School of Informatics

Informatics Research Review
Curriculum Learning for Neural Machine Translation

B242552
January 2024

Abstract

Neural Machine Translation has advanced the state-of-the-art on various translation tasks. However, training large-scale NMT systems can be challenging, as it involves complex heuristics that can be time-consuming and costly to optimize. In this review, I investigate if curriculum learning (CL) can improve translation performance and reduce the training time of NMT systems. Qualitative and quantitative analysis confirm the above hypothesis. However an unified evaluation benchmark is required for accurate comparison of various CL approaches.

Date: Sunday 28th January, 2024

Supervisor: Xingran Ruan

1 Introduction

Machine translation converts a sentence from the source language into the target language. Neural Machine Translation (NMT) involves a singular, extensive neural network to learn the conditional distribution of training examples and utilizes an encoder-decoder structure (Sutskever et al. 2014, Bahdanau et al. 2014). The encoder transforms the source sentence into a hidden representation, while the decoder predicts the subsequent target word based on the hidden vector and previously predicted words. NMT has become the de facto standard on various translation tasks. However, training large-scale NMT systems such as Transformers (Vaswani et al. 2017) is very time-consuming. During the training process, the examples are fed to the model randomly, significantly increasing the training time (Popel & Bojar 2018).

One way to speed up the training process of NMT is to utilize Curriculum Learning (CL) (Elman 1993, Krueger & Dayan 2009, Bengio et al. 2009). The concept of CL can be traced back to Elman (1993). The fundamental idea of the paper is to begin with simple tasks, master the easier aspects, and then progressively increase the level of difficulty. The learning process for both humans and animals is significantly improved when examples are arranged in a logical sequence that progressively introduces new concepts and increases in complexity, rather than being presented in a random order. In machine learning, the curriculum involves controlling order to present the samples in a strategically sequenced manner, aligning with the model's evolving understanding and competence during the training process.

There are two primary reasons for using Curriculum Learning (CL). The first is to *guide* and regularize the training process towards more optimal areas in the parameter space, considering it from an optimization problem perspective. The second is to *denoise*, which involves concentrating on easier, high-confidence areas to reduce the impact of noisy data, viewed from a data distribution perspective. Most CL applications can be grouped into these two categories (Wang et al. 2020). Bengio et al. (2009) confirm that a carefully selected curriculum strategy improves generalization and speeds up convergence in various tasks, such as shape recognition, language modelling, and deep neural network training.

The paper starts with a review of recent literature that utilized CL for NMT in the past five years in chronological order followed by a thematic discussion. It does not review the papers focusing on parallel well-known research topics in CL for NMT such as (1) CL for domain adaptation of NMT models; and (2) CL for non-autoregressive translation (NAT). This literature review aims to analyze the effectiveness of using CL for NMT and identify the current state-of-the-art methods. Furthermore, the paper also focuses on critically examining the strengths and weaknesses of various difficulty metrics and curriculum schedules used to implement CL in NMT.

2 Background

2.1 Curriculum design

To design a curriculum for students, teachers need to organize material based on its difficulty using a *difficulty measurer*. Also, teachers need to determine the optimal pace at which the material should present to the students to maximize learning using a curriculum scheduler or *training scheduler*. While rushing through basic concepts may result in more harm than good, a slow pace can lead to disinterest and ineffective learning (Ornstein & Hunkins 2017). There are two major types of CL based on the tasks outlined above: predefined CL; and automatic CL. In

predefined CL, a human expert design both the difficulty measurer and training scheduler. If any of the two tasks are learned by data-driven algorithms or models, the CL method is called automatic CL.

Automatic CL can be classified into three major methods: (1) Self-paced learning (SPL) methods involve the student or model measuring the difficulty of examples based on the losses; (2) Transfer Teacher methods use a teacher or a pre-trained network calculating the difficulty of the examples; and (3) Reinforcement Learning (RL) Teacher methods use RL models or agents to serve as the teacher and perform dynamic data selection based on student feedback.

2.2 BLEU

BLEU (Papineni et al. 2001), or Bilingual Evaluation Understudy, is a metric used to evaluate the quality of a machine-translated text. The score reflects the similarity of the machine-translated text to a set of high-quality reference translations. However, BLEU does not consider factors like intelligibility or grammatical correctness. It's also important to note that BLEU scores should only be compared when the test set, language pair, and Machine Translation engine are the same. A different test set will yield a different BLEU score. The ultimate goal of Machine Translation, and thus the BLEU score, is to produce results that match the quality of a professional human translator.

