Uncertainty-Aware Curriculum Learning for Neural Machine Translation

Yikai Zhou[†] **Baosong Yang**[‡] **Derek F. Wong**[†*] **Yu Wan**[†] **Lidia S. Chao**[†]
[†]NLP[2]CT Lab, Department of Computer and Information Science, University of Macau
nlp2ct.{yzhou,ywan}@gmail.com, {derekfw,lidiasc}@um.edu.mo
[‡]Alibaba DAMO Academy
yangbaosong.ybs@alibaba-inc.com

Abstract

Neural machine translation (NMT) has proven to be facilitated by curriculum learning which presents examples in an easy-to-hard order at different training stages. The keys lie in the assessment of data difficulty and model competence. We propose *uncertainty-aware curriculum learning*, which is motivated by the intuition that: 1) the higher the uncertainty in a translation pair, the more complex and rarer the information it contains; and 2) the end of the decline in model uncertainty indicates the completeness of current training stage. Specifically, we serve cross-entropy of an example as its data difficulty and exploit the variance of distributions over the weights of the network to present the model uncertainty. Extensive experiments on various translation tasks reveal that our approach outperforms the strong baseline and related methods on both translation quality and convergence speed. Quantitative analyses reveal that the proposed strategy offers NMT the ability to automatically govern its learning schedule.

1 Introduction

Neural machine translation (NMT) has advanced the state-of-the-art on various translation tasks (Hassan et al., 2018; Chen et al., 2018). A well-performed NMT is trained using an end-to-end framework (Sutskever et al., 2014) that profits from large-scale training corpus and various optimization tricks (Ott et al., 2018; Xu et al., 2019; Li et al., 2020). These techniques boost the translation quality, in the meanwhile, leading to massive hyper-parameters to be tuned and expensive development costs (Popel and Bojar, 2018). Recent studies (Zhang et al., 2018, 2019; Platanios et al., 2019; Liu et al., 2020) have proven that feeding training examples in a meaningful order rather than considering them randomly can accelerate the model convergence thus reducing the computational cost. Such methods refer to curriculum learning (CL, Bengio et al., 2009), in which a model is taught as a human from simple concepts to complex ones.

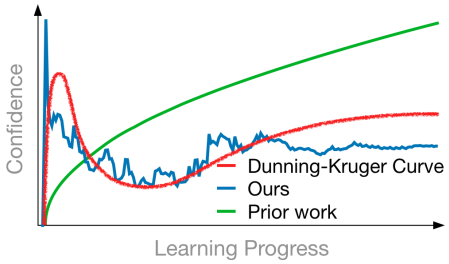

Figure 1: The change of confidence in an area during the learning. Humans (red) experience the process of overconfidence⇒despair⇒enlightenment (Dunning-Kruger Curve), while prior work that exploits CL in NMT assumes a monotonically increased curve (green, Platanios et al., 2019). Interestingly, our model automatically draws a similar tendency as humans (blue).

There exists two open problems in the integration of CL with NMT, i.e. the assessment of data difficulty and the programme of learning schedule. Considering the former, prior studies (Kocmi and Bojar, 2017; Platanios et al., 2019) intuitively treat human linguistic knowledge, e.g. either sentence length or word rarity, as the measure of difficulty. Nevertheless, each linguistic feature merely considers an aspect of sentences which fails to fully cope with the data difficulty for a model (Jiang et al., 2015). For the latter, existing methods pre-define the duration of curriculum based on an assumption that the model confidence monotonically increases with the training (Zhang et al., 2018, 2019). We argue that this assumption does not conform to human behavior, i.e. Dunning-Kruger Curve (Figure 1, Kruger and Dunning, 1999), and limits the adaptability and flexibility of curriculum learning.

In response to these problems, we propose to strengthen CL for NMT through determining the

*Corresponding author

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6934–6944
July 5 - 10, 2020.

data difficulty and scheduling the curriculum according to model ability rather than human intuitions. We introduce a novel *uncertainty-aware curriculum learning* framework, which serves uncertainty as its principle to order the input examples and control the duration of each training stage. Specifically, we measure the data uncertainty of a sentence pair according to its joint distribution that is estimated by a language model pre-trained on the training corpus. The intuition behind is that the higher the cross-entropy and uncertainty have in an example, the harder it is to learn and translate (Brown et al., 1990). Besides, we calculate the model uncertainty using the variance of the distribution over the network presented by Bayesian neural networks (Buntine and Weigend, 1991). Accordingly, the model uncertainty reflects whether our model can best describe the data distribution (Xiao and Wang, 2019), and the stop of its decline indicates the completeness of the current training stage.

One principle in our work is to maintain the simplicity and efficiency in CL. Several researchers may doubt that the use of Bayesian inference over the training corpus may significantly raise the computational cost. To this end, we apply Monte Carlo Dropout (Gal and Ghahramani, 2016) to approximate Bayesian inference. Besides, we categorize examples into subsets according to their difficulty, which is then be progressively added into the training set at different training stages, namely baby step (Cirik et al., 2016). The model uncertainty can be calculated after each epoch using the samples randomly selected from the current training set, thus avoiding affect training efficiency.

