Self-Paced Learning for Neural Machine Translation

Yu Wan[†‡]   **Baosong Yang**[‡*]   **Derek F. Wong**[†*]   **Yikai Zhou**[†]
Lidia S. Chao[†]   **Haibo Zhang**[‡]   **Boxing Chen**[‡]
[†]NLP²CT Lab, University of Macau
`nlp2ct.{ywan,yzhou}@gmail.com, {derekfw,lidiasc}@umac.mo`
[‡]Alibaba Group
`{yangbaosong.ybs,zhanhui.zhb,boxing.cbx}@alibaba-inc.com`

Abstract

Recent studies have proven that the training of neural machine translation (NMT) can be facilitated by mimicking the learning process of humans. Nevertheless, achievements of such kind of curriculum learning rely on the quality of artificial schedule drawn up with the hand-crafted features, e.g. sentence length or word rarity. We ameliorate this procedure with a more flexible manner by proposing self-paced learning, where NMT model is allowed to 1) automatically quantify the learning confidence over training examples; and 2) flexibly govern its learning via regulating the loss in each iteration step. Experimental results over multiple translation tasks demonstrate that the proposed model yields better performance than strong baselines and those models trained with human-designed curricula on both translation quality and convergence speed.[1]

1 Introduction

Neural machine translation (NMT) has achieved promising results with the use of various optimization tricks (Hassan et al., 2018; Chen et al., 2018; Xu et al., 2019; Li et al., 2020; Yang et al., 2020). In spite of that, these techniques lead to increased training time and massive hyper-parameters, making the development of a well-performed system expensive (Popel and Bojar, 2018; Ott et al., 2018).

As an alternative mitigation, curriculum learning (CL, Elman, 1993; Bengio et al., 2009) has shown its effectiveness on speeding up the convergence and stabilizing the NMT model training (Zhang et al., 2018; Platanios et al., 2019). CL teaches NMT model from easy examples to complex ones rather than equally considering all samples, where the keys lie in the definition of "diffi-

culty" and the strategy of curricula design (Krueger and Dayan, 2009; Kocmi and Bojar, 2017). Existing studies artificially determine data difficulty according to prior linguistic knowledge such as sentence length (SL) and word rarity (WR) (Platanios et al., 2019; Zhang et al., 2019; Zhou et al., 2020), and manually tune the learning schedule (Liu et al., 2020; Fomicheva et al., 2020). However, neither there exists a clear distinction between easy and hard examples (Kumar et al., 2010), nor these human intuitions exactly conform to effective model training (Zhang et al., 2018).

Instead, we resolve this problem by introducing self-paced learning (Kumar et al., 2010), where the emphasis of learning can be dynamically determined by model itself rather than human intuitions. Specifically, our model measures the level of confidence on each training example (Gal and Ghahramani, 2016; Xiao and Wang, 2019), where an easy sample is actually the one of high confidence by the current trained model. Then, the confidence score is served as a factor to weight the loss of its corresponding example. In this way, the training process can be dynamically guided by model itself, refraining from human predefined patterns.

We evaluate our proposed method on IWSLT15 En⇒Vi, WMT14 En⇒De, as well as WMT17 Zh⇒En translation tasks. Experimental results reveal that our approach consistently yields better translation quality and faster convergence speed than TRANSFORMER (Vaswani et al., 2017) baseline and recent models that exploit CL (Platanios et al., 2019). Quantitative analyses further confirm that the intuitive curriculum schedule for a human does not fully cope with that for model learning.

