IterativeVAE: Non-Autoregressive Neural Sequence Modeling by Iterative Refinement from Latent Space

刘泓尊
2018011446 计 84
liu-hz18@mails.tsinghua.edu.cn

唐李源
2018011497 计 85
tly18@mails.tsinghua.edu.cn

严宇康
2018011282 计 81
yanyk18@mails.tsinghua.edu.cn

叶鲁斌
2018080123 计 84
1524311468@qq.com

1 Introduction

现有的序列到序列 (Seq2Seq) 框架大多使用自回归方式 (Auroregressive(AR))，从左到右依次生成目标句子，生成的每个单词依赖于模型输入和前面生成的单词。相反，采用非自回归方式的 Seq2Seq 模型 (Non-Autoregressive Seq2Seq Model(NAR)) 一次生成所有单词，可以充分利用硬件并行能力，大幅提升解码效率。但是，非自回归生成模型难以生成有意义的结果，相互独立地生成每个位置的单词难以建模复杂的词间依赖关系，这使得模型的输出序列存在很严重的重复和缺失。即便是使用更复杂的结构，其表现也远远落后于自回归模型。

我们提出了一种新的基于 Iterative Refinement 的 NAR: IterativeVAE. 它通过在连续空间建模复杂隐变量，并且逐层迭代修正，以期待生成更流畅、合理的结果。这种方法几乎适用于任何的 Seq2Seq 任务：我们在机器翻译 (De→En) 和开放域对话生成 (Open-domain neural dialogue generation) 上评估了我们的模型，发现它可以在保证句子质量的同时大幅提高解码效率。开放域对话生成以对话的上文 (dialog contexts) 作为源句，将其回复作为目标，并且使用 Encoder-Decoder Model 进行生成。实验结果表明，IterativeVAE 在生成序列质量 (BLEU) 上超过了 NAR [6] Baseline，并且有比现有非自回归模型如 CMLM [5] 和 FlowSeq [12] 更快的解码速度。

2 Related Work

2.1 Autoregressive SEQ2SEQ Model

给定一个输入序列 $X = \{x_1, x_2, \cdots, x_T\}$, 自回归机器翻译模型会预测输出序列在词表上的概率分布，并将其建模为从左到右 ($L2R$) 或从右到左 ($R2L$) 的条件概率链:

$$P_{\mathcal{AR}}(T|X;\theta) = \prod_{t=1}^{T'+1} p(y_t|y_{0:t-1}, x_{1:T'};\theta) \tag{1}$$

其中 y_0(e.g. $\langle bos \rangle$) 和 $y_{T'+1}$(e.g. $\langle eos \rangle$) 代表句子的开始和结束。每个输出位置的概率分布依赖于输入序列和在它之前生成的所有输出。通常人们会使用 LSTMs [15], CNNs [4] 或 transformers [16] 来对输入序列 X 和输出序列 $Y_{0:t-1}$ 进行映射。

在生成阶段，解码算法会在本步输出是 $\langle eos \rangle$ 时停止，其中常用的方法是 Greedy-Decoding 和 Beam-Search.

2.2 Non-Autoregressive SEQ2SEQ Models

2.2.1 Overview

自回归的机器翻译模型存在 2 个主要问题：它通过迭代的方式生成句子，每次只能生成一个，导致较低的速度和 GPU 利用率；同时前面步骤生成的错误单词会被累积，导致生成偏离合理的结果；即使采用 Beam-Search，也存在词表增大时性能下降的问题 [8]。

