How does Length Prediction Influence the Performance of Non-Autoregressive Translation?

Minghan Wang[1], Jiaxin Guo[1], Yuxia Wang[2], Yimeng Chen[1], Chang Su[1],
Hengchao Shang[1], Min Zhang[1], Shimin Tao[1], Hao Yang[1]
[1]Huawei Translation Services Center, Beijing, China
[2]University of Melbourne, Melbourne, Australia
{wangminghan,guojiaxin1,chenyimeng,suchang8,
shanghengchao,zhangmin186,taoshimin,yanghao30}@huawei.com
yuxiaw@student.unimelb.edu.au

Abstract

Length prediction is a special task in a series of NAT models where target length has to be determined before generation. However, the performance of length prediction and its influence on translation quality has seldom been discussed. In this paper, we present comprehensive analyses on length prediction task of NAT, aiming to find the factors that influence performance, as well as how it associates with translation quality. We mainly perform experiments based on Conditional Masked Language Model (CMLM) (Ghazvininejad et al., 2019), a representative NAT model, and evaluate it on two language pairs, En-De and En-Ro. We draw two conclusions: 1) The performance of length prediction is mainly influenced by properties of language pairs such as alignment pattern, word order or intrinsic length ratio, and is also affected by the usage of knowledge distilled data. 2) There is a positive correlation between the performance of the length prediction and the BLEU score.

1 Introduction

Though Transformer (Vaswani et al., 2017) has promoted conventional autoregressive generation (AR) by leveraging multi-head self-attention to avoid recurrence at training, decoders that generate each token conditioned on previously generated tokens still make it impossible to take full advantage of parallelism during inference: $p(Y|X) = \prod_i^T p(y_i|y_{\leq i}|X)$. Non-autoregressive Machine Translation (NAT) (Gu et al., 2018; Kaiser et al., 2018) was proposed to parallelize the generation by allowing the prediction of each token independently, known as the conditional independence assumption i.e. $p(Y|X) = \prod_i^T p(y_i|X)$. However, it always results in significant performance degradation. Therefore, how to improve NAT performance remains an open question.

There are basically two directions of NAT models. One is insertion-based models where multiple tokens can be inserted into an existing unfinished sentence, and the sentence length dynamically changes during insertion (Stern et al., 2019; Gu et al., 2019; Chan et al., 2019). Another predetermines the target side length N and generates tokens in a fixed length space conditioned on well-predicted tokens and optionally copied source embedding within N iterations (Gu et al., 2018; Lee et al., 2018; Ghazvininejad et al., 2019). In this paper, we mainly focus on the critical component in the second direction — length prediction which has been rarely discussed. Specifically, we investigate *RQ1: what influences accuracy of length prediction*, and *RQ2: how accuracy of target sentence length prediction affects the translation quality*.

To answer these questions, we first analyze intrinsic properties of sentence lengths for different language pairs. Then we evaluate various length prediction strategies and dig into potential influential factors. Finally the correlation between accuracy of length prediction and translation quality is investigated.

2 Related Work

Gu et al. (2018) predict target length with fertility which essentially is an alignment of source and target tokens generated by a tool named IBM Model 2 (Brown et al., 1993). The hidden state of each source token is used to predict the number of the aligned target token. Then each source token is copied one or multiple times as decoder input, depending on the corresponding predicted length during inference. However, it may introduce noise during training as no ground truth alignment is available. As such, noisy parallel decoding (NPD) was proposed to ameliorate this issue, by applying an autoregressive teacher model to select the best translation from generated candidates sampled in fertility space.

Instead of predicting aligned length token by token mentioned above, naive linear projection has

Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 205–213
Online, November 11, 2021. ©2021 Association for Computational Linguistics

been applied extensively due to its simplicity (Guo et al., 2019; Wang et al., 2019; Li et al., 2019), that is to learn a ratio α from the training set, mapping source length M to the length of the target side: $N = \alpha \cdot M$. Obviously the frequentist statistical notion of α neglects the uniqueness of each individual instance. Thus, classifiers based on the source representation are used to predict target length or length offset (Lee et al., 2018; Ghazvininejad et al., 2019), but the fixed length determined in advance still limits the flexibility during inference.