2 Self-Paced Learning for NMT

As mentioned above, translation difficulty for humans may not match that for neural networks. Even
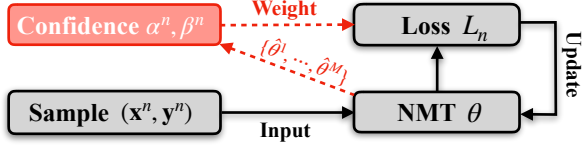

Figure 1: Illustration of the proposed self-paced learning. The black components compose the vanilla NMT training process, while our model (red) assigns confidence scores for each input to weight its loss.

if these artificial supervisions are feasible, the long sequences or rare tokens are not always "difficult" as the model competence increases. From this view, we design a self-paced learning algorithm that offers NMT the abilities to 1) estimate the confidences over samples appropriated for the current training state; and 2) automatically control the focus of learning through regulating the training loss, as illustrated in Fig. 1.

2.1 Confidence Estimation

We propose to determine the learning emphasis according to the model confidence (Ueffing and Ney, 2005; Soricut and Echihabi, 2010), which quantifies whether the current model is confident or hesitant on translating the training samples. The model confidence can be quantified by Bayesian neural networks (Buntine and Weigend, 1991; Neal, 1996), which place distributions over the weights of network. For efficiency, we adopt widely used Monte Carlo dropout sampling (Gal and Ghahramani, 2016) to approximate Bayesian inference.

Given current NMT model parameterized by θ and a mini-batch consisting of N sentence pairs $\{(\mathbf{x}^1, \mathbf{y}^1), \cdots, (\mathbf{x}^N, \mathbf{y}^N)\}$, we first perform M passes through the network, where the m-th pass $\hat{\theta}^m$ randomly deactivates part of neurons. Thus, each example yields M sets of conditional probabilities. The lower variance of translation probabilities reflects higher confidence that the model has with respect to the instance (Dong et al., 2018; Wang et al., 2019). We propose multi-granularity strategies for confidence estimation:

Sentence-Level Confidence (SLC) A natural choice for measuring the confidence of sentence pair $(\mathbf{x}^n, \mathbf{y}^n)$ is to assess the variance of translation probability $\mathrm{Var}\{P(\mathbf{y}^n|\mathbf{x}^n, \hat{\theta}^m)\}_{m=1}^M$. Accordingly, confidence score $\hat{\alpha}^n$ can be formally expressed as:

$$\hat{\alpha}^n = (1 - \mathrm{Var}\{P(\mathbf{y}^n|\mathbf{x}^n, \hat{\theta}^m)\}_{m=1}^M)^k, \quad (1)$$

Here, we assign a hyper-parameter k to scale the gap between scores of confident and unconfident

examples. The larger absolute value of k represents higher discriminative manner and vice versa. In some extreme cases, all the confidence scores in a mini-batch may tend to small or big value, e.g. the estimation at the early stage of the training.[2] In order to stabilize the training process and maintain the same loss scale as conventional model, we normalize the confidence scores by $softmax$:

$$\alpha^n = \frac{\exp(\hat{\alpha}^n)}{\sum_{t=1}^N \exp(\hat{\alpha}^t)}. \quad (2)$$

Token-Level Confidence (TLC) Intuitively, confidence scores can be evaluated at more fine-grained level. We extend our model into token-level so as to estimate the confidence on translating each element in target sentence $\mathbf{y}^n$. The confidence $\hat{\beta}_j^n$ of the j-th token $\mathbf{y}_j^n$ is:

$$\hat{\beta}_j^n = (1 - \mathrm{Var}\{P(\mathbf{y}_j^n|\mathbf{x}^n, \mathbf{y}_{<j}^n, \hat{\theta}^m)\}_{m=1}^M)^k, \quad (3)$$

where $\mathrm{Var}\{P(\mathbf{y}_j^n|\mathbf{x}^n, \mathbf{y}_{<j}^n, \hat{\theta}^m)$ denotes the variance of the translation probability with respect to $\mathbf{y}_j^n$. Similar to sentence-level strategy, the confidence scores of tokens are normalized as:

$$\beta_j^n = \frac{\exp(\hat{\beta}_j^n)}{\sum_{t=1}^J \exp(\hat{\beta}_t^n)}, \quad (4)$$

where J indicates the length of target sentence $\mathbf{y}^n$.

