Norm-Based Curriculum Learning for Neural Machine Translation

Xuebo Liu[1*] **Houtim Lai**[2*] **Derek F. Wong**[1†] **Lidia S. Chao**[1]
[1]NLP[2]CT Lab, Department of Computer and Information Science, University of Macau
[2]NewTranx Information Technology, Shenzhen, China
nlp2ct.xuebo@gmail.com, haotian.li@newtranx.com,
{derekfw,lidiasc}@um.edu.mo

Abstract

A neural machine translation (NMT) system is expensive to train, especially with high-resource settings. As the NMT architectures become deeper and wider, this issue gets worse and worse. In this paper, we aim to improve the efficiency of training an NMT by introducing a novel *norm-based curriculum learning* method. We use the norm (aka length or module) of a word embedding as a measure of 1) the difficulty of the sentence, 2) the competence of the model, and 3) the weight of the sentence. The norm-based sentence difficulty takes the advantages of both linguistically motivated and model-based sentence difficulties. It is easy to determine and contains learning-dependent features. The norm-based model competence makes NMT learn the curriculum in a fully automated way, while the norm-based sentence weight further enhances the learning of the vector representation of the NMT. Experimental results for the WMT'14 English–German and WMT'17 Chinese–English translation tasks demonstrate that the proposed method outperforms strong baselines in terms of BLEU score (+1.17/+1.56) and training speedup (2.22x/3.33x).

1 Introduction

The past several years have witnessed the rapid development of neural machine translation (NMT) based on an encoder–decoder framework to translate natural languages (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). Since NMT benefits from a massive amount of training data and works in a cross-lingual setting, it becomes much hungrier for training time than other natural language processing (NLP) tasks.

Based on self-attention networks (Parikh et al., 2016; Lin et al., 2017), Transformer (Vaswani et al., 2017) has become the most widely used architecture for NMT. Recent studies on improving Transformer, e.g. deep models equipped with up to 30-layer encoders (Bapna et al., 2018; Wu et al., 2019; Wang et al., 2019; Zhang et al., 2019a), and scaling NMTs which use a huge batch size to train with 128 GPUs (Ott et al., 2018; Edunov et al., 2018), face a challenge to the efficiency of their training. Curriculum learning (CL), which aims to train machine learning models *better* and *faster* (Bengio et al., 2009), is gaining an intuitive appeal to both academic and industrial NMT systems.

The basic idea of CL is to train a model using examples ranging from "easy" to "difficult" in different learning stages, and thus the criterion of difficulty is vital to the selection of examples. Zhang et al. (2018) summarize two kinds of difficulty criteria in CL for NMT: 1) linguistically motivated sentence difficulty, e.g. sentence length, word frequency, and the number of coordinating conjunctions, which is easier to obtain (Kocmi and Bojar, 2017; Platanios et al., 2019); 2) model-based sentence difficulty, e.g. sentence uncertainties derived from independent language models or the models trained in previous time steps or epochs, which tends to be intuitively effective but costly (Zhang et al., 2017; Kumar et al., 2019; Zhang et al., 2019b; Zhou et al., 2020).

In this paper, we propose a novel norm-based criterion for the difficulty of a sentence, which takes advantage of both model-based and linguistically motivated difficulty features. We observe that the norms of the word vectors trained on simple neural networks are expressive enough to model the two features, which are easy to obtain while possessing learning-dependent features. For example, most of the frequent words and context-insensitive rare words will have vectors with small norms.

[*]Equal Contribution
[†]Corresponding author

Batch	Len.	Source sentence
Vanilla		
$\mathcal{B}_1$	16	In catalogues, magazines ...
	27	Nevertheless, it is an ...
$\mathcal{B}_2$	38	The company ROBERT ...
	37	Ottmar Hitzfeld played ...
The Proposed Method		
$\mathcal{B}_1^*$	3	Second Part.
	4	It was not.
$\mathcal{B}_2^*$	5	Thank you very much.
	4	We know that.

Table 1: Training batches on the WMT'14 English–German translation task. "Len." denotes the length of the sentence. The proposed method provides a much easier curriculum at the beginning of the training of the model.

Unlike existing CL methods for NMT, relying on a hand-crafted curriculum arrangement (Zhang et al., 2018) or a task-dependent hyperparameter (Platanios et al., 2019), the proposed norm-based model competence enables the model to arrange the curriculum itself according to its ability, which is beneficial to practical NMT systems. We also introduce a novel paradigm to assign levels of difficulty to sentences, as sentence weights, into the objective function for better arrangements of the curricula, enhancing both existing CL systems and the proposed method.