We evaluate the effectiveness of our methods on WMT16 English-to-German, IWSLT15 English-to-Vietnamese, and WMT17 Chinese-to-English translation tasks. The experimental results demonstrate that the proposed model consistently improves translation performance over the strong TRANSFORMER (Vaswani et al., 2017) baseline and related methods that exploit CL into NMT. Extensive analyses confirm that: 1) our approach significantly speeds up the model convergence; 2) using data uncertainty to present the translation difficulty surpasses its sentence length and word rarity counterparts, and this superiority can be further expanded by exploiting a language model that is trained on large-scale external data, i.e. BERT (Devlin et al., 2019); 3) the model uncertainty performs a self-adaptive manner to assess the model competence regardless the pre-defined patterns.

2 Preliminary

NMT uses a single, large neural network to build translation model, aiming to maximize the conditional distribution of sentence pairs using parallel corpus (Sutskever et al., 2014; Bahdanau et al., 2015; Yang et al., 2019; Wan et al., 2020). Formally, the learning objective is to minimize the following loss function over the training corpus $\mathcal{D} = \{(x^n, y^n)\}_{n=1}^{N}$, with the size being N:

$$\mathcal{L} = \mathbb{E}_{(x^n,y^n)\sim\mathcal{D}}[-\log \mathbf{P}(y^n|x^n;\theta)] \quad (1)$$

where x^n and y^n indicate the source and target sides of the n-th example in training data. θ denotes the trainable parameters of NMT model. During the training, the examples randomly feed to vanilla model, regardless of their order, making the development of a well-performed NMT system time-consuming (Sennrich et al., 2016a; Popel and Bojar, 2018; Yang et al., 2020). An alternative way to speed up the training process and boost the performance of a neural network is to exploit CL (Elman, 1993; Krueger and Dayan, 2009; Bengio et al., 2009).

Related Work on Exploring CL Several studies have shown the effectiveness of CL in the field of computer vision (Sarafianos et al., 2017; Wang et al., 2019c; Guo et al., 2018), as well as a range of NLP tasks, including math word problem (Zaremba and Sutskever, 2014), sentiment analysis (Cirik et al., 2016), and natural answer generation (Liu et al., 2018). They point out that CL can solve the problem in some tasks that is hard to train through presenting training data in an easy-to-hard order.

Kocmi and Bojar (2017) first apply CL into NMT and suggest two sticking points, i.e. data difficulty and learning schedule. Partially inspired by their findings, Thompson et al. (2018), Zhang et al. (2019), Wang et al. (2019b) and Kumar et al. succeed on handling the problem in domain translation. Concerning the general translation tasks, Zhang et al. (2018) investigate a variety of difficulty criteria based on human intuition, e.g. sentence length and word rarity, which show distinct performance across language pairs and model settings. While Platanios et al. (2019) pay attention to the schedule that determines the duration of each curriculum. They introduce monotonically increased curves, e.g. either linear or square root, to represent the

changes of the model ability across the training process. These early successes presuppose the limited heuristic knowledge on both the data difficulty and the tendency of model competence.

3 Methodology

Motivation As mentioned above, one of the main challenges in CL is the identification of easy and hard samples which is onerous and conceptually difficult in translation community. For example, neither the sentence length or word rarity can fully express the complexity of a translation. Another problem in CL is the programme of learning schedule, in which the patterns pre-defined by humans lack in adaptability and lead to massive additional hyper-parameters that have to be tuned. Even if these artificial supervisions are feasible, what is intuitively "easy" and "competent" for a human may not match that for neural networks (Kumar et al., 2010; Jiang et al., 2015).

To this end, we approach these problems from the model perspective. In this section, we first introduce data uncertainty to quantify the translation difficulty for each training example (Section 3.1). Then, we propose to predict the model uncertainty at the training time which is a self-adaptive manner to govern curriculum by the model itself (Section 3.2). Finally, we describe how to exploit the proposed two factors in NMT training (Section 3.3). The proposed framework is illustrated in Figure 2.

