神经机器翻译词级别正则化技术研究

邱石贵，章华奥，段湘煜，张民

（苏州大学计算机科学与技术学院，江苏 苏州，215000）

摘要：神经机器翻译是利用大型人工网络对翻译建模的过程，在机器翻译质量上获得很大提升，但是作为深度网络模型，神经机器翻译模型面临泛化能力不足的问题以及在低资源场景下更容易出现过拟合现象。针对此问题，本文研究了词级别上的正则化技术(Word Regularization, WR)，通过对模型输入句子中的单词进行随机的扰动，以此削弱数据的特异性，从而抑制模型对于数据的过度学习防止过拟合，同时提高模型对于未知数据的泛化能力。我们选择 Transformer 模型在中-英数据集和英语-土耳其语数据集上进行相关的实验，结果显示模型在训练收敛后更加稳定不易出现过拟合的情况，并且翻译质量也有明显提升。

关键词：神经机器翻译；泛化能力；过拟合；正则化

中图分类号：TP391　　**文献标志码：**A

神经机器翻译（Neural Machine Translation, NMT）将机器翻译任务看作是一种序列到序列的转化问题，其端到端的建模过程在 2014 年由 Sutskever 等人[1]提出，该方法采用编码器解码器框架，不依赖人工定义的特征，在短句上的性能十分优越；2015 年 Bahdanau 等人[2]在此基础上引入注意力机制在翻译性能上获得显著的提升，并超越了传统的统计机器翻译 (Statistical Machine translation，SMT)；2017 年，由 Ashish Vaswani 等人[3]提出的 Transformer 模型更是在翻译的性能和速度上进一步得到了提升，该模型仅通过注意力机制进行建模，输入的源语言句子通过编码器编码成上下文内容的中间表示，基于这些句子的中间表示，解码器逐词地生成目标语言的译文。

相比于传统的 SMT，NMT 是端到端的训练，全局只优化一个目标函数，这样的网络对于上下文信息的学习和利用更加充分。深度神经网络模型通常拥有千万级的参数，对于句子语义特征，句法结构上的特征拥有更加强大的捕获能力。但是庞大复杂的网络结构带来的拟合能力造成 NMT 在低资源场景下很容易出现过拟合的现象，以及对于未知数据的泛化能力不足，导致在真实数据上较低的翻译性能[4]。因此本文旨在如何提升 NMT 的泛化能力以及防止 NMT 在低资源场景下的过拟合现象。

为了提升模型泛化能力并且有效阻止过拟合，增加训练数据是比较简单的做法，但是高质量的平行语料的获取是费时费力的，所以一般会应用数据增强技术来对原始数据进行扩展，这在图像处理领域是应用广泛的技术，但是针对文本这样的数据进行增强需要更加地谨慎，因为文本作为离散数据存在句法上的约束，如果像对图片一样裁剪，旋转，那文本大概率就不是一个表达正确信息的句子了，所以文本数据的增强技术还需要更多地探索和实践[5]。除了数据增强，正则化技术也是解决模型泛化能力，阻止过拟合的有效方法，正则化技术通常是在兼顾模型性能的前提下，约束模型复杂度的一种技术，模型对于训练数据的细节过度学习主要是因为模型过于复杂导致的，所以通过一定的技术来削弱模型的学习能力或者简化模型的结构是合理的[6]。正则化技术的实现可以从两个角度进行考虑：一、通过简化模型结构，获得一个对数据稍微欠拟合的模型，以此保证模型的泛化能力并且防止过拟合，类似的方法

有 Srivastava 等人[7]提出的 dropout，通过对模型的部分神经元随机“丢弃”，以此集成若干更为简单的模型；二、通过减少数据的细节或者削弱监督约束来干扰模型对于数据特定细节的学习，比如对输入数据信息进行加噪或者 Christian 等人[8]提出的标签平滑技术，而本文的方法就是第二种思路。

本文研究的词级别正则化技术，目标在于通过对 NMT 的编码器和解码器两端的输入句子进行扰动来减少训练数据的细节并且削弱监督信号的约束，从而抑制模型对训练数据的过度学习、防止过拟合。由于文本数据离散的特性，每一个单词在句子中扮演着不同的语义角色，相比于给句子整体的扰动，给予粒度更小的单词上的扰动会更加灵活并且更具针对性，而除此之外，采样机制的应用也让这样的扰动更具随机性，而如何给予输入句子单词合适的扰动是本文的研究重点。