3 Literature Review

Paper	Criteria	Schedule	Dataset	BLEU
Kocmi & Bojar (2017)	length of sentence, number of coordinating conjunctions, word frequency	batching	WMT-17 ($En \rightarrow Cz$)	15.24
Zhang et al. (2018)	confidence; sentence length; word rarity	batching	MTTT corpus (Duh, 2018) ($De \rightarrow En$)	28.4
Platanios et al. (2019)	sentence length; word rarity	sampling	IWSLT-15 ($En \rightarrow Vi$)	29.81
			IWSLT-16 ($Fr \rightarrow En$)	35.83
			WMT-16 ($En \rightarrow De$)	30.16
Wang et al. (2018)	noise	batching	Paracrawl ($En \rightarrow Fr$)	37.5
			WMT ($En \rightarrow Fr$)	37.7
Kumar et al. (2019)	noise	batching	Paracrawl ($En \rightarrow Fr$)	37.5
			WMT ($En \rightarrow Fr$)	38.4
Zhou et al. (2020)	uncertainty	batching	IWSLT-15 ($En \rightarrow Vi$)	30.75
			WMT-17 ($Zh \rightarrow En$)	25.04
			WMT-16 ($En \rightarrow De$)	33.85
Liu et al. (2020)	norm	sampling	WMT-14 ($En \rightarrow De$)	28.81
			WMT-17 ($Zh \rightarrow En$)	25.25
Zhao et al. (2020)	teacher network supervision	stages	-	-

Table 1: Overview of papers that utilize CL for NMT

3.1 An Empirical Exploration of Curriculum Learning for Neural Machine Translation by Zhang et al. (2018)

Zhang et al. (2018) show that CL accelerates NMT training time without any significant loss in quality. The difficulty for a sentence pair is based on model and linguistic difficulty. Hence the paper examines both predefined and automatic CL. The model difficulty score is the probability of the one-best translation from an auxiliary NMT model based on RNN given the input sentence. This serves as a measure of prediction confidence of the model for the translation. This automatic difficulty measure falls under the *transfer teacher* methodology of automatic CL. The linguistic difficulty score is based on sentence length and vocabulary frequency of the sentence pair.

The authors use the *baby step* scheduler (Bengio et al. 2009) and organize the samples into shards (buckets) based on the Jenks Natural Breaks classification algorithm (Jenks & Geography 1977). The baseline model uses the entire training set bucketed by sentence length. A Long Short Term Memory network is trained with the *default* curriculum schedule (easy shards first) along with four variants such as *reverse*, *boost* (addition of a copy of hardest shard to the training data), *reduce* (removal of easy shards) and *no-shuffle* (no shuffling among buckets).

The experiments confirm that 20 of 100 CL strategies converge faster than the baseline without a loss in BLEU but there is not single CL strategy that clearly outperformed others. The results are also sensitive to hyperparameters such as initial learning rate which is one of the disadvantages of predefined CL. The paper provides a foundational understanding of using CL for NMT but it has the following limitations: (1) Sentence length and vocabulary frequency cannot model the linguistic difficulty of the sentence comprehensively and introduces many hyperparameters that are difficult to tune; and (2) The results of the paper may not hold good for complex networks such as the Transformer (Vaswani et al. 2017) network.

3.2 Denoising Neural Machine Translation Training with Trusted Data and Online Data Selection by Wang et al. (2018)

This paper proposes a method for denoising neural machine translation (NMT) training using online data selection. The authors hypothesize that training on noise-reduced data batches can improve the NMT performance and robustness, especially when the training data is noisy. This paper uses predefined CL with noise as the predefined difficulty metric and continuous pacing function as the curriculum schedule. The authors use a dynamic online data selection strategy to schedule the data batches based on their noise level, which is computed by the pair of noisy and denoised models. The denoised model is generated by fine-tuning the noisy model over a small amount of high-quality trusted data. The noise level of a sentence pair is calculated to be the difference in log probabilities between the noisy model and the denoised model. The selection ratio, which controls the portion of data to select from, is gradually tightened by an exponential decaying function. The authors also compare their method with a language model-based data selection method, which is commonly used for domain adaptation. They observe that NNLM does not discern noise and cannot be used for denoising noisy data.