非自回归方法抛弃了这种逐个单词生成的方法，选择同时生成所有目标序列，它可以被表示成:

$$p_{\mathcal{NAR}}(Y|X;\theta) = \prod_{t=1}^{T'} p(Y_t|X;\theta) \tag{2}$$

每个输出单词 y_t 仅仅依赖于输入序列 X, 解码操作可以一次同时完成。但是非自回归方法的面临着大量丢失和重复单词的问题 [6]，所以提升模型性能至关重要。

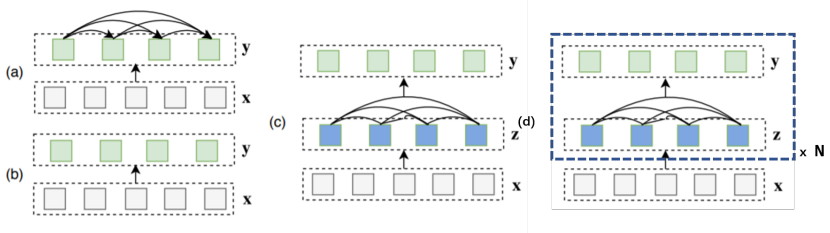

图 1: **(a)** AutoRegressive Models. **(b)** Non-AutoRegressive Models. **(c)** VAE/Flow-based Models. **(d)** Our Proposed Sequence Generation Models. x is the source, y is the target, and z are latent variables.

2.2.2 The Variational Encoder-Decoder

基于 VAE [7] 的模型选择通过隐变量来捕捉生成单词之间的依赖关系，一定程度上改进了模型性能。Variational inference([]) 引入了隐变量 z:

$$P_{\mathcal{VAE}}(y|x;\theta) = \int_z P_\theta(y|z,x)p_\theta(z|x)dz \tag{3}$$

并且引入了一个并行的 inference 网络 $q_\phi(z|y,x)$(a.k.a posterior), 通过对 z 的采样来近似上述积分。这些模型优化证据下界 (ELBO):

$$\log p(y|x;\theta) \geq E_{q_{\phi(z|y,x)}}[\log p(y|z,x;\theta)] - KL(q_{\phi(z|y,x)}||p_{\theta(z|x)}) \tag{4}$$

上式可以被看做 "重构损失" $\log p(y|z,x;\theta)$ 和 posterior 与 prior 之间的 KL 散度。这项工作 [2] 基于 RNNs 在词表这一连续空间建模隐变量，还有工作 [1] 基于 transformers 引入了 Variational Attention 机制，把注意力向量也视为隐变量，解决了 *bypassing phenomenon* 问题，使得隐变量有更强的表征能力。

2.2.3 Iterative Refinement and Mask-Predict Models

由于单次生成所有单词的效果较差，部分工作选择采用半自回归的方式，以句子为单位逐步修正前面几次生成的结果。这项工作 [9] 将上一次 Decoder 的输出作为下一次迭代的输入，并且优化每一步输出的重构损失:

$$\mathcal{L}(\theta) = -\sum_{l=0}^{L-1}(\sum_{t=1}^{T}\log p_\theta(y_t|\hat{Y}^{l-1}, X)) \tag{5}$$

其中 $\hat{Y}^{l-1} = (\hat{y}^{l-1}, \cdots, \hat{y}_T^{l-1})$;

Mask-Predict [5] 的方法则在训练阶段随机遮住部分目标序列，让模型以 "上下文填空" 的方式预测输出单词，并在生成阶段采用 "生成 → 遮蔽 → 再填空" 的方式逐步优化生成结果，达到了很不错的性能。

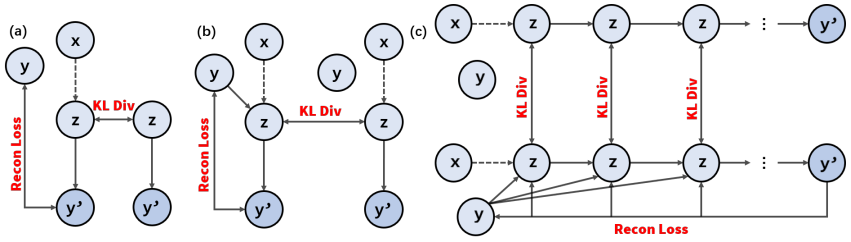

图 2: Graphical model representations. **(a)** Variational autoencoder (VAE). **(b)** Variational encoder-decoder (VED). **(c)** VED with iterative refinement (ours). **Dashed lines**: Encoding phase. **Solid lines**: Decoding phase.