2.2 Training Strategy

A larger confidence score indicates that the current model is confident on the corresponding example. Therefore, the model should learn more from the predicted loss. In order to govern the learning schedule automatically, we leverage the confidence scores as factors to weight the loss, thus controlling the update at each time step. To this end, the sentence log-likelihood can be defined as:

$$\mathcal{L}_n = \sum_{j=1}^J \beta_j^n \log P(\mathbf{y}_j^n|\mathbf{x}^n, \mathbf{y}_{<j}^n, \theta), \quad (5)$$

Finally, the loss of a batch is calculated as:

$$\mathcal{L} = \sum_{n=1}^N \alpha^n \mathcal{L}_n. \quad (6)$$

---

[2]When implementing the computation of SLC&TLC scores, we use negative log-likelihood values instead of conventional probabilities. Besides, we use the maximum value to refactorize them within $[0, 1]$ by division.

At the early stage of the study, the model learns more from confident samples, thus accelerating the training. The hesitant samples are not completely ignorant, but relatively few can be learned. As training proceeds, the loss of high-confidence samples gradually reduce, and the model will pay more attention on "complex" samples with low prediction accuracy, thus raising their confidence. In this way, the loss of different samples are dynamically revised, eventually balancing the learning.

Contrast to related studies (Zhang et al., 2018, 2019; Kumar et al., 2019; Platanios et al., 2019) which adopt CL into NMT with predefined patterns, the superiority of our model lies in its flexibility on both learning emphasis and strategy. Several researchers may concern about the processing speed when integrating Monte Carlo Dropout sampling. Contrary to prior studies which estimate confidence during inference (Dong et al., 2018; Wang et al., 2019), we only perform forward propagation $M = 5$ times in training time, which avoids the auto-regressive decoding for efficiency.

3 Experiments

We evaluate our method upon TRANSFORMER-*Base*/*Big* model (Vaswani et al., 2017) and conduct experiments on IWSLT15 English-to-Vietnamese (En⇒Vi), WMT14 English-to-German (En⇒De) and WMT17 Chinese-to-English (Zh⇒En) tasks. For fair comparison, we use the same experimental setting as Platanios et al. (2019) for En⇒Vi and follow the common configuration in Vaswani et al. (2017) for En⇒De and Zh⇒En.

During training, we apply 0.3 dropout ratio and batch size as 4,096 for En⇒Vi task, and experiments are conducted upon one Nvidia GTX1080Ti GPU device. For En⇒De and Zh⇒En task, we use 32,768 as batch size, and use four Nvidia V100 GPU devices for experiments. We use beam size as 4, 5, 10, and decoding alpha as 1.5, 0.6, 1.35 for each task, respectively (Vaswani et al., 2017). We compare our models with two baselines:

- *Base* and *Big* represent the vanilla TRANSFORMER (Vaswani et al., 2017) models.

- +CL is the recent NMT model that exploits CL (Platanios et al., 2019). Difficulty of each training sample is estimated according to its sentence length (SL) or averaged word rarity (WR). The curriculum schedule depends on the number of training step.
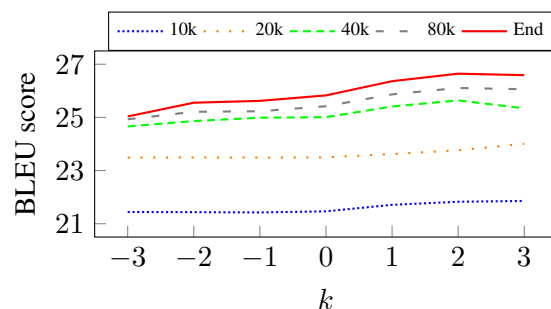

Figure 2: Affects of k on best performance after certain training steps upon En⇒De dev set. At the early stage of the training, a higher k yields better translation quality, denoting a faster convergence speed.