Aiming to alleviate this issue, insertion-based models are proposed, they implicitly learn the latent variable length by dynamic insertion. Stern et al. (2019) generates text by inserting multiple tokens to readily existing unfinished sentence. Levenshtein Transformer (LevT) (Gu et al., 2019) extends the model by incorporating the deletion operation and decomposing the insertion operation into two steps — first predicting insertion positions and then specific tokens given the position.

Though various approaches are proposed to address the length prediction either explicitly or implicitly, comprehensive comparison, analysis on potential factors affecting their accuracy and the impact on final translation quality are not fully explored. In this work, we shall fill this gap.

3 Model and Dataset

In this section, we first introduce Conditional Masked Language Model (CMLM) framework (Ghazvininejad et al., 2019), on which we perform all experiments throughout this work, followed by datasets details and the experimental setup.

3.1 Conditional Masked Language Model

CMLM is a representative NAT model due to both the simple implementation and the impressive performance. It's built up on the standard Transformer architecture without the decoder self-attention mask.

Formally, given source and target sentence $X = (x_1, ..., x_k)$ and $Y = (y_1, ...y_n)$, the model first predicts target length with hidden state of the source sentence: $p(L|X)$, and initializes a sequence of [MASK] $\times L$ accordingly as decoder input. Then it repeats an alternative between predicting and re-masking tokens in sequence until all tokens are well-predicted with high confidence, this mechanism is referred as iterative refinement with mask-predict.

Notably, at step t, for an unfinished sentence that contains observed tokens $Y^{(t)}_{\text{obs}}$ and masked tokens $Y^{(t)}_{\text{mask}}$, the model predicts masked token under the condition of observed and source tokens:

$$\mathcal{L} = \sum_{y_i \in Y^{(t)}_{\text{mask}}} logP(y_i|X, Y^{(t)}_{\text{obs}}; \theta). \tag{1}$$

During training, the target sentence is corrupted by randomly replacing tokens with [MASK], and the model should learn to recover it. During inference, at step t, it predicts tokens of Y_{mask} with the argmax operation:

$$y^{(t)}_i = \arg\max P(y_i = w|X, Y^{(t)}_{\text{obs}}) \tag{2}$$

$$p^{(t)}_i = \max P(y_i = w|X, Y^{(t)}_{\text{obs}}), \tag{3}$$

and keeps the probability and token unchanged for Y_{obs}. When prediction finishes, k tokens with the lowest probabilities are re-masked, and the remaining tokens are considered as observation in the next iteration:

$$Y^{(t+1)}_{\text{mask}} = \arg\min(p^{(t)}_i, k) \tag{4}$$

$$Y^{(t+1)}_{\text{obs}} = Y^{(t)} \setminus Y^{(t+1)}_{\text{mask}}, \tag{5}$$

where k is determined by a linear decay function with respect to current t and max step T:$k = L \times \frac{T-t}{T}$. For the first step, the model creates a sequence full of [MASK]s with length of L and predicts entire target sequence with it.

3.2 Datasets

We perform experiments on two commonly-used corpora for convenient comparison, including WMT14 En↔De (train=4.5M / valid=3k / test=3k) and WMT16 En↔Ro (train=2.8M/ valid=2k/ test=2k). Following Gu et al. (2018); Zhou et al. (2020), we employ knowledge distilled (KD) data to train our NAT model. The KD data is generated with a pre-trained autoregressive Transformer-big teacher model by completely translating the source text into the target language. We set beam size as five in distillation and only keep the first candidate to make KD data equally sized with raw data. Corpus for all language pairs are tokenized into subwords with BPE (Sennrich et al., 2016), the vocabulary sizes are 42k and 40k for En↔De and En↔Ro respectively.

