Learning to Generalize to More: Continuous Semantic Augmentation for Neural Machine Translation

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Weihua Luo, Jun Xie, Rong Jin
Alibaba DAMO Academy, Chinese Academy of Sciences

论文来源：ACL2022
分 享 人：语音翻译条线 陈伟

1. 背景介绍

➢ 机器翻译中的数据增强

- **离散空间中数据增强的困境：**
- 一、离散空间中的数据增强**实例缺乏多样性。**
- 二、离散空间中的数据增强会**破坏语句本意。**
- **核心idea：**

从隐含语义的连续空间生成多样的训练数据

- **主要挑战**
- 一、是否可以在连续的语义空间中进行数据增强？从而最大程度保持语义的不变性。
- 二、如何有效和高效的从语义空间中生成多样性的训练数据？从而保证数据增强实例的多样性。

➢ 本文贡献

- 提出了一种**新的数据增强范式**CSANMT，该范式将邻接语义区域作为每个训练实例的邻接流形。
- 该方法属于插件式，**不受模型架构限制。**
- 在**高、低资源机器翻译任务上都有效。**特别的，在泛化能力有限的训练数据下仍可生成更多不可见实例。**仅使用25%的训练数据即可达到基线模型效果。**

2.本文提出方案

➢ 整体流程：

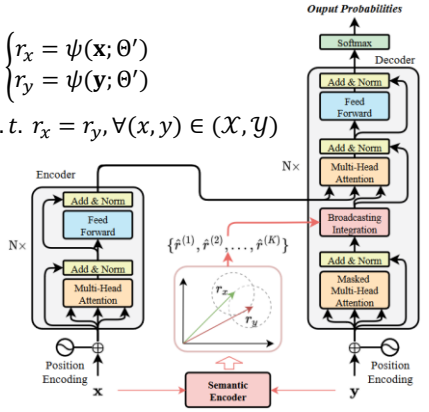
- **Stage1**：从头预训练一个语义编码器Θ'建模连续语义空间，将源句**x**和目标句**y**分别转换为实值向量r_x和r_y

$$\begin{cases} r_x = \psi(\mathbf{x};\Theta') \\ r_y = \psi(\mathbf{y};\Theta') \end{cases} \quad s.t.\ r_x = r_y, \forall (x,y) \in (\mathcal{X},\mathcal{Y})$$

- **Stage2**：对于每个句对，有一个邻近的语义区域$v(r_x, r_y)$，从中采样一系列K个向量$\mathcal{R}$，集成在解码器中。

$$\mathcal{R} = \{\hat{r}^{(1)}, \hat{r}^{(2)}, \dots, \hat{r}^{(K)}\}, \qquad \hat{r}^{(K)} \sim v(r_x, r_y)$$

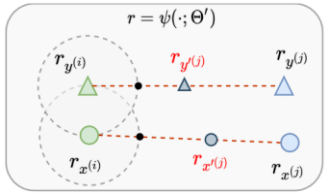
$$\hat{o}_t = W_1 \hat{r}^{(k)} + W_2 o_t + b$$



➢ 难点及方案：

- **Q1**：如何优化语义编码器，使其为每个训练对产生一个有意义的邻接语义区域？——切线式对比学习

几何视角下的邻接语义区域$v(r_x, r_y)$：
以$r_{x^{(i)}}$和$r_{y^{(i)}}$为中心的两个闭合球的并集。



$$J_{ctl}(\Theta') = \mathbb{E}_{(\mathbf{x}^{(i)},\mathbf{y}^{(i)})\sim\mathcal{B}}\left(log \frac{e^{s(r_{x^{(i)}}, r_{y^{(i)}})}}{e^{s(r_{x^{(i)}}, r_{y^{(i)}})} + \xi}\right) \qquad \xi = \sum_{j\&j\neq i}^{|B|} \left(e^{s(r_{y^{(i)}}, r_{y'^{(i)}})} + e^{s(r_{x^{(i)}}, r_{x'^{(i)}})}\right)$$

线性插值形成困难负样本：

$$r_{x'^{(j)}} = \begin{cases} r_{x^{(i)}} + \lambda_x \left(r_{x^{(j)}} - r_{x^{(i)}}\right), \lambda_x \in (d/d'_x, 1] & if\ d < d'_x \\ r_{x^{(i)}} & if\ d \geq d'_x \end{cases} \qquad r_{y'^{(j)}} \text{类似} r_{x'^{(j)}}$$

- **Q2**：如何从邻接语义区域中有效且高效地获取样本？——混合高斯循环链采样

变换偏置向量$\hat{r}$的范数和方向

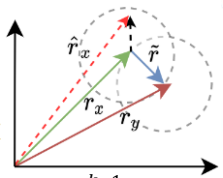
$$\hat{r} = r_y - r_x$$

$$\hat{r} = r + \omega \odot \tilde{r}, \omega \in [-1,1]$$

采样即寻找一组(K个)尺度向量

$$\omega^{(k)} \sim p(\omega|\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(k-1)})$$

$$p = \eta\mathcal{N}(\mathbf{0}, \mathrm{diag}(\mathcal{W}_r^2)) + (1.0 - \eta)\mathcal{N}\left(\frac{1}{k-1}\sum_{i=1}^{k-1}\omega^{(i)}, \mathbf{1}\right)$$