Empirical results for the two widely-used benchmarks show that the proposed method provides a significant performance boost over strong baselines, while also significantly speeding up the training. The proposed method requires slightly changing the data sampling pipeline and the objective function without modifying the overall architecture of NMT, thus no extra parameters are employed.

2 Background

NMT uses a single large neural network to construct a translation model that translates a source sentence $\mathbf{x}$ into a target sentence $\mathbf{y}$. During training, given a parallel corpus $\mathcal{D} = \{\langle \mathbf{x}^n, \mathbf{y}^n \rangle\}_{n=1}^N$, NMT aims to maximize its log-likelihood:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= L(\mathcal{D}; \boldsymbol{\theta}_0) \\ &= \arg\max_{\boldsymbol{\theta}_0} \sum_{n=1}^{N} \log P(\mathbf{y}^n | \mathbf{x}^n; \boldsymbol{\theta}_0) \end{aligned} \quad (1)$$

where $\boldsymbol{\theta}_0$ are the parameters to be optimized during the training of the NMT models. Due to the intractability of N, the training of NMT employs *mini-batch gradient descent* rather than *batch gradient descent* or *stochastic gradient descent*, as follows:

$$\mathcal{B}_1, \cdots, \mathcal{B}_t, \cdots, \mathcal{B}_T = \text{sample}(\mathcal{D}) \quad (2)$$

$$\hat{\boldsymbol{\theta}} = L(\mathcal{B}_T; L(\mathcal{B}_{T-1}; \cdots L(\mathcal{B}_1, \boldsymbol{\theta}_0))) \quad (3)$$

where T denotes the number of training steps and $\mathcal{B}_t$ denotes the tth training batch. In the training of the tth mini-batch, NMT optimizes the parameters $\boldsymbol{\theta}_{t-1}$ updated by the previous mini-batch.

CL supposes that if mini-batches are bucketed in a particular way (e.g. with examples from easy to difficult), this would boost the performance of NMT and speed up the training process as well. That is, upgrading the $\text{sample}(\cdot)$ to

$$\mathcal{B}_1^*, \cdots, \mathcal{B}_t^*, \cdots, \mathcal{B}_T^* = \text{sample}^*(\mathcal{D}) \quad (4)$$

where the order from easy to difficult (i.e. $\mathcal{B}_1^* \rightarrow \mathcal{B}_T^*$) can be: 1) sentences with lengths from short to long; 2) sentences with words whose frequency goes from high to low (i.e. word rarity); and 3) uncertainty of sentences (from low to high uncertainties) measured by models trained in previous epochs or pre-trained language models. Table 1 shows the sentences of the training curricula provided by vanilla Transformer and the proposed method.

3 Norm-based Curriculum Learning

3.1 Norm-based Sentence Difficulty

Most NLP systems have been taking advantage of distributed word embeddings to capture the syntactic and semantic features of a word (Turian et al., 2010; Mikolov et al., 2013). A word embedding (vector) can be divided into two parts: the norm and the direction:

$$\boldsymbol{w} = \underbrace{||\boldsymbol{w}||}_{\text{norm}} \cdot \underbrace{\frac{\boldsymbol{w}}{||\boldsymbol{w}||}}_{\text{direction}} \quad (5)$$

In practice, the word embedding, represented by $\boldsymbol{w}$, is the key component of a neural model (Liu et al., 2019a,b), and the direction $\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$ can also be used to carry out simple word/sentence similarity and relation tasks. However, the norm $||\boldsymbol{w}||$ is rarely considered and explored in the computation.

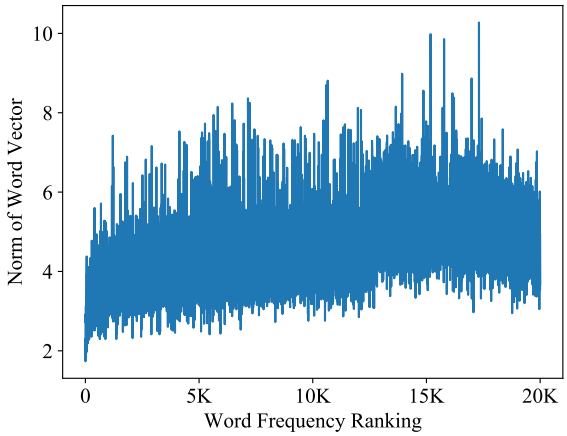


Figure 1: Word vector norm of the word embedding model trained on the WMT'14 English–German (source side) training data. The x-axis is the word frequency, ranked in descending order. Rare words and significant words have higher norms.