3.1 Confidence/Unconfidence Balancing

As mentioned in Sec. 2.1, we assign k to balance the extent of discrimination between confident and unconfident examples. We first conduct experiments on En⇒De to evaluate the impact of k. Fig. 2 shows that, the larger k, the faster convergence speed. However, the final performance slightly decreases when $k > 2$. We believe that the overlarge k leads to overfit on confident samples and ignore initial hesitated samples. This demonstrates that an appropriate balance on the discriminative manner contributes to both convergence acceleration and final performance.

Besides, when k is negative, models will pay more attention to unconfident examples. This circumstance is identical to reverse-CL (Zhang et al., 2019), where training is advised to offer examples in a hard-to-easy order. Our results confirm that unconfidence-first strategy ($k < 0$) performs worse than baseline, which is similar with previous findings on CL (Zhang et al., 2018). We attribute this to the fact that the heuristic design forces NMT model to unceasingly learn more from unconfident examples, and finally leads to the strait of catastrophic forgetting (Goodfellow et al., 2014). Therefore, we set $k = 2$ for subsequent experiments.

3.2 Main Results

As shown in Tab. 1, our baseline models outperform the reported results (Vaswani et al., 2017; Platanios et al., 2019) on the same data, making the evaluation convincing. The proposed self-paced learning method (SPL) achieves better results than existing CL approaches that artificially determine the difficulty (SL or WR), demonstrating the effectiveness of our method. Specifically, removing either SLC or TLC decreases the translation quality,

Model	IWSLT15 En⇒Vi	WMT14 En⇒De	WMT17 Zh⇒En	Acc.
TRANSFORMER-*Base*	30.05 ± 0.14	27.90 ± 0.24	24.11 ± 0.10	-
+CL-SL	29.91 ± 0.13	27.99 ± 0.22	24.10 ± 0.08	1.02
+CL-WR	30.05 ± 0.17	28.02 ± 0.24	24.25 ± 0.09	1.06
SPL	$\mathbf{31.21 \pm 0.15}\uparrow$	$\mathbf{28.87 \pm 0.19}\uparrow$	$\mathbf{24.86 \pm 0.12}\uparrow$	1.46
w/o TLC	$30.91 \pm 0.17\uparrow$	$28.51 \pm 0.21\uparrow$	$24.62 \pm 0.12\uparrow$	1.17
w/o SLC	$31.14 \pm 0.14\uparrow$	$28.73 \pm 0.24\uparrow$	$24.79 \pm 0.10\uparrow$	1.28
TRANSFORMER-*Big*	30.61 ± 0.12	28.72 ± 0.23	24.57 ± 0.14	-
SPL	$\mathbf{31.45 \pm 0.15}\uparrow$	$\mathbf{29.68 \pm 0.25}\uparrow$	$\mathbf{25.26 \pm 0.15}\uparrow$	1.21

Table 1: Overall experimental results of all approaches upon three translation tasks. Each cell contains the mean value and standard variance of BLEU scores derived from 5 independent experimental runs. "SPL": proposed self-paced learning model. "Acc.": Acceleration ratio of training time required to achieve the best performance of baseline. "↑": the improvement is significant by contrast to TRANSFORMER-*Base/Big* baseline model ($p < 0.01$).

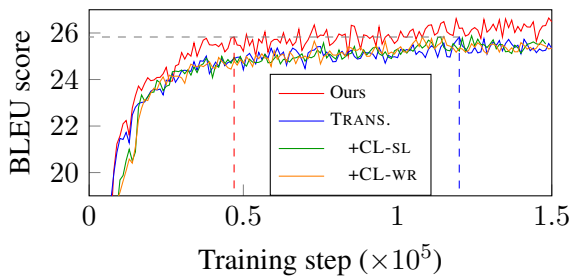Figure 3: Learning curves of models on validation set. Our model achieves reductions in iterations of 2.43x.
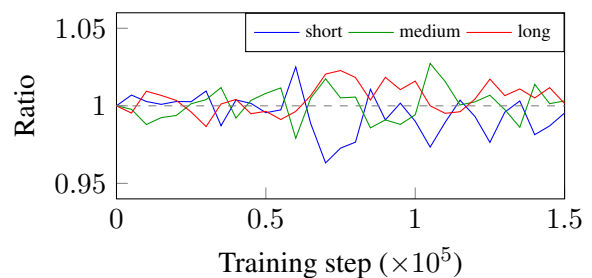