3.3 Experimental setup

We use the implementation of fairseq (Ott et al., 2019) for AT and NAT models, with similar model

	En-De	De-En	En-Ro	Ro-En
$\alpha_{\text{ref,train}}$	1.0982	1.0043	1.0665	0.9580
$\alpha_{\text{KD,train}}$	1.0333	0.9312	1.0258	0.9410
$\alpha_{\text{ref,test}}$	1.0401	0.9894	1.0704	0.9570
$\alpha_{\text{KD,test}}$	1.0606	0.9620	1.0757	0.9341
$\alpha^{\text{D}}_{\text{hyp,test}}$	1.0530	0.9419	1.0700	0.9275
$\alpha^{\text{R}}_{\text{hyp,test}}$	1.0779	0.9613	1.0824	0.9267

Table 1: The average length ratio for each language pairs: $\alpha^{\{\text{D}|\text{R}\}}_{\{\text{ref}|\text{KD}|\text{hyp}\},\{\text{train}|\text{test}\}} = L_{\text{tgt}}/L_{\text{src}}$, where subscripts represent for the length of reference, knowledge distilled (KD) data or hypothesis, computed on training or testing set, and the superscripts represent for model trained on KD or raw (R) data.

configuration in (Vaswani et al., 2017; Ghazvininejad et al., 2019). Transformer-big with 6/6 layers, 16 heads and 1024/4096 dimensions is used as our AT model. The NAT model is CMLM-base with 6/6 layers, 8 heads and 512/2048 dimensions.

All models are trained on 4 Tesla-V100 GPUs with gradient accumulation for 2 batches per update. Batch size is set to be maximum of 8k tokens per card. All AT models are trained for 100k updates and NAT models are trained for 300k updates. Adam (Kingma and Ba, 2015) is used as optimizer and inverse-sqrt is used as scheduler. The max learning rate and number of warmup steps are set to be 1e-4 and 4000 for AT models, 5e-4 and 20000 for NAT models.

4 Length Prediction Method

Modelling target length can be implemented in either simple or complex manners depending on learnable parameters, but it's mathematically defined as below:

$$J_L(\theta) = P(L|X;\theta), \tag{6}$$

As such, it can be solved by any maximum likelihood estimation (MLE) algorithms.

4.1 Length to Length

The most straightforward intuition is to estimate a mapping from the length of source to target in training set: $L_y = \alpha L_x$, which can be inferred by optimization and also directly measured through a statistical ratio α from the averaged length of source sentences to averaged length of target sentences of training set. We discuss four interesting findings below, acting as prior knowledge for further investigation on length prediction.

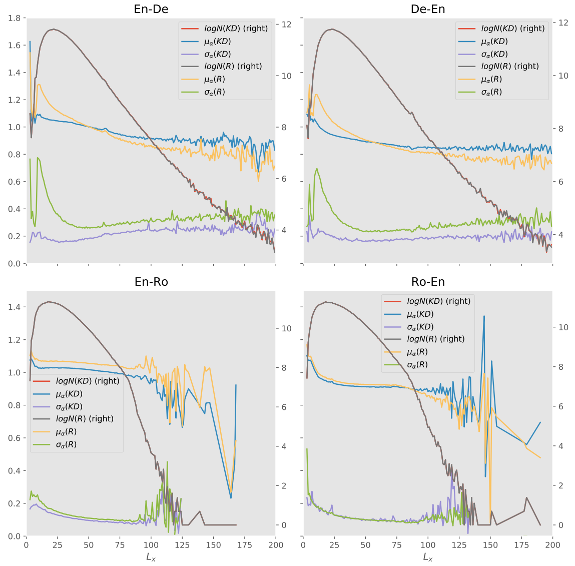

Figure 1: The distribution of the length ratio α with respect to the source length L_x (L_{src}) for raw and KD training set, where frequency is converted with logarithm. Outliers with more than 200 tokens are removed. μ and σ are mean and standard deviation of α where values are shown on the left axis. Right axis is the log frequency.

Analysis of Statistical Length Ratio Table 1 and Figure 1 show the statistical results and the distribution of the length ratio $\alpha = L_{\text{tgt}}/L_{\text{src}}$. We summarize our findings as follows:

- De and Ro sentences are usually longer than En sentences.
- For all language pairs, the length ratio changes (often decreases) with the increase of source length, which means using a linear model to fit the ratio is not enough.
- There is usually a gap between the length ratio of training and testing set which might be a causality of the length prediction error.
- For En↔De language pair, the standard deviation and the slope of the mean value for raw data is consistently larger than distilled data, demonstrating the cleanness and simplicity of the distilled data. For En↔Ro the gap of mean and standard deviation is not that big.