Surprisingly, the norm which is simply derived from a single model parameter, can also capture delicate features during the optimization of a model. Schakel and Wilson (2015) observe that in the word embedding model (Mikolov et al., 2013), the word vector norm increases with a decrease of the word frequency, while polysemous words, such as "May", tend to have an average norm weighted over its various contexts. Wilson and Schakel (2015) further conduct controlled experiments on word vector norm and find that besides the word frequency, the diversities of the context of the word are also a core factor to determine its norm. The vector of a context-insensitive word is assigned a higher norm. In other words, if a word is usually found in specific contexts, it should be regarded as a significant word (Luhn, 1958). The word embedding model can exactly assign these significant words higher norms, even if some of them are frequent. The sentences consisting of significant words share fewer commonalities with other sentences, and thus they can also be regarded as difficult-to-learn examples.

Figure 1 shows the relationship between the word vector norm and the word frequency in the English data of the WMT'14 English–German translation task. The results stay consistent with prior works (Wilson and Schakel, 2015), showing that the rare words and significant words obtain a high norm from the word embedding model. Motivated by these works and our preliminary experimental results, we propose to use the word vector norm as a criterion to determine the difficulty of a sentence. Specifically, we first train a simple word embedding model on the training corpus, and then obtain an embedding matrix $\mathbf{E}^{\mathrm{w2v}}$. Given a source sentence $\mathbf{x} = x_1, \cdots, x_i, \cdots, x_I$, it can be mapped into distributed representations $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_i, \cdots, \boldsymbol{x}_I$ through $\mathbf{E}^{\mathrm{w2v}}$. The **norm-based sentence difficulty** is calculated as

$$d(\mathbf{x}) = \sum_{i=1}^{I} ||\boldsymbol{x}_i|| \tag{6}$$

Long sentences and sentences consisting of rare words or significant words tend to have a high sentence difficulty for CL.

The proposed norm-based difficulty criterion has the following advantages: 1) It is easy to compute since the training of a simple word embedding model just need a little time and CPU resources; 2) Linguistically motivated features, such as word frequency and sentence length, can be effectively modeled; 3) Model-based features, such as learning-dependent word significance, can also be efficiently captured.

3.2 Norm-based Model Competence

Besides finding an optimal sentence difficulty criterion, arranging the curriculum in a reasonable order is equally important. As summarized by Zhang et al. (2019b), there are two kinds of CL strategies: deterministic and probabilistic. From their observations, probabilistic strategies are superior to deterministic ones in the field of NMT, benefiting from the randomization during mini-batch training.

Without loss of generality, we evaluate our proposed norm-based sentence difficulty with a typical probabilistic CL framework, that is, competence-based CL (Platanios et al., 2019). In this framework, a notion of model competence is defined which is a function that takes the training step t as input and outputs a competence value from 0 to 1:[1]

$$c(t) \in (0, 1] = \min(1, \sqrt{t\frac{1 - c_0^2}{\lambda_t} + c_0^2}) \tag{7}$$

where $c_0 = 0.01$ is the initial competence at the beginning of training and λ_t is a hyperparameter determining the length of the curriculum. For the sentence difficulty, they use cumulative density function (CDF) to transfer the distribution of sentence difficulties into $(0, 1]$:

$$\hat{d}(\mathbf{x}^n) \in (0, 1] = \mathrm{CDF}(\{d(\mathbf{x}^n)\}_{n=1}^{N})^n \tag{8}$$

[1]We introduce the square root competence model since it has the best performance in Platanios et al. (2019).

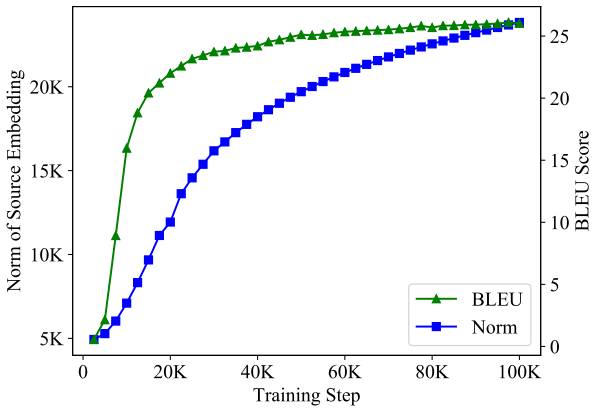


Figure 2: Norm of NMT source embedding and BLEU score of a vanilla Transformer on the WMT'14 English–German translation task. The BLEU scores are calculated on the development set. Both the norm and BLEU score grow rapidly until the 30K training step.

