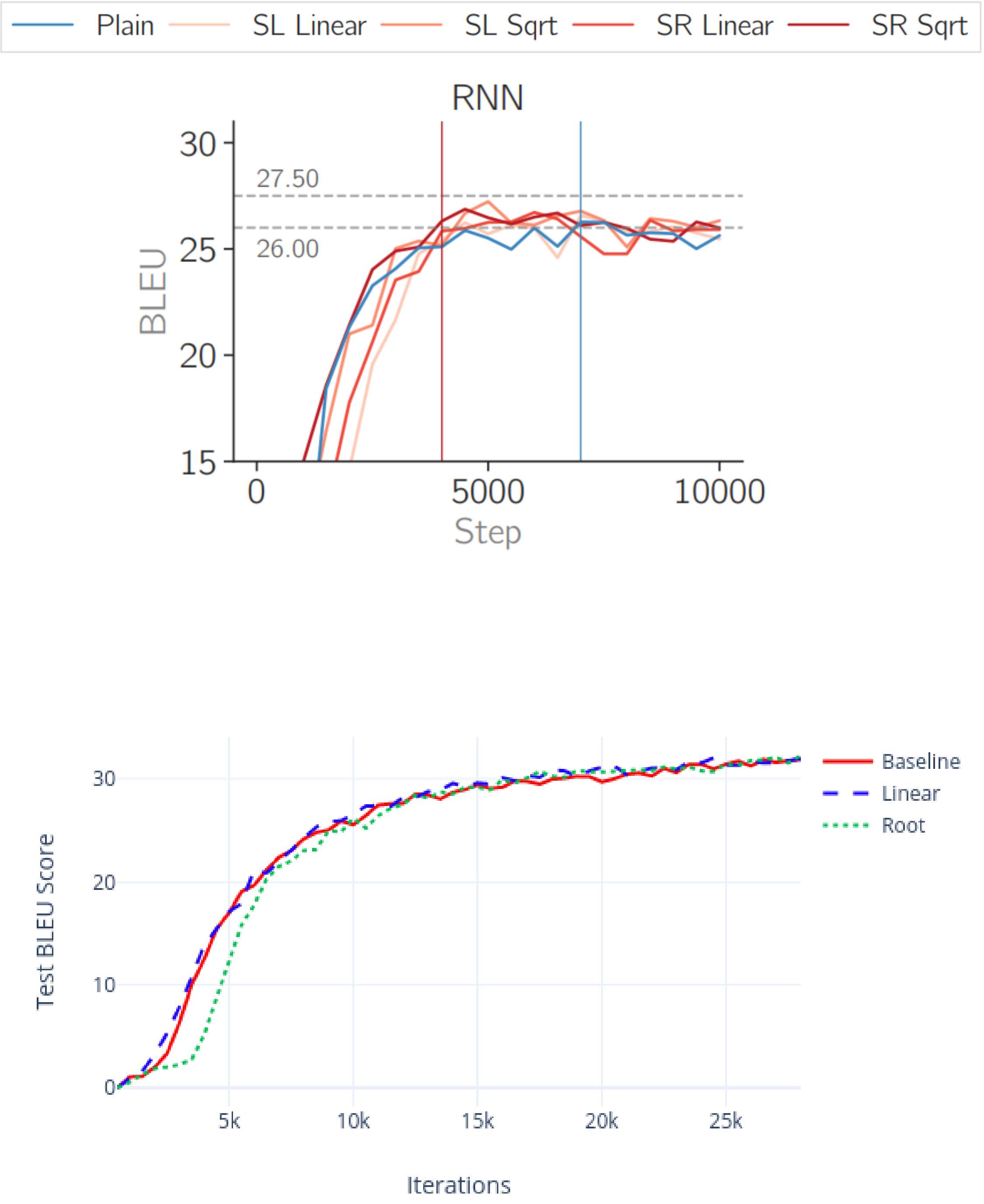

Figure 2: (Top) BLEU score on the test set over training iterations from [3]. (Bottom) BLEU score results from our implementation. Note our models converged slightly slower but achieved higher BLEU scores. Overall, Curriculum learning only slightly affects BLEU score convergence in both the original paper and ours.