3 Methods

3.1 Overview

基于 Latent Variable 和 Iterative Refinement 的思路，我们提出了一种新的基于 transformer 的非自回归模型: IterativeVAE. 如图3所示，这个模型主要由 4 个模块组成: encoder stack, decoder stack, target length predictor 和 translation predictor.

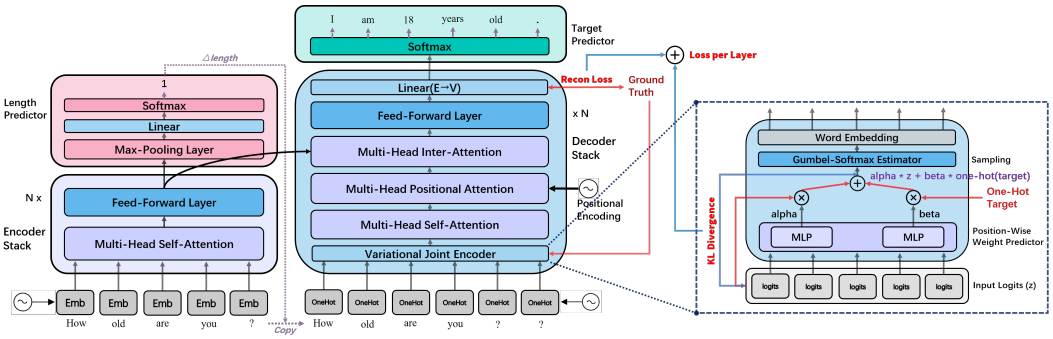

图 3: The architecture of the proposed model, where the black solid arrows represent differentiable connections and the purple dashed arrows are non-differentiable operations. Each sublayer inside the encoder and decoder stacks are standard, and uses both absolute and relative attention.

3.2 Encoder Stack

和自回归 Transformer 类似，Encoder 和 Decoder 都由 Multi-head Attetion 和 Feed-forward Networks(MLPs) 组成. 我们使用 $N = 6$ 的 Encoder Stack, 模型组成和标准 transformer 相同。给定模型的输入 $x = \{x_1, \cdots, x_n\}$，Encoder 将在最后一层输出它的上下文表示 $H = \{h_1, \cdots, h_n\}$.

3.3 Decoder Stack

3.3.1 Target Length

为了生成目标序列，我们首先需要预测输出句子的长度，并以合适的方式从 H 得到 Decoder 的输入 H'. 我们使用了先前工作 [6] 的思路，预测源语句和输出语句长度的差值 $\triangle m$, 它通过一个预测 $[-20, 20]$ 的分类器得到。分类器的输出为:

$$p(\triangle m + 20|x) = softmax(W_p(maxpool(H) + b_p) \tag{6}$$

之后我们将 H 均匀地映射到 H', 也就是

$$H'_i = H_{\lfloor (n*(i/m)} \rfloor \tag{7}$$

其中 n 为输入序列长度, m 为输出序列长度。因为上述操作是在隐空间进行的，所以梯度可以从 Decoder 回传到 Encoder.

3.3.2 Decoder Structure

Decoder 也由 $N = 6$ 层 Decoder Blocks 组成。与自回归方式不同的是, 因为我们不再需要根据先前的词生成当前位置的单词, 我们不再采用 Masked Multi-head Attention [16], 而是仅仅采用 diag-mask 的 Multi-head Attention, 使得模型能够依赖上下文学习到输出序列的隐空间表示。

没有了自回归特性使得模型难以捕捉语序关系，为了让输出的句子因位置而有所差别，我们借鉴了先前工作的方法，采用 Positional Attention [6] 和 Relative Attention [14] 来给模型提供更多的位置信息，使得模型能够在对应的位置生成更合理的结果。