The score of difficult sentences tends to be 1, while that of easy sentences tends to be 0. The model uniformly samples curricula whose difficulty is lower than the model competence at each training step, thus making the model learn the curriculum in a probabilistic way.

One limitation of competence-based CL is that the hyperparameter λ_t is task-dependent. In detail, for each system, it needs to first train a vanilla baseline model and then use the step reaching 90% of its final performance (BLEU score) as the value of the length hyperparameter. As we know, training an NMT baseline is costly, and arbitrarily initializing the value might lead to an unstable training process.

To alleviate this limitation and enable NMT to learn curricula automatically without human interference in setting the hyperparameter, it is necessary to find a way for the model to determine the length of a curriculum by itself, according to its competence, which should be independent of the specific task.

To this aim, we further introduce a **norm-based model competence** criterion. Different from the norm-based difficulty using the word vector norm, the norm-based model competence uses the norm of the source embedding of the NMT model $\mathbf{E}^{\text{nmt}}$:

$$m_t = ||\mathbf{E}_t^{\text{nmt}}|| \tag{9}$$

where m_t denotes the norm of $\mathbf{E}^{\text{nmt}}$ at the tth training step, and we write m_0 for the initial value of the norm of $\mathbf{E}^{\text{nmt}}$. This proposal is motivated by the empirical results shown in Figure 2, where we show the BLEU scores and the norms of the source embedding matrix at each checkpoint of a vanilla Transformer model on the WMT'14 English–German translation task. We found the trend of the growth of the norm m_t to be very similar to that of the BLEU scores. When m_t stays between 15K to 20K, which is about from twice to three times larger than the initial norm m_0, both the growth of the norm and that of the BLEU score have slowed down. It shows strong clues that m_t is a functional metric to evaluate the competence of the model, and thus we can avoid the intractability of λ_t in Equation 7:

$$\hat{c}(t) = \min(1, \sqrt{(m_t - m_0)\frac{1 - c_0^2}{\lambda_m m_0} + c_0^2}) \tag{10}$$

where λ_m is a task-independent hyperparameter to control the length of the curriculum. With this criterion, the models can, by themselves, fully automatically design a curriculum based on the feature (norm). At the beginning of the training, there is a lower m_t, so the models tend to learn with an easy curriculum. But with an increase of the norm m_t, more difficult curricula will be continually added into the learning.

3.3 Norm-based Sentence Weight

In competence-based CL, the model uniformly samples sentences whose difficulty level is under the model competence, and then learns with the samples equally. As a result, those simple sentences with low difficulty (e.g. $\hat{d}(\mathbf{x}) < 0.1$) are likely to be repeatedly used in the model learning. This is somewhat counterintuitive and a waste of computational resources. For example, when students are able to learn linear algebra, they no longer need to review simple addition and subtraction, but can keep the competence during the learning of hard courses. On the other hand, a difficult (long) sentence is usually made up of several easy (short) sentences. Thus, the representations of easy sentences can also benefit from the learning of difficult sentences.

To alleviate this limitation of competence-based CL and further enhance the learning from the curriculum of different levels of difficulty, we propose a simple yet effective **norm-based sentence weight**:

$$w(\mathbf{x}, t) = (\frac{\hat{d}(\mathbf{x})}{\hat{c}(t)})^{\lambda_w} \tag{11}$$

Algorithm 1 Norm-based Curriculum Learning Strategy

Require: Parallel corpus $\mathcal{D} = \{\langle \mathbf{x}^n, \mathbf{y}^n \rangle\}_{n=1}^{N}$; Translation system θ;

```
Train the word2vec Embedding E^{w2v} on {x^n}_{n=1}^N.
Compute norm-based sentence difficulty {d̂(x^n)}_{n=1}^N using E^{w2v}, Eq. 6 and 8.
for t = 1 to T do
    Compute norm-based model competence ĉ(t) using Eq. 9 and 10.
    Generate training batch B_t^* uniformly sampled from {⟨x,y⟩ | d̂(x) < ĉ(t), ⟨x,y⟩ ∈ D}.
    Compute norm-based length weight W = {w(x,t) | ⟨x,y⟩ ∈ B_t^*} using Eq. 11.
    Update θ with batch loss E_{⟨x,y⟩∼B_t^*} calculated by W and Eq. 12.
end for
return θ
```

where λ_w is the scaling hyperparameter smoothing the weight, $\hat{d}(\mathbf{x})$ is the norm-based sentence difficulty, and $\hat{c}(t)$ is the model competence. For each training step t, or each model competence $\hat{c}(t)$, the weight of a training example $w(\mathbf{x}, t)$ is included in its objective function:

$$l(\langle \mathbf{x}, \mathbf{y} \rangle, t) = -\log P(\mathbf{y}|\mathbf{x}) w(\mathbf{x}, t) \qquad (12)$$

where $l(\langle \mathbf{x}, \mathbf{y} \rangle, t)$ is the training loss of an example $\langle \mathbf{x}, \mathbf{y} \rangle$ at the tth training step. With the use of sentence weights, the models, at each training step, tend to learn more from those curricula whose difficulty is close to the current model competence. Moreover, the models still benefit from the randomization of the mini-batches since the length weight does not change the curriculum sampling pipeline.

3.4 Overall Learning Strategy

Algorithm 1 illustrates the overall training flow of the proposed method. Besides the component and training flow of vanilla NMT models, only some low-cost operations, such as matrix multiplication, have been included in the data sampling and objective function, allowing an easy implementation as a practical NMT system. We have also found, empirically, that the training speed of each step is not influenced by the introduction of the proposed method.

4 Experiments

4.1 Data and Setup

We conducted experiments on the widely used benchmarks, i.e. the medium-scale WMT'14 English–German (En-De) and the large-scale WMT'17 Chinese–English (Zh-En) translation tasks. For En-De, the training set consists of 4.5M sentence pairs with 107M English words and 113M German words. The development is newstest13 and the test set is newstest14. For the Zh-En, the training set contains roughly 20M sentence pairs. The development is newsdev2017 and the test set is newstest2017. The Chinese data were segmented by `jieba`,[2] while the others were tokenized by the `tokenize.perl` script from Moses.[3] We filtered the sentence pairs with a source or target length over 200 tokens. Rare words in each data set were split into sub-word units (Sennrich et al., 2016). The BPE models were trained on each language separately with 32K merge operations.

All of the compared and implemented systems are the *base* Transformer (Vaswani et al., 2017) using the open-source toolkit `Marian` (Junczys-Dowmunt et al., 2018).[4] We tie the target input embedding and target output embedding (Press and Wolf, 2017). The Adam (Kingma and Ba, 2015) optimizer has been used to update the model parameters with hyperparameters β_1= 0.9, β_2 = 0.98, ε = 10^{-9}. We use the variable learning rate proposed by Vaswani et al. (2017) with 16K warm up steps and a peak learning rate 0.0003.

We employed `FastText` (Bojanowski et al., 2017)[5] with its default settings to train the word embedding model for calculating the norm-based sentence difficulty; an example is given in Figure 1. The hyperparameters λ_m and λ_w controlling the norm-based model competence and norm-based sentence weight were tuned on the development set of En-De, with the value of 2.5 and 0.5, respectively. To test the adaptability of these two hyperparameters, we use them directly for the Zh-En translation task without any tuning. We compare the proposed methods with the re-implemented

[2] https://github.com/fxsjy/jieba
[3] http://www.statmt.org/moses/
[4] https://marian-nmt.github.io/
[5] https://github.com/facebookresearch/fastText

ID	Model	Dev.	Test	Updates	Speedup
Existing Baselines					
1	GNMT (Wu et al., 2016)	-	24.61	-	-
2	ConvS2S (Gehring et al., 2017)	-	25.16	-	-
3	Base Transformer (Vaswani et al., 2017)	25.80	27.30	-	-
4	Big Transformer (Vaswani et al., 2017)	26.40	28.40	-	-
Our Implemented Baselines					
5	Base Transformer (Vaswani et al., 2017)	25.90	27.64	100.0K	1.00x
6	5 + Competence-based CL (Platanios et al., 2019)	26.39	28.19	60.0K	1.67x
Our Proposed Method (Individual)					
7	6 + Norm-based Model Competence	26.59	28.51	50.0K	2.00x
8	6 + Norm-based Sentence Complexity	26.61	28.61	50.0K	2.00x
9	6 + Norm-based Sentence Weight	26.63	28.32	52.5K	1.90x
Our Proposed Method (All)					
10	5 + Norm-based CL	**26.89**	**28.81**	**45.0K**	**2.22x**

Table 2: Results on the WMT'14 English–German translation task. Dev. is the newstest2013 while Test is newstest2014. 'Updates' means the step of each model reaching the best performance of model (5) (K = thousand), while 'Speedup' means its corresponding speedup.