3.1 Data Uncertainty

In order to estimate the data uncertainty, we propose to pre-train a language model (LM) over the monolingual sentences from the parallel training corpus $\mathcal{D}$ to account the cross-entropy of each sentence. The intuition behind this is that the higher cross-entropy and perplexity represents an uncertain sentence, since it is hard to be generated and determined by the LM (Brown et al., 1990). This provides an explainable and comprehensive way to evaluate the difficulty of an example. Accordingly, we assign several types of data uncertainty, which can be used individually or combined together:

Source Difficulty The difficulty of a source sentence affects the language understanding of NMT model. Inspired by Zhang et al. (2018) and Platanios et al. (2019), an interpretable way is to use the source difficulty to approximate the complexity of a sentence pair. Given the source sentence x^n, we can calculate the source uncertainty $u^{data}(x^n)$ by

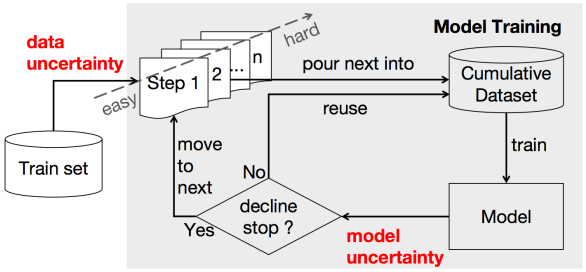

Figure 2: Illustration of the proposed *uncertainty-aware curriculum learning* framework. We categorize training corpus into baby steps according to their data uncertainty. The sign of entering the next curriculum is the stop of decline in model uncertainty which is estimated over random samples in the current training stage.

estimating its perplexity, namely:

$$u^{data}(x^n) = -\frac{1}{I}\sum_{n=1}^{N}\log \mathbf{P}(x_i^n|x_{<i}^n) \quad (2)$$

where I indicates the length of source sentence.

Target Difficulty Since the complex and rare target sentence directly makes NMT have a harder time in generating the sentence (Kocmi and Bojar, 2017), another natural choice is to apply the target uncertainty to present the data difficulty. Analogous to the source side, the target uncertainty $u^{data}(y^n)$ is:

$$u^{data}(y^n) = -\frac{1}{J}\sum_{j=1}^{J}\log \mathbf{P}(y_i^n|y_{<i}^n) \quad (3)$$

where J denotes the length of target sentence y^n.

Joint Difficulty Intuitively, the complexity of a translation pair should be contributed by two sides, thus reflecting the difficulty of both understanding and generating processes in NMT. We can combine the concepts of source and target uncertainty:

$$u^{data}(x^n, y^n) = u^{data}(x^n) + u^{data}(y^n) \quad (4)$$

To our best knowledge, due to the lack of interpretability on scoring the joint difficulty in a sentence pair, all the existing methods that exploit CL into NMT merely measure data difficulty on either source or target. Our method provides an alternative way to tackle this problem with the concept of joint probability distribution. We expect the joint uncertainty to further improve the performance.

In this paper, we examine three widely used LMs to appraise the data uncertainty: a statistical n-gram LM – KENLM (Heafield, 2011), a neural LM – RNNLM (Mikolov et al., 2010), and a multilingual neural LM that trained on billions of external sentences – BERT (Devlin et al., 2019). Note that, the modeling of data uncertainty is not limited to our approach. It can be also quantified by other manners, e.g. estimating the data likelihood with Monte Carlo approximation (Der Kiureghian and Ditlevsen, 2009) or validating the translation distribution using a well-trained NMT model (Zhang et al., 2018). In contrast to these time-consuming techniques, LM marginally increases the computational cost and easy to be applied, conforming to the original motivation of CL.

Algorithm 1: Uncertainty-Aware CL

Input: Train set $\mathcal{D} = \{(x^n, y^n)\}_{n=1}^{N}$.

Compute the data uncertainty u^{data} for each sentence pair in $\mathcal{D}$ (Section 3.1).
Split $\mathcal{D}$ into T baby steps according to u^{data} in ascending order, $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$.
Initialize cumulative dataset $\mathcal{C} = \mathcal{D}_1$.
for *training epoch* $e = 1, \ldots$ **do**
    Train NMT model θ using $\mathcal{C}$.
    $\mathcal{D}^U \leftarrow$ Sample M examples from $\mathcal{C}$.
    Calculate the model uncertainty $u^{mod}(\theta)$ on $\mathcal{D}^U$ (Section 3.2).
    if u^{mod} *stop decline* **then**
        $\mathcal{C} \leftarrow$ Pull next baby step $\mathcal{D}_{next}$ into $\mathcal{C}$.
        Use $\mathcal{C}$ for next epoch training.

3.2 Model Uncertainty

Moreover, we propose to regulate the duration of each curriculum by quantifying the model uncertainty rather than presetting before the training. Model uncertainty, which is also known as epistemic uncertainty (Kendall and Gal, 2017), can be used to measure whether the model parameters are able to best describe the data distribution (Dong et al., 2018; Xiao and Wang, 2019). In our work, a small score of model uncertainty indicates the model is confident that the current training data has been well learned (Wang et al., 2019a), and the termination of the decline in scores represents the signal to shift to the next curriculum stage.