Positional Attention 该模块基于 Multi-head Attention 模块，将 query 和 key 设置为 Positional Encoding[1]. 先前的工作 [6] 表明 Positional Attention 将提供比为 Positional Encoding 更强的位置信息。

Relative Attention 对于相对位置信息的引入，我们参考了这项工作 [14] 的做法，在 Self-Attention 中用一个可学习的 embedding 矩阵根据 query 和 key 的位置之差构建相对位置的 Attention. 形式化地，可学习参数矩阵为 w, 那么

$$a_{ij}^K = w_{clip(j-i,k)}^K \tag{8}$$

$$e_{ij} = \frac{x_i W^Q (x_j W^K)^T + x_i W^Q (a_{ij}^K)}{\sqrt{d_z}} \tag{9}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tag{10}$$

$$z_i = \sum_{j=1}^{n} \alpha_{ij} (x_j W^V + a_{ij}^V) \tag{11}$$

其中 $W_K, W_Q, W_V \in \mathbb{R}^{d_x \times d_z}$ 为参数矩阵, 输入 $x \in \mathbb{R}^{d_x}$, 输出 $z_i \in \mathbb{R}^{d_z}$, $w_i^K, w_i^V \in \mathbb{R}^{d_a}$, $w_K = (w_{-k}^K, \cdots, w_k^K)$, $w_V = (w_{-k}^V, \cdots, w_k^V)$. $clip(j-i, k) = max(-k, min(k, j-i))$.

3.3.3 Latent Variables

我们使用 Variational Decoder Stack 在连续的词表空间获得隐变量。Decoder 的每层输入 H_i' 会经过 MLP 映射到词表空间 $V_i \in \mathbb{R}^{\mathbb{V}}$，之后将 one-hot 形式的目标序列表示 T(作为 Hint 信息以提高隐变量能力) 和 V_i 在每个位置做加权求和，之后进行多项式分布采样 (使用 Gumbel Softmax 进行重参数化) 得到隐变量 $z_i \in \mathbb{R}^V$。之后通过 Embedding 矩阵映射到 $\mathbb{R}^d$，作为 Decoder Block 的输入。

因为隐变量是逐层得到的，而不是基于整个的 Decoder，所以上述方法能够最大限度地减少增加的参数量，保证生成速度。

[1]The positional encoding p is computed as $p(j,k) = \sin(j/1000^{k/d})$(for even j) or $\cos(j/1000^{k/d})$(for odd j), where d is the hidden size, j is the timestep index and k is the channel index.

IterativeVAE 在训练阶段逐层优化证据下界 (ELBO)，并使用逐层减少的 KL_{weight} 控制 Hint 信息比重:

$$\mathcal{L}(\theta; x, y) = \mathbb{E}_{h'_i \sim q_\theta(h'_{i-1}|x,y)}[\log p_\theta(h'_i|h'_{i-1}, x)] - \alpha_i \dot{K}L\left[q_\theta(h'_{i-1}|x, y)||p(h'_{i-1}|x)\right] \quad (12)$$

其中 h'_{i-1} 是第 i 层的输入，h'_i 是第 i 层的输出，x 为源序列, y 为目标序列。模型通过引入隐变量和逐层修正生成结果的方式，能够获得较好的生成质量，同时在每层上进行 Refinement 保证了生成速度不会下降。

4 Experiments

4.1 Experimental Settings

Datasets 我们分别在单轮对话任务和机器翻译任务上评估我们的模型。在开放域对话任务上，我们使用 OpenSubtitles[2](100K pairs) 作为训练集，the Internet Movie Script Database(IMSDB[3]，数据由我们实现的爬虫脚本获取) 作为验证和测试集 (just tokenized, 10K pairs)；在机器翻译上，我们使用 IWSLT14 [3](de→en)(use byte-pair encoding(BPE [13])) 训练 (150K pairs) 和测试 (10K pairs)。