This method is simple and time-efficient, but the frequentist α is too generalized to ignore the distinction of each instance. Furthermore, the length of target side is not merely associated with the length of source, but various linguistic properties, such as syntax and semantics.

4.2 Latent Features to Length

Naturally, the hidden state of source sentence embedded rich information is used to incorporate latent linguistic features into target length prediction, instead of just the property of length. They explicitly or implicitly model the target length conditioned on the representation of source X, for example, $L_y = f(X;\theta)$ to predict the absolute length, $L_y = f(X;\theta)L_x$ to predict length ratio and length offset by $L_y = L_x + f(X;\theta)$, as well as fine-grained manners such as the fertility (Gu et al., 2018): $L_y = \sum_i^N f(x_i)$.

4.3 Latent Features to Latent Variables

If we consider two paradigms above as $1 \rightarrow 1$ and $N \rightarrow 1$ respectively, the third direction is to model $N \rightarrow N$. That is, each source sentence is translated into multiple targets in different lengths, which largely increases the potential to generate optimal translations by multiple candidates. In practice, it is implemented by introducing a latent variable to control length prediction $p(L_y|X,Z)p(Z|X)$ which has not been fully investigated yet due to the limitation of reference number in existing open sourced corpora. The closest prior work is sampling from length space and then selecting such as length beam and NPD (Gu et al., 2018).

The second category of approaches are extensively explored partially because current dataset and evaluation metrics applied are more friendly to them, i.e. higher performance is liable to be obtained in comparison, but in our view, the third one is a significantly promising direction to investigate.

5 Error Analysis

In this section, we analyse the predicted length of NAT models trained on distilled (denoted as D) and raw (denoted as R) data, as well as the length of the hypothesis from the teacher model (denoted as KD). We define the following metrics to quantify the error for each sentence and average them over the corpus:

$$\epsilon = L_y - \hat{L}_y \tag{7}$$

$$\epsilon^{+} = |L_y - \hat{L}_y| \tag{8}$$

$$\epsilon^{\%} = 1 - \frac{\hat{L}_y}{L_y} \tag{9}$$

$$\epsilon^{\%+} = |1 - \frac{\hat{L}_y}{L_y}| \tag{10}$$

where the ϵ is the length error, ϵ^{+} is the absolute error regardless of the skew, $\epsilon^{\%}$ is the error ratio that excludes the influence of the sentence length (e.g. 3 tokens error in a 5 tokens target and a 50 tokens target is different), and $\epsilon^{\%+}$ is the absolute error ratio. All errors are measured in the subtokens level.

Error	Model	En-De	De-En	En-Ro	Ro-En
ϵ	D	-0.0513	1.2188	-0.0426	0.7382
	R	0.0793	0.8418	-0.4727	0.8671
	KD	-0.4392	0.6890	-0.1943	0.6735
ϵ^{+}	D	2.5534	2.6787	2.6124	2.6359
	R	2.7093	2.9790	2.6610	2.6444
	KD	2.8528	2.7995	2.7802	2.7277
$\epsilon^{\%}$	D	-0.0170	0.0341	-0.0119	0.0180
	R	-0.0176	0.0224	-0.0241	0.0191
	KD	-0.0319	0.0143	-0.0168	0.0116
$\epsilon^{\%+}$	D	0.0950	0.0965	0.0933	0.0961
	R	0.1003	0.1065	0.0949	0.0974
	KD	0.1069	0.1026	0.0985	0.1015

Table 2: The average value of four error types of all language pairs. In the model column, D and R indicate models trained with the distilled or raw data, and KD represents for the performance of the teacher model.

Table 2 is the statistical result for each error type. The ϵ^{+} shows that NAT models usually have better performance, and the model trained with distilled data outperforms the teacher in a large margin. We speculate that NAT explicitly models the distribution of length. It is reasonable for NAT to have better performance compared with the implicit way of AT model. Intuitively, the model trained with raw data should perform better in length prediction because of the identical distribution in the training and testing set. But the result is the opposite. The reason behind we guess is that distilled data is cleaner and more monotonous than the raw data. Although length error is larger, it still makes the model easier to fit the length distribution compared with the raw data. Figure 2 shows the distribution of length error, indicating that the most errors (75%) are within 5 tokens, and there are slight differences between distributions of D, R and KD especially in the large error zone.