The authors used seq2seq NMT models for training the models and used different datasets for training and evaluation. They show that their method significantly outperforms the baselines trained on random or LM-selected data batches and that their method correlates better with human ratings of data quality. The results prove that the proposed online denoising is robust to data noise and adapts to model changes over time, thereby providing better performance compared to incremental denoising by fine-tuning. They also show that their method is effective

for both noisy and clean datasets and that it is not a domain adaptation effect.

3.3 Competence-based Curriculum Learning for Neural Machine Translation by Platanios et al. (2019)

State-of-the-art NMT such as Transformers (Vaswani et al. 2017) are slow to train and require heuristics such as learning rate schedules and are trained with large batch sizes to reduce noise. The generic curriculum learning approach suggested by the authors solves the above issues and improves generalization over a variety of datasets. Also, it outperforms Zhang et al. (2018) by proposing the CL strategy with only the duration of the CL as a hyperparameter compared to discrete CL with multiple hyperparameters, thereby achieving a comparative reduction of up to 70% in training time and performance gain of up to 2.2 BLEU.

The authors propose a competency-based predefined CL strategy where the training samples are only sampled if the *difficulty score* of the sample is lower than the current *competence* of the model. The authors used sentence length and word rarity as difficulty metrics. The sentence length of only the source sentence is considered for the experiments. Since Zhang et al. (2018) show that combining word frequencies of words in a sentence into one difficulty score based on maximum, average or minimum does not work well in practice, the authors designed a better difficulty score based on the likelihood of the sentence, which holds information about both sentence length and word frequency. The competency of the model at time t, $c(t)$ is defined in the general form as follows:

$$c_{\text{root}-p}(t) \triangleq \min\left(1, \sqrt[p]{t\frac{1-c_0^p}{T}+c_0^p}\right) \tag{1}$$

where c_0 is the initial competence value, T is the total duration of the curriculum learning phase, and p is a parameter that controls the shape of the curve. The the value of T is task dependent and a vanilla baseline model has to be trained to determine its value. The value of T is set to be 90% of the training steps required for the baseline model to achieve 90% of its final performance. The authors train the model with linear ($p = 1$) and root function ($p = 2$). The motivation for authors to use the root function is to reduce new training examples per unit time as the training progresses, which gives the model sufficient time to understand and assimilate the new information. The above-discussed CL strategy is used to train an RNN based on Bahdanau et al. (2014) and a Transformer based on Vaswani et al. (2017). The experiments show that this CL strategy improves the speed and reliability of training Transformers but has little effect on RNNs. They observe that the curriculum with sentence rarity as difficulty metric and square root function as competence provides the best performance across the experiments. Thus, the competency-based CL strategy outperforms previous CL-based NMT systems and NMT models trained without any curriculum. One of the drawbacks of this approach is that it requires a vanilla baseline model to be trained to set the hyperparameter T for the pacing function.

3.4 Reinforcement Learning based Curriculum Optimization for Neural Machine Translation by Kumar et al. (2019)

The core idea of Kumar et al. (2019) is to treat the curriculum design itself as a learning problem, leveraging the strengths of Reinforcement Learning (RL) to make data-driven decisions about which training examples to present at different stages of the learning process rather than using

prior knowledge. The motivation behind the idea is to learn a curriculum schedule based on the data attributes automatically for any corpora without the need for extensive trial-and-error to find optimal hyperparameters. This paper uses automatic CL with noise as the predefined difficulty metric and RL teacher as the curriculum schedule learner. The authors chose *noise* as the attribute to design the curriculum. The difference between the log-likelihood of the sentence pair in the clean and noisy model represents the quality of the translation pair (Wang et al. 2018). The author uses Deep Q-learning (DQN) proposed by Mnih et al. (2015) as an RL agent to learn the curriculum. The agent's selects actions based on the Q-function, which is a state-action value function. Q-learning is a model-free, value-based, off-policy algorithm in the field of RL. The agent strives to learn this Q-function to optimize the cumulative expected rewards.