λ_m	Dev.	λ_w	Dev.
1.0	26.63	0	26.71
2.0	26.72	1/3	26.80
2.5	**26.89**	1/2	**26.89**
3.0	26.65	1	26.78
4.0	26.62	2	26.77

Table 3: Effects of different λ_m of the norm-based model competence function and λ_w of the norm-based sentence weight function.

competence-based CL (Platanios et al., 2019).[6]

During training, the mini-batch contains nearly 32K source tokens and 32K target tokens. We evaluated the models every 2.5K steps, and chose the best performing model for decoding. The maximum training step was set to 100K for En-De and 150K for Zh-En. During testing, we tuned the beam size and length penalty (Wu et al., 2016) on the development data, using a beam size of 6 and a length penalty of 0.6 for En-De, and a beam size of 12 and a length penalty of 1.0 for Zh-En. We report the 4-gram BLEU (Papineni et al., 2002) score given by the *multi-bleu.perl* script. The codes and scripts of the proposed norm-based CL and our re-implemented competence-based CL are freely available at https://github.com/NLP2CT/norm-nmt.

[6]We use its best settings, i.e. the rarity-based sentence difficulty and the square root competence function.

4.2 Main Results

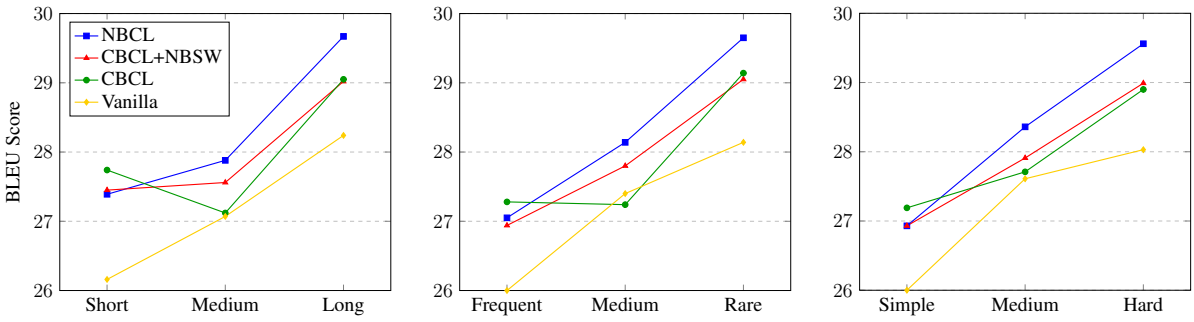
Table 2 shows the results of the En-De translation task in terms of BLEU scores and training speedup. Models (1) to (4) are the existing baselines of this translation benchmark. Model (5) is our implemented base Transformer with 100K training steps, obtaining 27.64 BLEU scores on the test set. By applying the competence-based CL with its proposed sentence rarity and square root competence function, i.e. model (6), it reaches the performance of model (5) using 60K training steps and also gets a better BLEU score.

For the proposed method, we first show the performance of each sub-module, that is: model (7), which uses the norm-based model competence instead of the square root competence of model (6); model (8), which uses the proposed norm-based sentence complexity instead of the sentence rarity of model (6); and model (9), which adds the norm-based sentence weight to model (6). The results show that after applying each sub-module individually, both the BLEU scores and the learning efficiency are further enhanced.

Model (10) shows the results combining the three proposed norm-based methods for CL, i.e. the norm-based sentence difficulty, model competence, and sentence weight. We call the combination of the proposed method norm-based CL. It shows its superiority in the BLEU score, which has an increase of 1.17 BLEU scores compared to the Trans-

ID	Model	Dev.	Test	Updates	Speedup
Existing Baselines					
11	Base Transformer (Ghazvininejad et al., 2019)	-	23.74	-	-
12	Big Transformer (Ghazvininejad et al., 2019)	-	24.65	-	-
Our Implemented Baselines					
13	Base Transformer (Vaswani et al., 2017)	22.29	23.69	150.0K	1.00x
14	13+Competence-based CL (Platanios et al., 2019)	22.75	24.30	60.0K	2.50x
Our Proposed Method					
15	13+Norm-based CL	**23.41**	**25.25**	**45.0K**	**3.33x**

Table 4: Results on the large-scale WMT'17 Chinese–English translation task. Dev. is the newsdev2017 while Test is newstest2017. 'Updates' means the step of each model reaching the best performance of model (13) (K = thousand), while 'Speedup' means its corresponding speedup.