6 Influence on Translations

After investigating the performance of length prediction, we move forward to associate it with the performance of translation and try to answer the question of how length prediction influences the translation. In this section, we use BLEU (Pap-

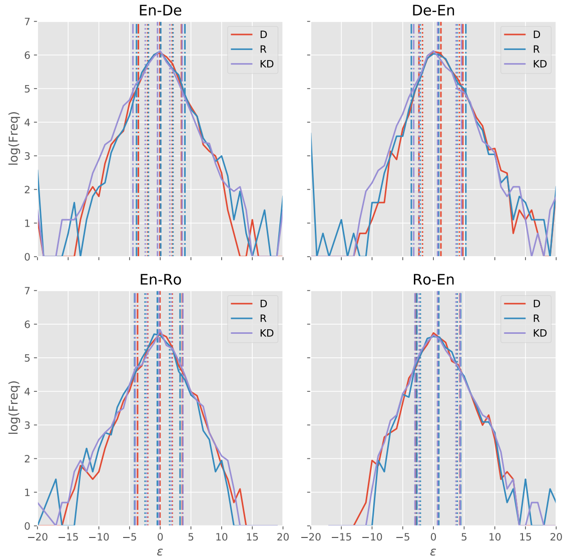

Figure 2: The distribution of the length error on the test set for each language pair, note that the error has been clipped between -20 to 20 and the logarithm of the frequency is used for clearness. The dashed line, dashed dot line and dotted line represents for the mean, 1 × std and the 75% quantile.

Lang	Data	(0, 10]	(10, 20]	(20, 30]	(30, 40]	(40, 150]
En-De	D	22.67	24.52	24.35	25.68	26.30
	R	20.23	22.67	23.14	24.39	23.81
	KD	22.36	26.36	27.41	28.01	29.51
De-En	D	22.73	29.82	28.97	31.35	30.56
	R	21.17	27.40	27.66	29.41	28.14
	KD	24.12	31.63	31.22	34.14	33.32
En-Ro	D	25.02	27.63	28.74	29.59	31.99
	R	25.31	27.17	27.87	28.52	31.05
	KD	27.29	29.31	29.51	31.39	33.41
Ro-En	D	24.18	31.72	31.94	32.13	33.12
	R	22.05	29.46	30.88	31.40	32.45
	KD	25.85	31.85	32.30	32.36	33.63

Table 3: The BLEU scores for the length interval on test set for NAT and AT model.

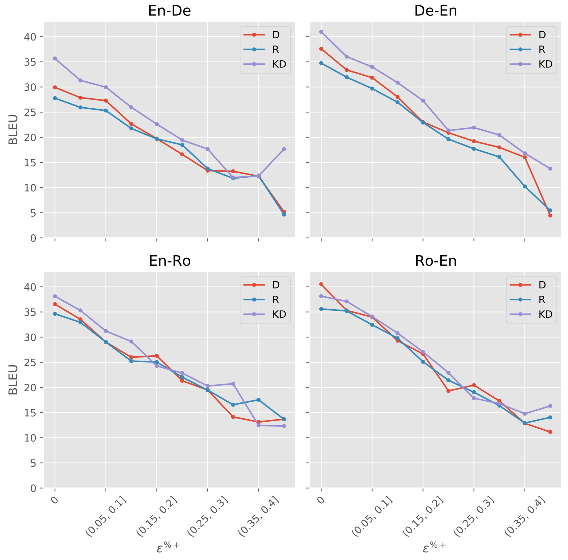

Figure 3: The BLEU scores for each error interval on test set for NAT and AT model.

ineni et al., 2002) and TER (Snover et al., 2006) as the corpus and sentence level metrics, respectively. Two additional factors are also discussed including the sentence length and the number of refinements.