The model uncertainty can be quantified by Bayesian neural networks (Buntine and Weigend, 1991; Neal, 1996), which place a probabilistic distribution over the model parameters on constant input and output data, and serve its variance as the uncertainty. For reasons of computational efficiency, we adopt widely used Monte Carlo Dropout (Gal and Ghahramani, 2016) to approximate Bayesian inference. Given a dataset used to examine the model uncertainty $\mathcal{D}^U = \{(x^m, y^m)\}_{m=1}^{M}$ which consists of M sentence pairs, we perform K passes of forward propagation through the NMT model.[1] In each pass, part of neurons in network θ are randomly deactivated. Eventually, we yield K samples on model parameters $\{\hat{\theta}^1, \cdots, \hat{\theta}^K\}$ and corresponding translation probabilities. The model uncertainty on $\mathcal{D}^U$ can be formally expressed as:

$$u^{mod}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{Var}\left[\mathbf{P}(y^m|x^m, \hat{\theta}^k)\right]_{k=1}^{K} \quad (5)$$

Here, $\mathbf{Var}[\cdot]$ denotes the variance of a distribution which calculated following the common setting in Dong et al. (2018) and Xiao and Wang (2019). In this way, the model is offered the ability to determine its model competence by itself.

3.3 Self-Adaptive Training Strategy

In this work, we adopt a widely used CL strategy called baby step (Cirik et al., 2016; Zhang et al., 2018) to arrange training data and organize the training process. Specifically, the whole training set $\mathcal{D}$ is divided into different buckets, i.e. steps $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$, in which those examples with similar data uncertainty scores u^{data} are categorized into the same bucket. The training starts from the step that consists of examples with the lowest uncertainty. After that, data in the next step is aggregated to the current training dataset $\mathcal{C}$ when the model uncertainty ceases its reduction. Following existing studies (Platanios et al., 2019; Kocmi and Bojar, 2017) that the model should be trained from easy samples to hard ones, we schedule the curriculum with the order of increasing uncertainty.[2] To avoid overfitting and useless training, partially inspired by early stopping, we treat the third time when current model uncertainty is higher than the score

[1] K is empirically set to 10 in our work.

[2] Our preliminary experiments show that the model with a reverse order does not gain any performance improvement to the baseline model.

evaluated last time as the sign that the model is at the level of "expert" for the current curriculum. The hyperparameter of stopping criterion is important. A small value makes the training to easily enter the next baby step, and the current baby step fails to be fully trained, while a large value reduces training efficiency and cause over-fitting.

Considering that performing Monte Carlo Dropout over the NMT model on all the examples in $\mathcal{C}$ is time-consuming, while the superiority of CL lies in its ability to accelerate the model convergence. In order to maintain this advantage, we propose to estimate the model uncertainty after each epoch rather than every model updating steps. Furthermore, we randomly extract $M = 1$k samples from current training dataset $\mathcal{C}$ as $\mathcal{D}^U$. Then, the evaluation of model uncertainty is conducted on $\mathcal{D}^U$ to mirror the confidence over the current curriculum. Therefore, our approach reserves the efficiency in CL, in the meanwhile, guiding the duration of each curriculum in a self-adaptive fashion. The overall procedure is described in Algorithm 1.

4 Experiments

We examine our method upon advanced TRANSFORMER (Vaswani et al., 2017) and conduct experiments on widely used translation tasks: IWSLT15 English-to-Vietnamese (En⇒Vi), WMT16 English-to-German (En⇒De) and WMT17 Chinese-to-English (Zh⇒En).[3]

4.1 Experimental Setting

Dataset To compare with the results reported by previous work (Platanios et al., 2019), we evaluate our methods on IWSLT15 En⇒Vi and WMT16 En⇒De translation tasks. Our models are trained using all of the available parallel corpora from the IWSLT15 and WMT16 datasets, consisting of 133k and 4.5M sentence pairs. In order to verify the universality of the proposed method, we also conduct experiments on the large-scale training corpus, i.e. WMT17 Zh⇒En, in which, 20M examples are extracted as the training set. We use the standard validation and test sets provided in each translation task. The Chinese sentences are segmented by the word segmentation toolkit *Jieba*,[4] while the sentences in other languages are tokenized using the scripts provided in Moses.[5] All the data are processed by byte-pair encoding to alleviate the Out-of-Vocabulary problem (Sennrich et al., 2016b) with 32K merge operations for both language pairs. The case-sensitive 4-gram NIST BLEU score (Papineni et al., 2002) is used as the evaluation metric.

Model Our experiments are based on TRANSFORMER (Vaswani et al., 2017) and the compared methods are re-implemented on top of our in-house codes. Considering the small-scale translation task En⇒Vi, we use the setting same as Platanios et al. (2019) in which the dropout ratio is set to 0.3 and each iteration batch consists of 4,096 tokens. For translation models on En⇒De and Zh⇒En, we follow the common *Base* setting in Vaswani et al. (2017) except that we set dropout ratio to 0.1 and train models with a total batch of 32,768 tokens. As to LMs, we train 4-gram KENLM (Heafield, 2011)[6] and 2 layers RNNLM (Mikolov et al., 2010) with dimensionality being 200 on monolingual side of each training corpus. Besides, we also score sentences using multilingual BERT (Devlin et al., 2019) that pre-trained on external data with *Base* setting for comparison.