(a) Length (b) Frequency (c) Norm

Figure 3: Translation performance of each NMT system in (a) length-based, (b) frequency-based, and (c) norm-based difficulty buckets. The reported BLEU scores are evaluated on the three subsets evenly divided by the En-De test set based on sentence difficulty. NBCL and CBCL denote norm-based and competence-based CL, respectively. CBCL+NBSW denotes the integration of norm-based sentence weight and competence-based CL.

former baseline, as well as speeding up the training process by a factor of 2.22. One can note that all of our implemented systems have the same number of model parameters; besides, the training step of each model involves essentially the same execution time, resulting in a deployment-friendly system.

4.3 Effect of λ_m and λ_w

Table 3 shows the effects of the two hyperparameters used in the proposed method. For each experiment, we kept the other parameters unchanged and only adjusted the hyperparameter. For λ_m, controlling curriculum length, the higher the value, the longer the curriculum length. When setting λ_m to 2.5 with the curriculum length of nearly 29K steps, it achieves the best performance. For λ_w, the scaling sentence weight of the objective function, one achieves satisfactory results with a value of 0.5, which maintains the right balance between the learning of simple and hard examples.

4.4 Results on the Large-scale NMT

Although the hyperparameters λ_m and λ_w have been sufficiently validated on the En-De translation, the generalizability of the model trained using these two hyperparameters is still doubtful. To clear up any doubts, we further conducted the experiments on the large-scale Zh-En translation without tuning these two hyperparameters, that is, directly using $\lambda_m = 2.5$ and $\lambda_w = 0.5$. Specifically, the only difference is the use of a large number of training steps in Zh-En, namely, 150K, for the purpose of better model fitting.

We first confirm the effectiveness of competence-based CL in large-scale NMT, that is model (14), which shows both a performance boost and a training speedup. Model (15), which trains NMT with the proposed norm-based CL, significantly improves the BLEU score to 25.25 (+1.56) and speeds up the training by a factor of 3.33, showing the generalizability of the proposed method. The results

Source	Last year a team from the University of Lincoln found that dogs turn their heads to the left when looking at an aggressive dog **and to the right when looking at a happy dog**.
Reference	Letztes Jahr fand ein Team der Universität von Lincoln heraus, dass Hunde den Kopf nach links drehen, wenn sie einen aggressiven Hund ansehen, **und nach rechts, wenn es sich um einen zufriedenen Hund handelt**.
Vanilla	Im vergangenen Jahr stellte ein Team der Universität Lincoln fest, dass Hunde beim Blick auf einen aggressiven Hund nach links abbiegen.
NBCL	Letztes Jahr fand ein Team von der Universität von Lincoln heraus, dass Hunde ihren Kopf nach links drehen, wenn sie einen aggressiven Hund sehen **und rechts, wenn sie einen glücklichen Hund sehen**.

Table 5: Example of a translation which is regarded as a difficult sentence in terms of the norm-based sentence difficulty, from the En-De test set. The vanilla Transformer omits translating the **bold** part of the source.

show that large-scale NMT obtains a greater advantage from an orderly curriculum with enhanced representation learning. The proposed norm-based CL enables better and faster training of large-scale NMT systems.

4.5 Effect of Sentence Weight

As discussed in Section 3.3, competence-based CL over-trains on the simple curriculum, which might lead to a bias in the final translation. To verify this, we quantitatively analysed the translations generated by different systems. Figure 3 presents the performance of the vanilla Transformer, and of the NMTs trained by competence-based CL and norm-based CL. By dividing the En-De test set (3,003 sentences) into three subsets (1001 sentences) according to the length-based sentence difficulty, the frequency-based sentence difficulty, and the norm-based sentence difficulty, we calculated the BLEU scores of each system on each subset.

The results confirm our above assumption, although competence-based CL performs much better in translating simple sentences due to its over-training, the translation of sentences of medium difficulty worsens. However, the norm-based CL benefits from the norm-based sentence weight, successfully alleviating this issue by applying a scale factor to the loss of simple curricula in the objective function, leading to a consistently better translation performance over the vanilla Transformer.

To further prove the effectiveness of the proposed norm-based sentence weight, we explore the model integrating norm-based sentence weight with competence-based CL, and find that it can also strike the right balance between translating simple and medium-difficulty sentences.

4.6 A Case Study

Table 5 shows an example of a translation of a difficult sentence consisting of several similar clauses in the norm-based difficulty bucket. We observe that the translation by the vanilla model omits translating the last clause, but NMT with norm-based CL translates the entire sentence. The proposed method enhances the representation learning of NMT, leading to better understandings of difficult sentences, thus yielding better translations.