6.1 Correlation to the Source Length

First of all, we use source length to group hypothesises into several buckets with equal interval size, then calculate BLEU for each bucket to find out to what extent source length can affect translation result. Table 3 shows the BLEU of each interval. We find that the first interval has relatively low BLEU which is caused by the imbalanced sample size. Later intervals don't show clear trends which means the translation performance is not quite sensitive to the sentence length thanks to the capability of Transformer architecture for modelling long term dependency (Vaswani et al., 2017), as well as the robustness of CMLM tested in (Ghazvininejad et al., 2019). From this result, we can basically exclude the influence of source length or consider it as a minor factor.

6.2 Correlation to the Length Error

Then, we group hypotheses with the $\epsilon^{\%+}$ and cut several equal sized intervals to calculate BLEU and TER for each of them. Figure 3 reveals the clear trend that with the increase of the error, the BLEU score decreases almost linearly. This result is consistent with intuition that the negative correlation between length error and the BLEU is not only applicable for AT model but also for the NAT model.

6.3 Upper-Bound of Translations

We basically conclude that incorrect length prediction has negative impact on the translation, but another question arises: if the oracle length is provided, what is the upper-bound the model could reach?

Table 4 shows the overall performance of the model trained and tested on KD or raw data with predicted ($\hat{L}$) or oracle length (L^*). Except for Ro-En result, there are clear improvements on the BLEU and TER for both types of models. In terms of the Ro→En, the contradictory result happens only on the raw test set. And noticeable unreason-

		Lang	En-De		De-En		En-Ro		Ro-En	
		Length Type	L^*	$\hat{L}$	L^*	$\hat{L}$	L^*	$\hat{L}$	L^*	$\hat{L}$
Trained on	Metric	Tested on								
D	BLEU	R	25.84	25.32	31.01	30.15	30.50	30.04	31.99	32.41
		D	56.41	53.09	61.81	57.95	59.32	55.93	79.09	74.88
	TER	R	0.54	0.58	0.48	0.50	0.50	0.53	0.48	0.50
		D	0.26	0.30	0.23	0.27	0.24	0.28	0.11	0.15
R	BLEU	R	24.56	23.58	29.21	28.17	29.81	29.09	31.66	31.42
		D	48.98	45.59	54.20	49.83	54.19	50.31	72.88	68.74
	TER	R	0.54	0.59	0.49	0.53	0.50	0.54	0.48	0.51
		D	0.31	0.36	0.28	0.34	0.27	0.32	0.14	0.19
Transformer-big		R	28.03		32.64		32.21		32.76	

Table 4: The overall performance of the NAT model trained and tested on distilled (D) or raw (R) data, with predicted ($\hat{L}$) or oracle (L^*) length. The last row shows the BLEU score of the autoregressive teacher model.

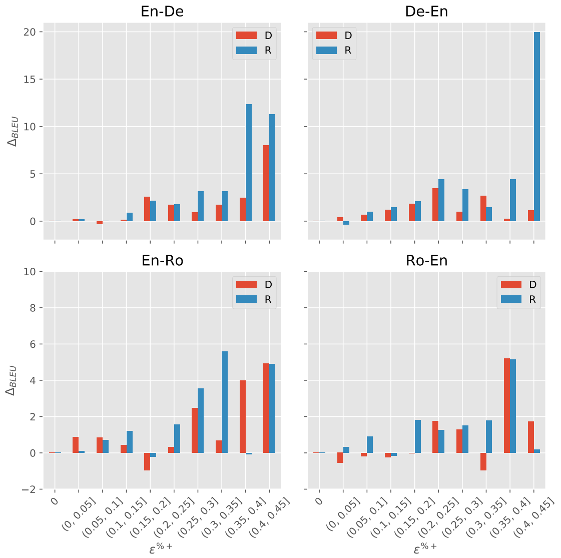

Figure 4: The difference value of the BLEU ($\Delta_{\text{BLEU}} = \text{BLEU}_{L^*} - \text{BLEU}_{\hat{L}}$) with or without oracle length for each error interval.