Figure 4: The ratios between averaged SLC scores gained by our model and baseline. Obviously, the model confidence on training samples with different lengths change constantly during model training.

indicating that two confidence estimations are complementary to each other. TLC outperforms its SLC counterpart, which confirms that more fine-grained information benefits to the training. Moreover, our method consistently improves the translation quality with around 1 BLEU score across all involved tasks and multiple model settings. This shows the universality and effectiveness of SPL on different scales of training data and model sizes.

4 Analysis

In this section, we further investigate how the proposed method exactly affects the NMT model training by conducting experimental analyses on 1) convergence speed, 2) self-paced adjustment and 3) sequential bucketing.

4.1 Convergence Speed

As aforementioned, one motivation of exploiting self-paced learning is to accelerate the convergence of model training. We visualize the learning curve of examined models on En⇒De dev set in Fig. 3. As seen, the vanilla NMT model reaches its convergence at step 120k, while the proposed one gets the same performance at step 47k, yielding 2.43

times faster. Although Monte Carlo Dropout sampling requires extra time to forward-pass the neural network at each iteration step, our method can still reach comparable result on dev set with shorter training time, achieving 1.46x faster training speed (column "Acc." in Tab. 1). Besides, we also observe that two methods proposed by Platanios et al. (2019) reveal a comparable tendency with baseline. We explain this with the view that Platanios et al. (2019) examined these approaches with a batch of 5,120 tokens, much smaller than that used in our experiments (32,768). Since larger batch size can considerably facilitate the training (Popel and Bojar, 2018; Ott et al., 2018), the benefits of their models may be marginal with this change.

4.2 Self-Paced Adjustment

It is interesting to investigate how our model adjusts its learning. We randomly extract 300 En⇒De training examples, which then be categorized into 3 subsets according to their sentence lengths. Fig. 4 shows the ratios of averaged SLC scores between our method and vanilla NMT system at different

checkpoints. As seen, at the beginning of the training, the ratio of confidence score with respect to short sentences is greater than 1, indicating our model pays more attention to shorter examples than baseline. This is identical with human intuition that the short sentences seem easier and should be learned earlier (Zhang et al., 2019; Zhao et al., 2020). However, as training continues, our model focuses on short and long sentences simultaneously and hesitates on sentences with medium length, which goes against human intuition and indicates that long sentences may easier than its medium counterparts for current model. From then on, the curves fluctuate and interlace continuously, revealing that SPL automatically regulates its learning emphasis. These phenomena show the flexibility of our model, and confirm that predefined data difficulty and learning schedule is insufficient to fully match the model learning.

4.3 Sequential Bucketing

Conventional model training sorts examples with similar lengths into buckets to keep efficiency. This may introduce bias when estimating confidence scores, because longer sequence may gain far less attention due to the productive multiplication of probabilities for SLC estimation. Generally, larger window size for bucketing increases the diversity of length within each batch, but reduce the efficiency of training due to extra padding tokens.

To investigate whether the diversity of sequential lengths within each batch may introduce bias to SLC score computation, we conduct a series of experiments with different settings of sequential bucketing. As shown in Fig.5, we explore the effect of this on En⇒De task, revealing that both baseline and our approach can gain improvement from larger bucket range. Nevertheless, the performance of baseline model decreases along with lower diversity of sequential lengths, whereas that of our model does not diminish. Our model gives better performance with smaller window size compared to baseline. Here we can conclude, that the performance of TRANSFORMER baseline model is bothered by close sequence lengths within each batch, whereas our model shows its flexibility of adjusting its learning to avoid such effect.