Baselines Baselines 包括自回归 AR(use Standard Encode-Decoder Transformer), 半自回归 CMLM [5], Flow-Based model: FlowSeq [12] 和 Energy-based mdoel: ENGINE. 其中 AR 和 CMLM 测试了 $d_{model} = 256$(base) 和 $d_{model} = 512$(large) 的模型参数。其中 AR, CMLM, NAT 来自于我们框架的实现。

Modules and Hyperparameters 对于我们提出的模型，参数配置为 $n_{layers} = 6$, $n_{head} = 8$, source/target embedding 共享, Base(default) models ($d_{model}/d_{hidden} = 256/1024$), Large models ($d_{model}/d_{hidden} = 512/2048$). Feed-forward Dropout = 0.3, Attention Dropout = 0.1.

Training Details 所有的实验结果都在 1 张 Nvidia TITAN Xp(12 G) 上进行 $mini-batch = 512$ 的训练。我们使用 RAdam [11] 作为优化器, $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e^{-8}$, $weight\_decay = 0.001$. 学习率使用指数退火，$init\_lr = 0.01, lr\_decay = 0.8$.

4.2 Automatic Evaluation

我们使用 BLEU1/BLEU2 评估对话生成质量, BLEU4 评估翻译质量, 使用 Distinct-1(dis-1), Distinct-2(dis-2) [10] 评估 unigrams 和 bigrams 的多样性。我们还在数据集上 finetune GPT-2 来评估生成句子的 Language Model Score(LM Score)，来评价模型生成结果的连贯性。测试结果如表1所示。

[2] http://opus.nlpl.eu/OpenSubtitles.php

[3] https://www.imsdb.com

从结果可以看到，我们提出的模型在保持了 NAR 模型生成速度的同时，在 BLEU、Distinct 和 LM Score 等指标上都超过了 NAT baseline, 拉近了非自回归模型和自回归模型 (如 AR) 与半自回归模型 (如 CMLM) 之间的差距。

	#iter	BLEU1/2↑	dis-1/dis-2↑	LM Score↓	Avg.length	Lat(ms)↓/Spdup↑
AR-base(b=1)	N	4.89/3.44	0.814/0.806	4.54	9.8	420/2.15x
AR-base(b=4)	N	5.17/3.56	0.812/0.805	**4.33**	12.2	624/1.45x
AR-large(b=1)	N	4.96/3.48	**0.821/0.811**	4.61	10.3	610/1.48x
AR-large(b=4)	N	**4.96/3.51**	0.800/0.802	4.34	10.3	906/1.0x
NAT-base	1	1.51/0.33	0.011/0.017	5.87	8.8	84/10.78x
CMLM-base	1	2.1/0.53	0.277/0.639	4.93	10.2	81/11.18x
CMLM-base	4	2.09/0.71	0.334/0.662	4.70	10.2	291/1.44x
CMLM-large	1	2.65/0.66	0.156/0.482	4.65	12.5	191/4.74x
CMLM-large	4	**2.93/0.69**	**0.211/0.539**	4.65	12.5	774/1.17x
ENGINE-large	1	2.07/0.69	0.332/0.663	5.85	12.9	215/4.76x
FlowSeq-base	1	1.85/0.59	0.042/0.034	**4.50**	4.5	133/6.81x
IVAE-base(ours)	1	1.86/0.61	0.143/0.313	4.81	7.9	**105/8.62x**
IVAE-large(ours)	1	**1.89/0.61**	**0.149/0.334**	**4.79**	8.1	167/5.42x

表 1: Automatic Metrics Evaluation for Different Models on **IMSDB dataset**. 'Lat' stands for Latency(ms), 'Spd' stands for 'Speedup compared to the slowest model', 'dis' stands for 'Distinct Metirc'.