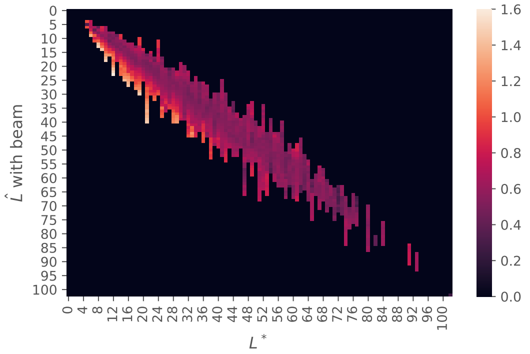

Figure 5: The heatmap of the averaged TER for candidates of length beam with respect to the oracle length.

able translations are observed in generated outputs amongst samples with higher TER, their L^* causes negative influences. This suggests that the modeling of length and token prediction are strongly related, as the length can be considered a discrete latent variable. We suspect that this counter-intuitive result is due to the overfitting of the model to the KD data and resulting in inability to correctly use the latent variable (length) from the unseen distribution (ground truth).

To further investigate the upper-bound of the performance beyond the oracle length, we perform another set of experiments with length beam to find the maximum BLEU at which a model could peak. To control the search space, we narrow search candidates within the area between the predicted length and oracle length i.e. with the beam size of $|L^* - \hat{L}|$ including L^* and $\hat{L}$. We take En→De as the example and plot the averaged TER for translations with same reference length for each beam candidates in Figure 5. In the heatmap, x axis is the oracle length and y axis is the predicted length, averaged TER for beam candidates are plotted as an vertical area. Without choosing the one with highest probability, we directly calculate the TER for all candidates towards the reference. Then, we filter hypothesises with the lowest TER for each sentence pair and calculate the corpus BLEU, which is 28.688, and the averaged ϵ^+ is 1.567. The searched upper-bound is better than those hypothesis with oracle length with approximately 3 BLEU (28.69 to 25.84), and the ϵ^+ is small than the predicted length. Through the empirical experiment results, it should be highlighted

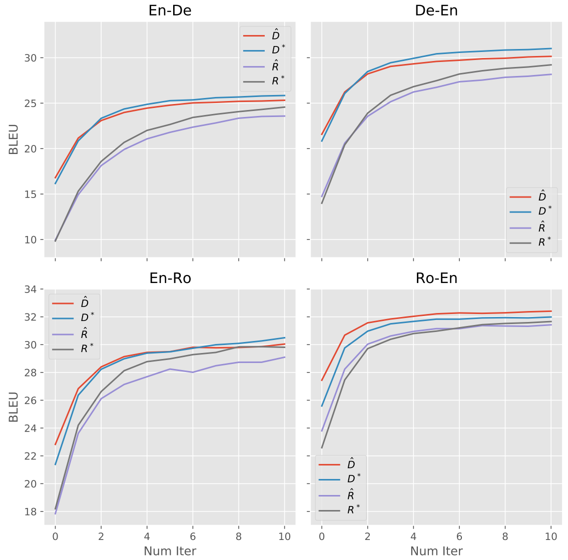

Figure 6: BLEU scores of each run when setting max refinement steps ranging from 0 to 10 with or without oracle length (asterisk or hat), where R and D represents for training on raw or distilled data.

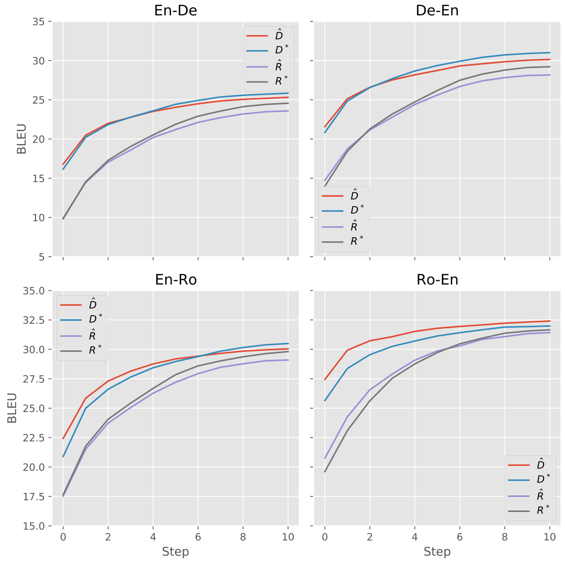

Figure 7: BLEU scores of each step with or without oracle length (asterisk or hat) when max refinement step is set to 10. R and D represents for training on raw or distilled data.

that potential improvements can be obtained if the optimal length for the model can be inferred.