The authors create a *prototype* batch with a fixed number of sentences, whose CDS scores are closer to the mean, from each bin of the training data. The *observation* from the *environment* is a vector containing log-likelihoods of sentences in the prototype batch as per the NMT system. The *reward* for the RL agent is the change in the log-likelihood of the NMT system on the development set since the last evaluation. To balance exploration versus exploitation for the RL agent, a linearly decaying ϵ strategy is used with three stages: (1) warmup stage with complete exploration; (2) decay period where exploration reduces and exploitation increases; (3) floor period with complete exploitation. The warmup phase prevents large rewards from being given to RL agents during the start of the training when the likelihood of the NMT model aggressively decays.

The authors used an NMT model similar to RNMT+ (Chen et al. 2018) and trained with training data split into 6 bins based on noise level calculated by CDS score. The learned curricula match the baselines in Paracrawl and beat the baselines in the WMT dataset, showing that Q-learning can find an effective curriculum for NMT training. The authors compare the hand-designed curriculum of Wang et al. (2018) with the policy learned by the RL agent and found that they are different. Thus, RL can be used to learn sophisticated policies based on data attributes and perform better than various sampling and filtering baselines.

3.5 Uncertainty-Aware Curriculum Learning for Neural Machine Translation by Zhou et al. (2020)

Zhou et al. (2020) argue that existing methods of curriculum learning (CL) rely on heuristic measures of data difficulty and pre-defined schedules of model competence, which may not match the actual learning process of NMT models. That is, they argue that the difficulty of a translation cannot be accurately expressed by heuristics such as word rarity or sentence length. Also, they hypothesize that monotonically increasing functions cannot represent the competency of a model during training as proposed by Platanios et al. (2019). To address these limitations, they introduce two types of uncertainty to guide CL: data uncertainty and model uncertainty. This paper falls under automatic CL as it uses transfer teacher to calculate the difficulty metric based on uncertainty and predefined discrete scheduler (baby step).

Data uncertainty quantifies the complexity and rarity of a sentence pair using the cross-entropy estimated by a pre-trained language model. The authors suggest using a difficulty indicator called joint difficulty, which is the sum of uncertainties of the source and target sentences. Model uncertainty measures the confidence of the NMT model on the current training data using the variance of the distribution over the network parameters. The authors use Monte Carlo Dropout (Gal & Ghahramani 2015) to approximate Bayesian inference and calculate the model uncertainty after each epoch.

The authors trained the Transformer network based on Vaswani et al. (2017) using the proposed CL strategy. They show that data uncertainty is more relevant and comprehensive than sentence length and word rarity, which are commonly used as difficulty criteria in CL. Moreover, they demonstrate that using a joint uncertainty of both source and target sentences can further improve the performance of CL. They evaluate their method on three translation tasks and show that it outperforms the strong baseline and related methods such as Platanios et al. (2019) on both translation quality and convergence speed. The proposed model is approximately 50% faster than the transformer baseline without CL when compared to the 30% acceleration achieved by Platanios et al. (2019). The authors show that the confidence curve of the proposed curriculum is similar to human learning behaviour (Kruger & Dunning 1999).

3.6 Norm-Based Curriculum Learning for Neural Machine Translation by Liu et al. (2020)

Liu et al. (2020) propose to use the norm of word embeddings as a measure of data difficulty and model competence. They argue that this criterion can capture both linguistic features, such as word frequency and sentence length, and model-based features, such as learning-dependent word significance. This paper proposes a predefined CL with noise as the difficulty measure and a predefined norm-based model competence as the curriculum scheduler.

Norm-based sentence difficulty measures the complexity of a sentence based on the sum of the norms of its word embeddings. The norm of rare words and significant words is high, which makes it an effective indicator of word difficulty. Hence, it combines the benefits of linguistically motivated sentence difficulty (Zhang et al. 2018, Platanios et al. 2019) and model-based sentence difficulty (Zhang et al. 2018, Kumar et al. 2019, Zhou et al. 2020). The norm-based sentence difficulty for a sentence is calculated as the sum of the norm of word vectors of the words in the sentence. Hence, long sentences and sentences containing rare words have high sentence difficulty scores.