We investigate the following methods:

- LENGTH measures data difficulty with sentence length (Kocmi and Bojar, 2017).
- RARITY measures data difficulty with word rarity (Zhang et al., 2018).
- DATA-U represents the proposed method which measures difficulty with data uncertainty on source sentence (src), target sentence (trg), and both sides (joint).
- SQRT governs curriculum with the square root model competence (Platanios et al., 2019).
- MOD-U governs curriculum with the proposed model uncertainty. In our experiments, we set baby steps to 4 as default.

4.2 Ablation Study

In this section, we evaluate the effectiveness of different components in CL on the En⇒De task. In the first two series of experiments, we investigate the effects of different measures of data difficulty and model competence. Then, we check how the baby steps applied in our training strategy affect the performance. The results are concluded in Table 1.

[3] Our code is available at https://github.com/NLP2CT/ua-cl-nmt

[4] https://github.com/fxshy/jieba

[5] https://github.com/mosesdecoder

[6] https://github.com/kpu/kenlm

Model		SQRT	MOD-U
TRANSFORMER		32.76	
LENGTH		32.80	$33.23^{\uparrow}$
RARITY		32.84	$33.39^{\uparrow}$
DATA-U	KENLM (src)	33.03	$33.64^{\Uparrow}$
	KENLM (trg)	33.09	$33.69^{\uparrow}$
	KENLM (joint)	33.15	$\mathbf{33.85}^{\Uparrow}$
	RNNLM (joint)	33.17	$33.73^{\uparrow}$
	BERT (joint)	33.35*	$33.93^{\uparrow}$*

Table 1: Ablation study of various measures with respect to data difficulty and model competence for CL in NMT. The results are evaluated using BLEU on En⇒De translation task, where * indicates that the result is produced with a LM trained on external data. "$^{\uparrow}$ / $^{\Uparrow}$" indicates statistically significant difference from the SQRT counterpart ($p < 0.05/0.01$), tested by bootstrap resampling (Koehn, 2004).

Effectiveness of Data Uncertainty We first compare different difficulty measures in CL. Considering the existing methods, both the LENGTH and RARITY yield improvements over the baseline model, which is consistent with prior findings in Kocmi and Bojar (2017), Zhang et al. (2018) and Platanios et al. (2019). The proposed data uncertainty strategies outperform the baseline and existing measures. This verifies our hypothesis that data uncertainty is of higher relevance in respect to the difficulty of an example for a NMT model than its sentence length and word rarity counterparts.

Specifically, the results show the utility of estimating the uncertainty on either the source or target side of a translation pair. Among the two strategies, the target one performs better. We attribute this to the fact that the target uncertainty brings a more direct reflex of the sentence generation difficulty, thus playing a crucial role in CL. Moreover, "joint", which provides a more comprehensive way to model data uncertainty, achieves the best results. This success indicates that the two strategies are complementary to each other and the complexity of a translation pair is contributed by both sides.

We attempt three kinds of LMs to quantify data uncertainty. As seen, all the models contribute to the model performance. Concerning LMs trained on the monolingual side of a parallel corpus, KENLM and RNNLM get comparable translation qualities. Besides, as a state-of-the-art LM, BERT has recently attracted a lot of interests since it learns from billions of external sentences. As expected, it outperforms all the LMs trained on internal data. Although this comparison is unfair, the results suggest that the performance of LM significantly affects the evaluation of data uncertainty. Since the statistical approach can be faster developed and it does not rely on external data, we choose KENLM as the default in the subsequent experiments.

Effectiveness of Model Uncertainty In this experiment, we evaluate the impacts of different assessments on model competence. Obviously, our approach "MOD-U" consistently gains improvements over the vanilla method "SQRT" with the same setting. These results reveal that applying model uncertainty to determine the duration of each curriculum by the model itself is conductive to CL in NMT. Moreover, the combination of data uncertainty and model uncertainty can progressively boost the model performance, confirming that the two methods are complementary to each other.

Different Baby Steps We further explore the effects of the number of baby steps on En⇒ De translation task. The experiments are conducted on the proposed uncertainty-aware CL model as plotted in Figure 3. The vanilla NMT system without using any curriculum strategy could be considered as the model that sets the total number of steps to 1. As seen, dividing training corpus into 4 baby steps is superior to other settings. Before that, the translation performance increases with progressively subdividing baby steps, since the model with fine-grained steps can benefit more from CL. When the total number of subsets is greater than 4, the tendency of translation qualities decreases. A plausible explanation is that to train the model on an over-small subset leads to the problem of overfitting.