6.4 Correlation to Refinement

It has been known that with multiple iterations of refinement in decoding, the translation performance could be dramatically improved. Therefore, we want to investigate the role of the length prediction during refinement. In CMLM, the refinement is achieved by re-masking predicted tokens with low confidence. Assuming we set the max step to T, at step i, $(1 - \frac{i+1}{T}) \times 100\%$ of tokens will be re-masked and predicted again at next step, note that the prediction of i-th step for different max step T might be different. In the experiment, we set maximum step as 10, run the model from 1 to 10 refinement iterations and compute the BLEU in each iteration. We also observe the translation for each iteration in 10-th run.

Figure 6 depicts the BLEU score of each run. We find that decoding with fewer iteration can be negatively scored with L^* compared to $\hat{L}$. This also happens in step wise BLEU shown in Figure 7. From this case study, we find that translations at initial several steps or with fewer refinement iterations, often have many repeated tokens. We then statistics the average repetition rate of the first step, the result shows that translations with L^* have higher repetition rate compared with using $\hat{L}$ (15.43% to 12.84% for En→De), which confirms our assumption that when given the oracle length, the model may not be able to correctly use it. However, after few steps of refinement, the incompatibility to the oracle length can be eliminated thereby makes translations with correct length can be more similar to the reference. Figure 8 is the increase of BLEU $\Delta_{\text{BLEU}} = \text{BLEU}_{L^*} - \text{BLEU}_{\hat{L}}$ for each error interval at each step, which also reveals that increases in later steps are more significant.

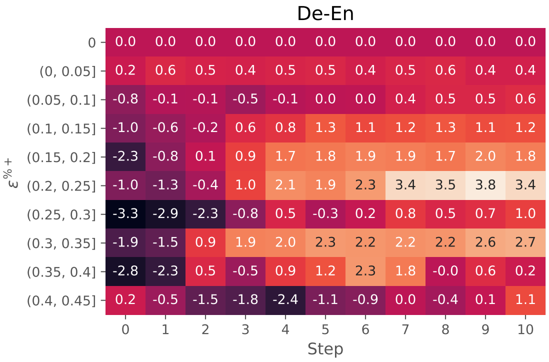

Figure 8: The heat map of the delta BLEU for each error interval and each refinement step while given the oracle length or not (max refinement step = 10).

7 Conclusion

From the analysis of length prediction as well as related factors, we have a deeper understanding of it and draw following conclusions:

- Length is an important latent variable for full or iterative based NAT models, it strongly affects the ultimate translation quality, so it should be paid more attention and emphasised in future work of the community.

- Accurately predicting the target side length is beneficial to achieving better BLEU, but it's far from the upper-bound BLEU score bonus brought by length variable.

  Specifically, even if 100% accuracy is achieved, it still can't peak the best BLEU when only judged by reference translation which is not the unique ground truth, because it ignores the linguistic diversity. We find significant improvement can be obtained by a flexible length beam search by over 3 BLEU.

This reveals that, with regard to improving the NAT translation, flexible decoding strategies are more effective than exhausting towards accurate length prediction, since essentially no ground truth of length exists owing to the complexity and diversity of languages. Elegant parallel decoding methods are more promising, such as dynamically changing the length as we evaluated, and meanwhile retaining the simplicity of mask predicting of NAT.

References

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguistics*, 19(2):263–311.

William Chan, Nikita Kitaev, Kelvin Guu, Mitchell Stern, and Jakob Uszkoreit. 2019. KERMIT: generative insertion-based modeling for sequences. *CoRR*, abs/1906.01604.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6111–6120. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3723–3730. AAAI Press.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2395–2404. PMLR.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1173–1182. Association for Computational Linguistics.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5707–5712. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Matthew G. Snover, Bonnie J. Dorr, Richard M. Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, AMTA 2006, Cambridge, Massachusetts, USA, August 8-12, 2006*, pages 223–231. Association for Machine Translation in the Americas.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.

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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5377–5384. AAAI Press.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.