Norm-based model competence uses the norm of the source embedding matrix of the NMT model to determine the learning stage of the model and the length of the curriculum. The above method enables a fully automatic curriculum schedule, without the need for a hand-crafted curriculum (Zhang et al. 2018) or task-dependent hyperparameter (Platanios et al. 2019). The training of the vanilla Transformer model shows the curves for the norm of the source embedding matrix and corresponding BLEU scores are similar. Based on this, the authors propose a competency metric that improves over the metric proposed by Platanios et al. (2019), which requires a vanilla baseline model to be trained to determine the value of a hyperparameter T. Hence, the authors propose the following competency metric:

$$\hat{c}(t) = \min(1, \sqrt{(m_t - m_0)\frac{1 - c_0^2}{\lambda_m m_0} + c_0^2}) \tag{2}$$

where λ_m is a task-independent hyperparameter to control the length of the curriculum and m_t is the norm of the source embedding of the NMT model at training step t with m_0 being the initial value.

Competence-based CL tends to over-train on simple sentences, which may cause bias and waste of resources. Norm-based sentence weight assigns a scaling factor to the loss of each sentence based on its ratio of data difficulty to model competence, which helps the NMT model to learn more from sentences whose difficulty is close to the model competence and to balance

the translation performance of simple and medium-difficulty sentences. The authors trained a Transformer (Vaswani et al. 2017) network using the proposed CL strategy along with previous baselines and show that the proposed method outperforms improves performance and reduces training time.

3.7 Reinforced Curriculum Learning on Pre-Trained Neural Machine Translation Models by Zhao et al. (2020)

Zhao et al. (2020) propose a reinforcement learning framework for curriculum learning on pre-trained neural machine translation models. The goal is to improve an existing model by re-selecting a subset of useful samples from the original training set. The framework consists of an actor-network that learns a data selection policy and a critic network that evaluates the action value of choosing each sample. The perplexity difference on a validation set after training the model with the selected example is chosen as the reward for the RL agent.

The research paper presents the task of data re-selection as a deterministic actor-critic issue. In this context, the *state* is defined as the features of a randomly selected batch of samples. The *action* involves selecting one of these samples, and the *reward* is gauged by the enhancement in the model's performance. The paper designs the state features based on three dimensions: informativeness, uncertainty, and diversity. The paper uses sentence length, sentence-level log-likelihood, n-gram rarity, and POS and NER taggings as the feature vectors. The paper conducts two rounds of data selection, with different batch sizes and selection criteria, to achieve further improvement.

The paper evaluates the proposed method on several Chinese-to-English translation datasets and compares it with other baseline methods, such as denoising, sentence length, and word rarity. The paper shows that the proposed method significantly outperforms the baseline methods by a large margin, and achieves a consistent performance boost on the pre-trained model. The paper also conducts ablation studies to analyze the impact of different features and rounds of data selection.

3.8 Predefined Difficulty Measurer

Difficulty criteria have been manually designed by researchers based on the data and task in hand. Complexity, diversity and noise are some of the common data characteristics used to measure difficulty. The complexity of text represents the structural complexity and it is measured through features such as sentence length (Zhang et al. 2018), number of coordinating conjunctions (Kocmi & Bojar 2017), number of phrases and parse tree depth. The diversity of data represents the amount of rare sub-entities in the data that the model needs to learn. Word rarity (Platanios et al. 2019) and Part-Of-Speech (POS) entropy are some of the heuristics used to measure diversity in text. Both high complexity and high diversity increase the degrees of freedom of the data. Additionally, noise correlates with diversity of the data and has been used to measure the difficulty (Wang et al. 2020).

Zhang et al. (2018) summarize two types of difficulty criteria in the field of CL for NMT: (1) Linguistic difficulty criteria such as sentence length, and word frequency (Zhang et al. 2018, Platanios et al. 2019); and (2) model-based difficulty criteria that measure difficulty based on auxiliary language model or models trained in previous time steps (Zhang et al. 2018, Kumar et al. 2019, Zhou et al. 2020). While this approach can be intuitively effective, it can also be computationally expensive. Linguistic features are a generic way to model the difficulty of

samples because they are independent of any translation model. While there are many different linguistic difficulty measures, they cannot comprehensively model the difficulty of text. This motivates the need for a model-based difficulty metric.

3.8.1 Complexity

Sentence length is the most commonly used complexity-based difficulty metric to implement CL for NMT since long sentences could have complex syntactic structures and long-range dependencies. Translating longer sentences is inherently more challenging because of the propagation of errors that could occur in the initial stages of generating the target sentence. Zhang et al. (2018) rank training data by the length of the source sentence, the target sentence and the sum of the lengths of both sentences. They found that sentence length did not improve training beyond the early stages for RNN-based NMT systems with CL. While training batches of samples of similar length may improve the computational efficiency (Khomenko et al. 2017), it may not improve the statistical efficiency since smaller sentences containing rare words are difficult to translate compared to long sentences containing common words. Hence, sentence length alone could not act as an optimal difficulty measure and a diversity measure is required to accurately model the data difficulty.