4.3 Main Results

In this section, we evaluate the proposed approach on both IWSLT15 En⇒Vi, WMT16 En⇒De, as well as WMT17 Zh⇒En tasks, as listed in Table 2. Our baseline TRANSFORMER and re-implemented existing methods outperform the reported results in Platanios et al. (2019), which we believe makes the evaluation convincing. As seen, the proposed *uncertainty-aware curriculum learning* strategy consistently outperforms strong baselines and recent studies that exploit CL into NMT across language pairs. These results demonstrate the universality and effectiveness of the proposed approach.

Model	WMT16 En⇒De		IWSLT15 En⇒Vi		WMT17 Zh⇒En	
Baseline & Related Methods						
TRANSFORMER	32.76	-	30.01	-	24.19	-
+SQRT+LENGTH	32.80	+0.04	29.83	-0.18	24.17	-0.02
+SQRT+RARITY	32.84	+0.08	30.10	+0.09	24.31	+0.12
The Proposed Models						
Uncertainty-Aware	33.85⇑	+1.09	30.75⇑	+0.74	25.04⇑	+0.85
Uncertainty-Aware with BERT	33.93⇑	+1.17	30.94⇑	+0.93	25.02⇑	+0.83

Table 2: Comparing with baseline and existing methods that exploit CL on IWSLT15 En⇒Vi, WMT16 En⇒De, as well as WMT17 Zh⇒En translation tasks. The evaluation metric is BLEU.

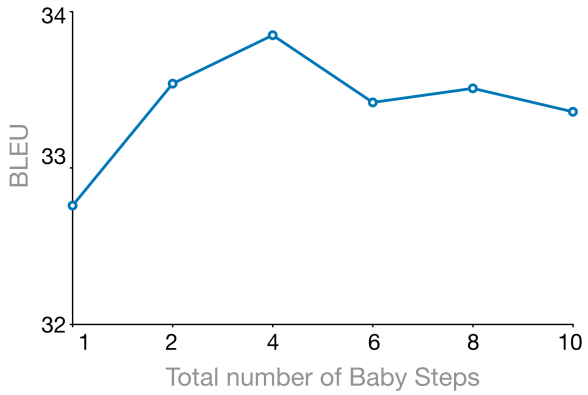

Figure 3: Evaluation of our models trained with different total number of baby steps, where the number of baby step being 1 represents the vanilla NMT system. The experiments are conducted on En⇒De task.

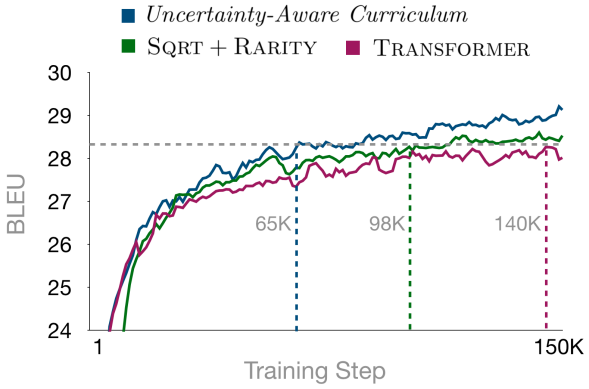

Figure 4: Convergence curves of different models on En⇒De development set. Obviously, our model is able to achieve the same performance as baseline with the reduction in update steps of 53.6%.

It is encouraging to see that the improvement does not diminish but enlarges with the increase of training data, indicating that the model is conducive to the large scale translation tasks. Interestingly, our model with BERT is superior to that with KENLM trained on small scale data, while the gap becomes minor when KENLM learns from a larger training corpus (e.g. 20M Zh⇒En task). We attribute this to the fact that, with the use of the large-scale training examples, KENLM can describe its data distribution well, and the superiority of BERT tends to marginal in these tasks.

5 Analysis

We conduct extensive analyses on En⇒De task to better understand our model. We investigate three problems: 1) whether the proposed model indeed speeds up the model convergence; 2) how different are between difficulty measures; and 3) how the model uncertainty exactly changes during training.

5.1 Model Convergence

As aforementioned, one intuition of CL is to speed up the model convergence. Figure 4 shows the learning curves of different models on En⇒De validation set. As seen, the conventional NMT model reaches the highest BLEU at 140k steps, while related CL method SQRT+RARITY obtains the same performance at step 98k, which achieves 30% accelerate rate. The acceleration effect is slightly asthenic than that reported in Platanios et al. (2019). This could be explained by the fact that their examined models are trained with a batch of 5,120 tokens, which is much smaller than 32,768 used in our experiments. The large batch facilitates the training (Popel and Bojar, 2018), thus weakening the acceleration effect. In spite of that, our model converges 53.6% faster than the baseline to get the same BLEU score (step 65k), showing the action of the proposed method on speeding up the training.