4.3 Qualitative Analysis

4.3.1 Decoding Speed and Quality

非自回归模型因为并行生成所有序列而得到了较高的速度。为了衡量模型生成速度和生成质量，我们在 IWSLT14 上测试模型的表现，如图4。从图4左图可以看到，非自回归模型在牺牲一定生成质量的情况下，能够比半自回归和非自回归模型的生成速度取得了极大提高。IterativeVAE 在翻译任务上的的生成质量和其他非自回归 Baseline 相近，同时也能维持较高的速度。自回归模型生成质量明显领先于非自回归模型，但是在速度上远远慢于非自回归方法。

图4右图是我们的方法比 NAT 和 FlowSeq 在 IWSLT14 任务上 BLEU1 指标上的提升，我们的模型引入了词表连续空间的隐变量，相应地增加了参数量，但是生成速度的略微下降换来了性能的提升，这也验证了我们的设想。实际上，我们的模型正是在 NAT [6] 的 Decoder 上增加了隐变量生成模块，这也作为消融实验验证了我们所提出方法的有效性。

4.3.2 Sampled Responses

我们随机选择了若干输入句子，来对比自回归模型 AR, 半自回归的 CMLM 以及非自回归的 NAR 和我们提出的 IterativeVAE 的表现与特点，如表2所示。从表中可以看到，

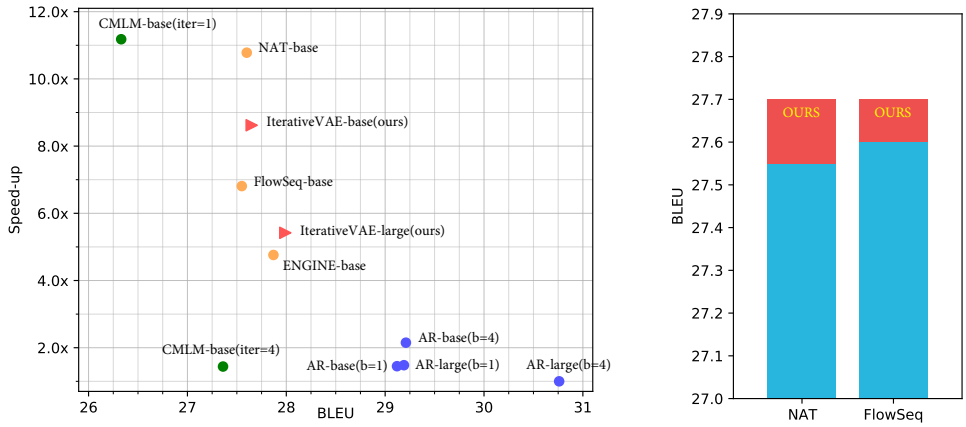

图 4: **Left**: The trade-off between speed-up and translation quality(BLEU) on dataset **IWSLT14(de→en)**. **Right**: Performance(BLEU) improved by our method compared with **NAR** and **FlowSeq** baseline on **IWSLT14**.

AR 模型的句子在语义和通顺度上都有很好的表现。CMLM 在短句子上和 AR 模型相似，但是在长句子上表现略差，说明半自回归模型依然有一些局限。注意到 NAT 模型几乎不能生成任何有意义的结果，在对话任务上只是简单词的无意义重复，在机器翻译上效果稍好，但是依然有大量的重复词和遗漏词 (这也说明了对话任务更明显的多模态特征与挑战性)。我们的模型 IterativeVAE 能够修正 NAT 的部分缺陷，在对话数据集上能够生成较为丰富的结果，说明模型捕获到了句子的语义特征和上下文关系。尽管 IterativeVAE 比 NAT 的性能有大幅提升，但是和自回归与半自回归模型依然差距较大。