3.8.2 Diversity

Word rarity provides a measure of the diversity of the data and is suitable for NMT. Arranging sentences based on word rarity or word frequency is similar to incrementally expanding the vocabulary size and conducting training on sentences that only include words from the existing subset of the vocabulary (Bengio et al. 2009), which is analogous to human learning. It is hard to translate sentences that contain rare words due to the low number of examples containing those words so word rarity could be a useful heuristic (Platanios et al. 2019). Zhang et al. (2018) ranked samples based on the max or average word frequency rank of the source sentence, target sentence and concatenations of both. They find that their concatenation of word frequency for the translation sample performs comparably to the expensive auxiliary NMT model scores. Platanios et al. (2019) show that the likelihood of the sentence calculated using word frequency is a better difficulty metric than standalone word rarity since it measures both the complexity and diversity of the data by containing information about sentence length and word rarity respectively.

3.8.3 Noise

Data noise is a common problem in web-crawled parallel data and it can negatively affect NMT performance. Since the noise of the data sometimes correlates with the diversity of the data (Wang et al. 2020), noise could be used as a difficulty metric. Wang et al. (2018) define noise to be relative as the difference between a noisy model and a denoised model and show that noise is relative to model competence and changes over training steps. They propose a dynamic data selection strategy to gradually reduce the noise level in data batches to improve the performance of NMT.

3.8.4 Composite difficulty measurer

Metrics such as the norm of the word embedding hold the information about multiple forms of difficulty such as complexity, diversity and noise so the norm is a generic and comprehensive difficulty metric. The norm of the word embedding vector represents the magnitude of the word. Word embeddings are continuous vector representations of words from large datasets that capture syntactic and semantic information about the word (Turian et al. 2010, Mikolov et al. 2013). The norm of the vector captures both the linguistic and model-based difficulty of the sentence. Liu et al. (2020) use norms to measure data difficulty and model competence to design and schedule the curriculum. They find that CL-based NMT models trained with the norm as a difficulty metric outperformed similar models trained with other predefined difficulty measures.

3.9 Predefined Training Scheduler

The predefined training schedulers, unlike predefined difficulty measurers, are data/task agnostic and are classified into *discrete* and *continuous* schedulers. The most popular discrete scheduler is called *Baby Step* (Bengio et al. 2009) which splits the training data into buckets (or shards/bins) of increasing difficulty. However, it introduces the number of bins as a hyperparameter. Training starts with the easiest bucket and after every fixed number of epochs, the next harder bucket is merged with the training data. This scheduler is used by most of the existing CL-based NMT systems. In automatic CL, an RL agent automates the bucket selection based on the data (Kumar et al. 2019). Zhang et al. (2018) find that no single variant of the baby step schedulers consistently outperforms the others. Continuous schedulers use a pacing function or competence function to modify the subset of training data at each epoch by correlating the epoch number with a percentage of the easiest examples available at that epoch. From Table 1, we can observe that Platanios et al. (2019) and Liu et al. (2020) use a continuous scheduler for training CL-based NMT and they show that optimal value of p described in the pacing function of equation (1) is $p >= 2$.

3.10 Automatic CL

Automatic CL has the following major advantages when compared to predefined CL: (1) Difficulty measurer and training schedule of automatic CL is dynamic and can adapt to feedback from the model; and (2) It does not require human expert to design the difficulty metric and curriculum schedule. On the other side, automatic CL requires more training time compared to predefined CL due to additional pre-training or training of RL agents/models. Kumar et al. (2019) used RL agent to automatically learn the curriculum schedule and matched the performance of carefully designed online denoising approach by Wang et al. (2018). Zhao et al. (2020) use an RL agent to re-select a subset of high quality examples from a pre-trained NMT model to maximise the performance.

4 Summary & Conclusion

The paper reviewed existing literature on Curriculum Learning for Neural Machine Translation. Also, the paper analyses the CL approaches based on existing CL theory. A thematic discussion followed by critical analysis of difficulty metrics and curriculum schedulers provide a

comprehensive overview of the current state of CL-based NMT. We observe that CL imrproves state-of-the-art performance for NMT with a high reduction in training time. Zhang et al. (2018) and Platanios et al. (2019) prove that CL improves the convergence time of complex networks such as Transformers (Vaswani et al. 2017) and not simple networks such as RNNs.