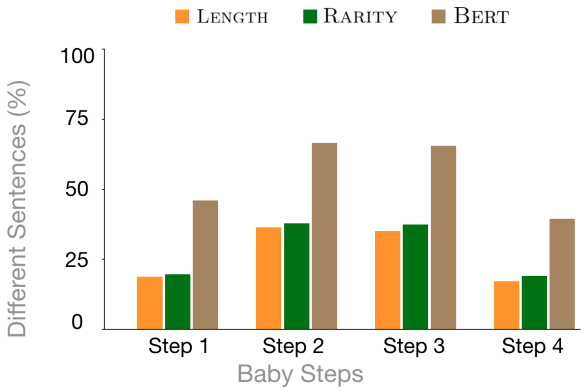

Figure 5: Statistics on the percentage of difference set between the corresponding baby steps that produced by our model (KENLM) and others. As seen, there exist obvious diversities among these methods.

5.2 Difference among Difficulty Measures

It is interesting to investigate the discrepancy among data difficulty measures. Accordingly, we compare the composition of the corresponding baby steps sorted by different difficulty methods. Figure 5 shows the percentage of distinct sentence contained in each subset of our method (KENLM) to that in others (LENGTH, RARITY, and BERT). As seen, there exists considerable diversity among associated baby steps produced by our method and existing approaches. Moreover, the difference in the middle period of curriculums, i.e. step 2 and step 3, is greater than that in step 1 and step 4. This phenomenon reveals that the most "simple" and "complex" sentences quantified by different measures are relatively similar, and the main diversity lies in those sentences of which the difficulties hardly to be distinguished. Therefore, we argue that the improvements of the proposed method may mainly contribute by the differences in these two steps. Besides, the subsets divided by KENLM and BERT have big gaps, which suggest again that the performance of LM plays a crucial role in our approach.

5.3 Variety in Model Uncertainty

In this section, we discuss the training process from the model uncertainty perspective. For better illustration, we define the model confidence as the reciprocal of model uncertainty ($1/u^{mod}$), since the two features are negative correlation (Dong et al., 2018; Wang et al., 2019a). Figure 6 visualizes the curves concerning the average of model confidence on En⇒De validation set during the curriculum learning. We analyze those models trained on two baby steps divided by different data difficulty measures, i.e. KENLM, BERT, and RARITY, for comparison.

Obviously, different models draw similar changing trends of model confidence during training, that is, the model confidence first increases sharply, then drops and rises, eventually balances. Surprisingly, the tendency highly accords with the psychology of human students when they getting into a new area, i.e. Dunning Kruger Curve (Figure 1, Kruger and Dunning, 1999). That is, starting from scratch, peoples rapidly grow their knowledge, they therefore have a large amount of confidence. Then, peoples begin to have awareness about how little they really know and are discouraged by their inability. Over time, humans gradually improve, making them more and more confident, and experienced. To some extent, both the artificial neural networks and human beings can be regarded as connectionist models (Munakata and McClelland, 2003). Accordingly, this interpretation can be also used to explain the phenomenon in NMT training. Such kind of fluctuates model confidence confirms that the curriculum duration should not be fixed, and the predefined strategies may be insufficient to cope with the model training. In addition, the models trained in different curriculums with various difficulty measures perform distinct change amplitudes on model uncertainty, indicating the adaptability of our method. These findings support our assumption that the model uncertainty is an effective and self-adaptive indicator to guide the CL.

6 Conclusion and Future Work

We propose a novel *uncertainty-aware* framework to improve the two key components in CL for NMT, i.e. data difficulty measurement and curriculum arrangement. Our contributions are mainly in:

- We propose to estimate the data uncertainty of each training example as its difficulty, which is more explainable and comprehensive.
- We introduce a self-adaptive CL strategy that evaluates the model uncertainty to govern the curriculum by the model itself.
- The extensive experiments on various translation tasks and model settings demonstrate the universal-effectiveness of the proposed framework. Our method is able to achieve over 50% accelerate rate on model convergence.
- Quantitative and qualitative analyses indicate

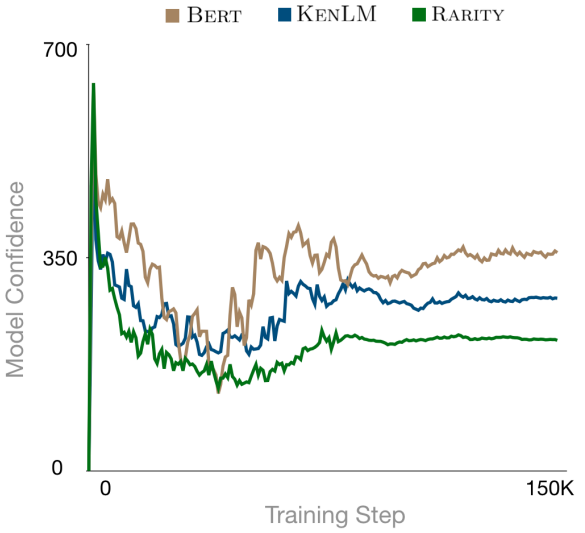

Figure 6: Curves of model confidence ($1/u^{mod}$) on En⇒De validation set at different checkpoints. We evaluate the model uncertainties of CL models that exploit different data difficulty measures. It is clear to see that different methods have the same change trend of model confidence but distinct change amplitudes.

that the model confidence is fluctuant at the training time. It surprisingly draws a similar changing curve as human confidence.