Algorithm 1 MGRC Sampling

Input: The representations of the training instance $(\mathbf{x}, \mathbf{y})$, i.e. r_x and r_y.
Output: A set of augmented samples $\mathcal{R} = \{\hat{r}^{(1)}, \hat{r}^{(2)}, \dots, \hat{r}^{(K)}\}$

1: Normalizing the importance of each element in $\tilde{r} = r_y - r_x$: $\mathcal{W}_r = \frac{|\tilde{r}| - min(|\tilde{r}|)}{max(|\tilde{r}|) - min(|\tilde{r}|)}$
2: Set $k = 1$, $\omega^{(1)} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\mathcal{W}_r^2))$, $\hat{r}^{(1)} = r + \omega^{(1)} \odot (r_y - r_x)$
3: Initialize the set of samples as $\mathcal{R} = \{\hat{r}^{(1)}\}$.
4: **while** $k \leq (K-1)$ **do**
5: $k \leftarrow k + 1$
6: Calculate the current scale vector: $\omega^{(k)} \sim p(\omega|\omega^{(1)}, \omega^{(2)}, \dots, \omega^{(k-1)})$ according to Eq. (6).
7: Calculate the current sample: $\hat{r}^{(k)} = r + \omega^{(k)} \odot (r_y - r_x)$.
8: $\mathcal{R} \leftarrow \mathcal{R} \cup \{\hat{r}^{(k)}\}$.
9: **end while**

3. 实验结果

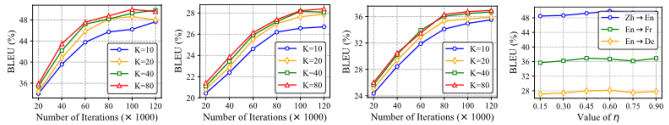
➢ 基准数据集上性能提升

Method	#Params.	Valid.	MT02	MT03	MT04	MT05	MT08	Avg.
Transformer, base (*our implementation*)	84M	45.09	45.63	45.07	46.59	45.84	36.18	43.86
Back-translation (Sennrich et al., 2016a)*	84M	46.71	47.22	46.86	47.36	46.65	36.69	44.96
SwitchOut (Wang et al., 2018)*	84M	46.13	46.72	45.69	47.08	46.19	36.47	44.43
SemAug (Wei et al., 2020a)	86M	-	-	-	49.15	49.21	40.94	-
AdvAug (Cheng et al., 2020)	-	49.26	49.03	47.96	48.86	49.88	39.63	47.07
CsANMT, base	96M	**50.46**	**49.65**	**48.84**	**49.80**	**50.40**	**41.63**	**48.06**

Model	WMT 2014 En→De			WMT 2014 En→Fr		
	#Params.	BLEU	SacreBLEU	#Params.	BLEU	SacreBLEU
Transformer, base (*our implementation*)	62M	27.67	26.8	67M	40.53	38.5
Transformer, big (*our implementation*)	213M	28.79	27.7	222M	42.36	40.3
Back-Translation (Sennrich et al., 2016a)*	213M	29.25	28.2	222M	41.73	39.7
SwitchOut (Wang et al., 2018)*	213M	29.18	28.1	222M	41.62	39.6
SemAug (Wei et al., 2020a)	221M	30.29	-	230M	42.92	-
AdvAug (Cheng et al., 2020)	†65M	29.57	-	-	-	-
Data Diversification (Nguyen et al., 2020)	†1260M	30.70	-	†1332M	**43.70**	-
CsANMT, base	74M	30.16	29.2	80M	42.40	40.3
CsANMT, big	265M	**30.94**	**29.8**	274M	**43.68**	**41.6**

结论:NIST中英比Transformer平均提4.2个点，比SOTA提1个点。WMT14英德big模型高Transformer 2.2个点，WMT14英法big模型高1.3个点

➢ K和η的参数效果



(a) NIST Zh→En (b) WMT14 En→De (c) WMT14 En→Fr (d) Effect of η.

结论：K越大越好，但是太大提升有限，因为多样性不是无穷的，MGRC在K较大时饱和，η在0.6时平衡两种高斯分布形式效果最好。

➢ 词汇丰富度和语义忠实度

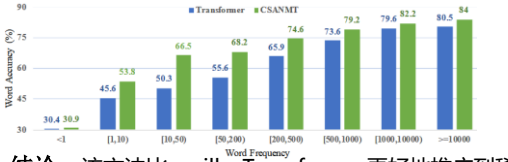
	TTR			BLEURT Score		
	Zh	De	Fr	Zh	De	Fr
Human	7.58%	22.08%	13.98%	-	-	-
Trans.	6.95%	20.32%	11.76%	0.570	0.635	0.696
CsANMT	**7.13%**	**21.26%**	**12.91%**	**0.581**	**0.684**	**0.739**

➢ 语义编码器的消融

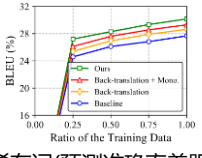
Model	BLEU	Dec. speed
Transformer-base	27.67	reference
Default 4-layer semantic encoder	30.16	0.95×
Remove the extra semantic encoder	28.71	1.0×
Take PTMs as the semantic encoder	31.10	0.62×

结论：该方法缩小了人工和机器翻译之间词汇多样性差距。同时对生成翻译的保留语义方面表现出更好的能力。最后，建模语义编码器是有效的。

➢ 词预测的准确度



➢ 离散增强 vs 连续增强



结论：该方法比vanilla Transformer更好地推广到稀有词(预测准确率差距高达16%)。另外性能始终优于BT(回译)，且25%的数据就能达到基线性能。