Studies of Zhou et al. (2020) and Liu et al. (2020) confirm that the use of composite metrics such as uncertainty and norm that encodes multiple data characteristics about the data such as complexity, diversity and noise. Hence, these composite metrics outperform metrics which model individual data characteristics such as sentence length and word rarity. Thus, they provide the state-of-the-art methods to implement CL for NMT and improved the translation performance and training time of NMT systems. The difficulty metric introduced by Zhou et al. (2020) is promising since it combines the difficulty metric (here, uncertainty) of source and target sentence in an optimal way. A similar approach could be explored in other methods such as norm (Liu et al. 2020). The CL approach of Zhou et al. (2020) could be improved by utilizing sentence weights similar to Liu et al. (2020) to enable models to focus on samples closer to model's competence.

Zhao et al. (2020) shows promise to improve the state-of-the-art performance since it uses RL to imporve the performance of pretrained NMT. However, the approach has to be evaluted in standardized datasets. Kumar et al. (2019) and Zhao et al. (2020) show the ability of RL teacher to learn complex curriculum schedules that maximizes performance. However, such networks are difficult to train due to the additional efforts involved in training RL agents/models. A hybrid approach that utilizes a combination of RL and transfer teacher to optimize the performance and training time simultaneously can be explored in the future. Also, we can observe that the research about automatic CL-based NMT is limited.

An overview of the literature shows the lack of standardized dataset for evaluation of machine translation, making it difficult for researchers to compare the results of various approaches. Further research about different composite difficulty metrics and variations of experimented composite metrics (For instance, norm variants other than the norm of the source embedding) can be performed. Research on automatic CL for NMT with RL teacher that can automatically learn the curriculum schedule based on the student feedback to learn sophisticated data agnostic training schedules can be explored. Amiri et al. (2017) enhanced the convergence speed for neural models by utilizing a technique known as spaced repetition. This approach, rooted in psychological studies, suggests that individuals can increase their learning efficiency by increasing the time gaps between revisiting previously learned content, which could be applied to CL-based NMT methods to improve the performance.

References

Amiri, H., Miller, T. & Savova, G. (2017), Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks, *in* M. Palmer, R. Hwa & S. Riedel, eds, 'Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Copenhagen, Denmark, pp. 2401–2410.
URL: *https://aclanthology.org/D17-1255*

Bahdanau, D., Cho, K. H. & Bengio, Y. (2014), 'Neural machine translation by jointly learning to align and translate', *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* .
URL: *https://arxiv.org/abs/1409.0473v7*

Bengio, Y., Louradour, J., Collobert, R. & Weston, J. (2009), 'Curriculum learning', *ACM International*

Conference Proceeding Series **382**.
URL: *https://dl.acm.org/doi/10.1145/1553374.1553380*

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Parmar, N., Shazeer, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Schuster, M., Chen, Z., Wu, Y. & Hughes, M. (2018), 'The best of both worlds: Combining recent advances in neural machine translation', *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* **1**, 76–86.
URL: *https://arxiv.org/abs/1804.09849v2*

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

Gal, Y. & Ghahramani, Z. (2015), 'Dropout as a bayesian approximation: Representing model uncertainty in deep learning', *33rd International Conference on Machine Learning, ICML 2016* **3**, 1651–1660.
URL: *https://arxiv.org/abs/1506.02142v6*

Jenks, G. & Geography, U. o. K. D. o. (1977), *Optimal Data Classification for Choropleth Maps*, Occasional paper, University of Kansas.
URL: *https://books.google.co.uk/books?id=HvAENQAACAAJ*

Khomenko, V., Shyshkov, O., Radyvonenko, O. & Bokhan, K. (2017), 'Accelerating recurrent neural network training using sequence bucketing and multi-gpu data parallelization', *Proceedings of the 2016 IEEE 1st International Conference on Data Stream Mining and Processing, DSMP 2016* pp. 100–103.
URL: *http://arxiv.org/abs/1708.05604 http://dx.doi.org/10.1109/DSMP.2016.7583516*

Kocmi, T. & Bojar, O. (2017), 'Curriculum learning and minibatch bucketing in neural machine translation', *International Conference Recent Advances in Natural Language Processing, RANLP* **2017-September**, 379–386.
URL: *http://arxiv.org/abs/1707.09533 http://dx.doi.org/10.26615/978-954-452-049-6*$_0$50

Krueger, K. A. & Dayan, P. (2009), 'Flexible shaping: How learning in small steps helps', *Cognition* **110**, 380–394.