For fair comparison as well as keeping the training efficiency, we follow the default setting from Vaswani et al. (2017) by determining 20 as the number of buckets across all experiments.
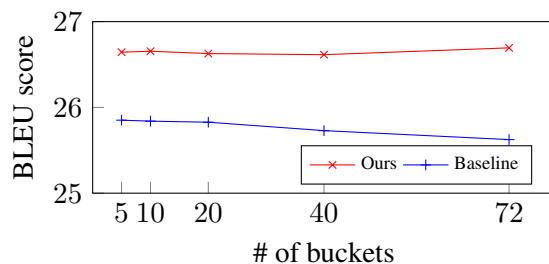

Figure 5: Performance upon WMT14 En⇒De dev set with different bucketing strategy. With window size for sequence bucketing being smaller, the number of buckets accordingly increases, and our model can maintain its performance whereas baseline drops.

5 Conclusion

In this paper, we propose a novel self-paced learning model for NMT in which the learning schedule is determined by model itself rather than being intuitively predefined by humans. Experimental results on three translation tasks verify the universal effectiveness of our approach. Quantitative analyses confirm that exploiting self-paced strategy presents a more flexible way to facilitate the model convergence than its CL counterparts. It is interesting to combine with other techniques (Li et al., 2018; Hao et al., 2019) to further improve NMT. Besides, as this idea is not limited to machine translation, it is also interesting to validate our model in other NLP tasks, such as low-resource NMT model training (Lample et al., 2018; Wan et al., 2020) and neural architecture search (Guo et al., 2020).

References

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

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, Goerge Foster, Llion Jones, Parmar Niki, Mike Schuster, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. In *ACL*.

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.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised Quality Estimation for Neural Machine Translation. *arXiv preprint arXiv:2005.10608*.

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

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks. In *ICLR*.

Yong Guo, Yaofo Chen, Yin Zheng, Peilin Zhao, Jian Chen, Junzhou Huang, and Mingkui Tan. 2020. Breaking the Curse of Space Explosion: Towards Efficient NAS with Curriculum Search. In *ICLR*.

Jie Hao, Xing Wang, Baosong Yang, Longyue Wang, Jinfeng Zhang, and Zhaopeng Tu. 2019. Modeling Recurrence for Transformer. In *NAACL*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. *arXiv preprint arXiv:1803.05567*.

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

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

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement Learning based Curriculum Optimization for Neural Machine Translation. In *NAACL:HLT*.

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

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-Based & Neural Unsupervised Machine Translation. In *EMNLP*.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-Head Attention with Disagreement Regularization. In *EMNLP*.

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*.

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

Radford M Neal. 1996. *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media.

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

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

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

Radu Soricut and Abdessamad Echihabi. 2010. Trustrank: Inducing Trust in Automatic Translations via Ranking. In *ACL*.

Nicola Ueffing and Hermann Ney. 2005. Word-Level Confidence Estimation for Machine Translation using Phrase-Based Translation Models. In *HLT/EMNLP*.

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 C. H. Ao. 2020. Unsupervised Neural Dialect Translation with Commonality and Diversity Modeling. In *AAAI*.

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

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, 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.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An Empirical Exploration of Curriculum Learning for Neural Machine Translation. *arXiv preprint arXiv:1811.00739*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum Learning for Domain Adaptation in Neural Machine Translation. In *NAACL:HLT*.

Mingjun Zhao, Haijiang Wu, Di Niu, and Xiaoli Wang. 2020. Reinforced Curriculum Learning on Pretrained Neural Machine Translation Models. In *AAAI*.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-Aware Curriculum Learning for Neural Machine Translation. In *ACL*.