As our model is not limited to machine translation, it is interesting to validate the proposed framework into other NLP tasks that need to exploit CL. Another promising direction is to design more powerful training strategies to replace the baby step.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61672555), the Joint Project of the Science and Technology Development Fund, Macau SAR and National Natural Science Foundation of China (Grant No. 045/2017/AFJ), the Science and Technology Development Fund, Macau SAR (Grant No. 0101/2019/A2), and the Multi-year Research Grant from the University of Macau (Grant No. MYRG2017-00087-FST). We thank the anonymous reviewers for their insightful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *ICML*.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

Wray L Buntine and Andreas S Weigend. 1991. Bayesian Back-Propagation. *Complex systems*, 5(6):603–643.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. In *ACL*.

Volkan Cirik, Eduard H. Hovy, and Louis-Philippe Morency. 2016. Visualizing and Understanding Curriculum Learning for Long Short-Term Memory Networks. *CoRR*, abs/1611.06204.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or Epistemic? Does It Matter? *Structural Safety*, 31(2):105–112.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence Modeling for Neural Semantic Parsing. In *ACL*.

Jeffrey L Elman. 1993. Learning and Development in Neural Networks: The Importance of Starting Small. *Cognition*, 48(1):71–99.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *ICML*.

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 *ECCV*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. *CoRR*, abs/1803.05567.

Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *EMNLP*.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-Paced Curriculum Learning. In *AAAI*.

Alex Kendall and Yarin Gal. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *NIPS*.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum Learning and Minibatch Bucketing in Neural Machine Translation. In *RANLP*.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *EMNLP*.

Kai A Krueger and Peter Dayan. 2009. Flexible Shaping: How Learning in Small Steps Helps. *Cognition*, 110(3):380–394.

Justin Kruger and David Dunning. 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.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. Reinforcement Learning Based Curriculum Optimization for Neural Machine Translation. In *NAACL*.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-Paced Learning for Latent Variable Models. In *NIPS*.

Jian Li, Xing Wang, Baosong Yang, Shuming Shi, Michael R Lyu, and Zhaopeng Tu. 2020. Neuron Interaction Based Representation Composition for Neural Machine Translation. In *AAAI*.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum Learning for Natural Answer Generation. In *IJCAI*.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *ACL 2020*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent Neural Network based Language Model. In *ISCA*.

Yuko Munakata and James L. McClelland. 2003. Connectionist Models of Development. *Developmental Science*, 6(4):413–429.

Radford M. Neal. 1996. *Bayesian Learning for Neural Networks*. Springer-Verlag.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-Based Curriculum Learning for Neural Machine Translation. In *NAACL*.

Martin Popel and Ondřej Bojar. 2018. Training Tips for The Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

N. Sarafianos, T. Giannakopoulos, C. Nikou, and I. A. Kakadiaris. 2017. Curriculum Learning for Multi-task Classification of Visual Attributes. In *ICCV(Workshops)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*.

Brian Thompson, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson, and Philipp Koehn. 2018. Freezing Subnetworks to Analyze Domain Adaptation in Neural Machine Translation. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NIPS*.

Yu Wan, Baosong Yang, Derek F. Wong, Lidia S. Chao, Haihua Du, and Ben CH Ao. 2020. Unsupervised Neural Dialect Translation with Commonality and Diversity Modeling. In *AAAI*.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019a. Improving Back-Translation with Uncertainty-Based Confidence Estimation. In *EMNLP*.

Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019b. Dynamically Composing Domain-Data Selection with Clean-Data Selection by "Co-Curricular Learning" for Neural Machine Translation. In *ACL*.

Yiru Wang, Weihao Gan, Wei Wu, and Junjie Yan. 2019c. Dynamic Curriculum Learning for Imbalanced Data Classification. *CoRR*, abs/1901.06783.

Yijun Xiao and William Yang Wang. 2019. Quantifying Uncertainties in Natural Language Processing Tasks. In *AAAI*.

Mingzhou Xu, Derek F. Wong, Baosong Yang, Yue Zhang, and Lidia S. Chao. 2019. Leveraging Local and Global Patterns for Self-Attention Networks. In *ACL*.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019. Assessing the Ability of Self-Attention Networks to Learn Word Order. In *ACL*.

Baosong Yang, Derek F. Wong, Lidia S. Chao, and Min Zhang. 2020. Improving Tree-based Neural Machine Translation with Dynamic Lexicalized Dependency Encoding. *Knowledge-Based System*, 188.

Wojciech Zaremba and Ilya Sutskever. 2014. Learning to Execute. *CoRR*, abs/1410.4615.

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. *CoRR*, abs/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 *NAACL*.