Kruger, J. & Dunning, D. (1999), 'Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments.', *Journal of personality and social psychology* **77 6**, 1121–34.
URL: *https://api.semanticscholar.org/CorpusID:2109278*

Kumar, G., Foster, G., Cherry, C. & Krikun, M. (2019), 'Reinforcement learning based curriculum optimization for neural machine translation', *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* **1**, 2054–2061.
URL: *https://aclanthology.org/N19-1208*

Liu, X., Lai, H., Wong, D. F. & Chao, L. S. (2020), 'Norm-based curriculum learning for neural machine translation', *Proceedings of the Annual Meeting of the Association for Computational Linguistics* pp. 427–436.
URL: *https://arxiv.org/abs/2006.02014v1*

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings* .
URL: *https://arxiv.org/abs/1301.3781v3*

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. & Hassabis, D. (2015), 'Human-level control through deep

reinforcement learning', *Nature* **518**, 529–533.
URL: *https://api.semanticscholar.org/CorpusID:205242740*

Ornstein, A. & Hunkins, F. (2017), *Curriculum: Foundations, Principles, and Issues, Global Edition*, Pearson Higher Education & Professional Group.
URL: *https://books.google.co.uk/books?id=D8OfjwEACAAJ*

Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. (2001), 'Bleu', *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02* p. 311.
URL: *https://dl.acm.org/doi/10.3115/1073083.1073135*

Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B. & Mitchell, T. M. (2019), 'Competence-based curriculum learning for neural machine translation', *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* **1**, 1162–1172.
URL: *https://arxiv.org/abs/1903.09848v2*

Popel, M. & Bojar, O. (2018), 'Training tips for the transformer model', *The Prague Bulletin of Mathematical Linguistics* **110**, 43–70.
URL: *http://arxiv.org/abs/1804.00247 http://dx.doi.org/10.2478/pralin-2018-0002*

Sutskever, I., Vinyals, O. & Le, Q. V. (2014), 'Sequence to sequence learning with neural networks', *Advances in Neural Information Processing Systems* **4**, 3104–3112.
URL: *https://arxiv.org/abs/1409.3215v3*

Turian, J., Ratinov, L. & Bengio, Y. (2010), 'Word representations: A simple and general method for semi-supervised learning'.
URL: *https://aclanthology.org/P10-1040*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Łukasz Kaiser & Polosukhin, I. (2017), 'Attention is all you need', *Advances in Neural Information Processing Systems* **2017-December**, 5999–6009.
URL: *https://arxiv.org/abs/1706.03762v7*

Wang, W., Watanabe, T., Hughes, M., Nakagawa, T. & Chelba, C. (2018), 'Denoising neural machine translation training with trusted data and online data selection', *WMT 2018 - 3rd Conference on Machine Translation, Proceedings of the Conference* **1**, 133–143.
URL: *https://arxiv.org/abs/1809.00068v1*

Wang, X., Chen, Y. & Zhu, W. (2020), 'A survey on curriculum learning', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 4555–4576.
URL: *https://arxiv.org/abs/2010.13166v2*

Zhang, X., Kumar, G., Khayrallah, H., Murray, K., Gwinnup, J., Martindale, M. J., McNamee, P., Duh, K. & Carpuat, M. (2018), 'An empirical exploration of curriculum learning for neural machine translation'.
URL: *https://arxiv.org/abs/1811.00739v1*

Zhao, M., Wu, H., Niu, D. & Wang, X. (2020), 'Reinforced curriculum learning on pre-trained neural machine translation models', *ArXiv* **abs/2004.05757**.
URL: *https://api.semanticscholar.org/CorpusID:211102338*

Zhou, Y., Yang, B., Wong, D. F., Wan, Y. & Chao, L. S. (2020), 'Uncertainty-aware curriculum learning for neural machine translation', *Proceedings of the Annual Meeting of the Association for Computational Linguistics* pp. 6934–6944.
URL: *https://aclanthology.org/2020.acl-main.620*