Src	**AR**	**CMLM**	**NAR**	**IterativeVAE**(ours)
nice to meet you what a day	nice to meet you	do you know what i 'm doing this yet ?	oh on god no	do you 't your your of your a bathroom ?
do you listen to this crap ?	yes I do.	i don 't know my name	i 's you a the ?	i don 't remember my forte
pick you up friday then	oh thank you very much.	that 's very smart	ok.	it 's very good.
als ich 11 Jahre alt war , wurde ich eines Morgens von de Klängen heller Freude geweckt .	at 11 years old , I was one morning from the kshed .	and when I was 11 years old I was swimming one morning by the length of real joy .	when I was 11 old old I I the the the the of joy .	when I was 11 years old I was swimming one morning of real joy .
er rief : ' die Taliban sind weg ! '	he called the Taliban away !	he called , ' The Taliban have gone away ! '	he called , Taliban Taliban Taliban away !'	he said , ' the taliban have gone!
diesen Morgen werde ich niemals vergessen .	this morning I'll never forget .	and this morning I will never forget .	I will forget forget forget morning .	I will never forget this morning .

表 2: Sampled results of different models.

4.3.3 Refinement Results

为了了解模型是否真的在'Iterative Refinement'，我们随机选取了一些句子，它们代表模型在第 1-6 层的输出，如表3所示。可以看到无论是在对话任务还是机器翻译任务上，

模型都可以逐层改进上一层的结果，生成更加通顺和合理的句子。这与我们的预期是一致的。

Layer# Output	**IMSDB**	**IWSLT14**
Source	I told him there was a gas leak.	Nur so konnten wir beide zur Schule gehen.
Target	But he didn't listen to you.	That was the only way we could both get educated.
1	He he him with him him	That's the the way way can go go to.
2	He was him with his him	That's the the we we can go go to.
3	It was him at his work	That's the the we we could go go school.
4	He was him with his work	That's the the way i can go to school.
5	It was him at to work	That's the only way we could go to school.
6	It was him not to work	That's just the way we could go to school.

表 3: Iterative refinement sampling results on **IMSDB** and **IWSLT14** dataset.

5 Conclusion

我们提出了一种基于 Iterative Refinement 的 Variational Seq2Seq Model，它可以在连续的隐空间建模复杂的特征分布，并使用迭代的方式不断改进生成结果。IterativeVAE 可以在保证 NAT 推理速度的同时提高生成质量，并且可以适用于机器翻译和对话生成等多个领域。未来的潜在发展包括深入研究模型的隐空间，从而获得更好的隐空间表示；以及引入 Masked Language Model 来进一步提高生成质量。

6 Acknowledgments

本工作得到了 CoAI 课题组的老师与学长们的大力支持。黄斐学长提供了本工作的设计思路，尤其要感谢他这一学期以来的悉心指导和帮助！他每周都会花很多时间带我一起探讨模型存在的问题，耐心解答我的疑惑，认真核对我的实现，不时对我进行鼓励与帮助。在本学期接近结束的时候，我曾因为工作遇到瓶颈而灰心丧气，但黄斐学长给了我充足的信心，提出了很多切实有效的建议，最终使得本工作如期完成，再次表示感谢！我还要感谢黄民烈老师的真切嘱托，这让我在如何平衡学习与科研、如何在课题组中快速成长等方面受益匪浅！此外，我要感谢 CoAI 课题组提供的宝贵的硬件资源，你们的支持是本工作得以完成的基石和动力！

参考文献

[1] Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. Variational attention for sequence-to-sequence models, 2018.

[2] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015.

[3] Mauro Cettolo, C. Girardi, and Marcello Federico. Wit3: Web inventory of transcribed and translated talks. *Proceedings of EAMT*, pages 261–268, 01 2012.

[4] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

[5] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models, 2019.

[6] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation, 2017.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[8] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics.

[9] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement, 2018.

[10] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055, 2015.

[11] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *CoRR*, abs/1908.03265, 2019.

[12] Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. Flowseq: Non-autoregressive conditional sequence generation with generative flow. 2019.

[13] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015.

[14] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations, 2018.

[15] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.