5 Related Work

The norm of a word embedding has been sufficiently validated to be highly correlated with word frequency. Schakel and Wilson (2015) and Wilson and Schakel (2015) train a simple word embedding model (Mikolov et al., 2013) on a monolingual corpus, and find that the norm of a word vector is relevant to the frequency of the word and its context sensitivity: frequent words and words that are insensitive to context will have word vectors of low norm values.

For language generation tasks, especially NMT, there is still a correlation between word embedding and word frequency. Gong et al. (2018) observe that the word embedding of NMT contains too much frequency information, considering two frequent and rare words that have a similar lexical meaning to be far from each other in terms of vector distance. Gao et al. (2019) regard this issue as a representation degeneration issue that it is hard to learn expressive representations of rare words due to the bias in the objective function. Nguyen and Chiang (2019) observe a similar issue during NMT decoding: given two word candidates with similar lexical meanings, NMT chooses the more frequent one as the final translation. They attribute

this to the norm of word vector, and find that target words with different frequencies have different norms, which affects the NMT score function. In the present paper, for the sake of obtaining an easy and simple word vector norm requirement, we use the norm derived from a simple word embedding model. In the future, we would like to test norms of various sorts.

There are two main avenues for future research regarding CL for NMT: sentence difficulty criteria and curriculum training strategies. Regarding sentence difficulty, there are linguistically motivated features (Kocmi and Bojar, 2017; Platanios et al., 2019) and model-based features (Zhang et al., 2017; Kumar et al., 2019; Zhang et al., 2019b; Zhou et al., 2020). Both types of difficulty criteria have their pros and cons, while the proposed norm-based sentence difficulty takes the best of both worlds by considering simplicity and effectiveness at the same time.

Regarding the training strategy, both deterministic (Zhang et al., 2017; Kocmi and Bojar, 2017) and probabilistic strategies (Platanios et al., 2019; Zhang et al., 2019b; Kumar et al., 2019) can be better than the other, depending on the specific scenario. The former is easier to control and explain, while the latter enables NMT to benefit from the randomization of mini-batch training. However, both kinds of strategy need to carefully tune the CL-related hyperparameters, thus making the training process somewhat costly. In the present paper, we have designed a fully automated training strategy for NMT with the help of vector norms, removing the need for manual setting.

6 Conclusion

We have proposed a novel norm-based curriculum learning method for NMT by: 1) a novel sentence difficulty criterion, consisting of linguistically motivated features and learning-dependent features; 2) a novel model competence criterion enabling a fully automatic learning framework without the need for a task-dependent setting of a feature; and 3) a novel sentence weight, alleviating any bias in the objective function and further improving the representation learning. Empirical results on the medium- and large-scale benchmarks confirm the generalizability and usability of the proposed method, which provides a significant performance boost and training speedup for NMT.

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 2015*.

Ankur Bapna, Mia Xu Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training Deeper Neural Machine Translation Models with Transparent Attention. In *EMNLP 2018*.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. In *TACL 2017*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP 2018*.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *ICLR 2019*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML 2017*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *EMNLP 2019*.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. FRAGE: Frequency-Agnostic Word Representation. In *NIPS 2018*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *ACL 2018*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP 2013*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015*.

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

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *NAACL 2019*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR 2017*.

Xuebo Liu, Derek F. Wong, Lidia S. Chao, and Yang Liu. 2019a. Latent attribute based hierarchical decoder for neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2103–2112.

Xuebo Liu, Derek F. Wong, Yang Liu, Lidia S. Chao, Tong Xiao, and Jingbo Zhu. 2019b. Shared-private bilingual word embeddings for neural machine translation. In *ACL 2019*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR 2013*.

Toan Nguyen and David Chiang. 2019. Improving lexical choice in neural machine translation. In *NAACL-HLT 2019*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *WMT@EMNLP 2018*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL 2002*.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *EMNLP 2016*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *NAACL 2019*.

Ofir Press and Lior Wolf. 2017. Using the Output Embedding to Improve Language Models. In *EACL 2017*.

Adriaan M J Schakel and Benjamin J Wilson. 2015. Measuring Word Significance using Distributed Representations of Words. *arXiv*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL 2016*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS 2014*.

Joseph P Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL 2010*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *ACL 2019*.

Benjamin J Wilson and Adriaan M J Schakel. 2015. Controlled Experiments for Word Embeddings. *arXiv*.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Depth growing for neural machine translation. In *ACL 2019*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019a. Improving deep transformer with depth-scaled initialization and merged attention. In *EMNLP 2019*.

Dakun Zhang, Jungi Kim, Josep Crego, and Jean Senellart. 2017. Boosting Neural Machine Translation. In *IJCNLP 2017*.

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*.

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

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *ACL 2020*.