本文会从 5 个方面对词级别正则化技术进行讨论：(1)背景知识，介绍基准系统 Transformer 的相关知识。(2)相关工作，介绍神经机器翻译中常见的正则化技术；(3)词正则化，详细介绍本文所研究的词级别正则化技术的细节。(4)实验设置和结果，设计相关实验来验证词正则化技术的有效性。(5)模型分析，通过不同的评价指标来综合评估词正则化的给模型带来的影响。

1 背景知识

本节介绍基于注意力机制的神经机器翻译，以 Transformer 为例，如图 1 所示，是典型的编码器解码器结构。其中编码器是由多层注意力层和前馈网络层组成，并将输入的源语言句子编码成对应的隐藏状态，相比于编码器，解码器多了一个编码器解码器的注意力网络，由此解码器可以通过计算目标输入和任意源端单词的注意力权重来获得生成目标单词所需的上下文信息最后输出目标语言的句子。在训练阶段，模型生成目标翻译可以定义为公式(1)所示：

$$Y = D(Z, E(X)) \tag{1}$$

其中$E(.), D(.)$分别对应编码器和解码器，X是输入到编码器中的源语言句子，Z 是输入到解码器中的翻译过程中已经生成的目标单词序列(训练阶段作为监督信号，所以是目标句子y右移一个单词的单词序列)，Y是目标句子。而对应的条件概率计算如公式(2)所示：

$$P(Y|X,Z) = \prod_{t=1}^{T} P(y_t|X,Z;\theta_{mt}) \tag{2}$$

其中$Y = y_1, y_2, \dots, y_T$，T 是目标句子的长度，θ_{mt}是模型参数，那么模型的损失函数可以定义为公式(3):

$$L(X,Y;\theta_{mt}) = -\sum_{(X,Y)\in S} \log P(Y|X,Z) \tag{3}$$

其中S是平行语料集。在预测阶段，模型以自回归的方式生成翻译单词：

$$p(y|x) = \prod_{t=1}^{T} P(y_t|y_{<t}, x) \tag{4}$$

其中y_t是当前时刻生成的目标单词，$y_{<t} = \{y_1, y_2, \dots, y_t\}$是$t$时刻已经生成的单词序列。

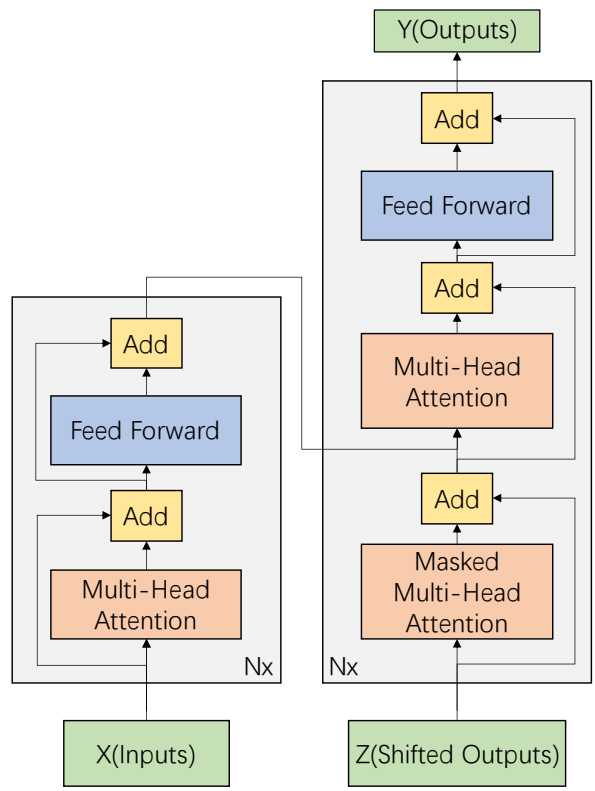

图 1 Transformer 结构示意图

Fig.1 Structure diagram of Transformer

2 相关工作

本节介绍针对神经机器翻译系统正则化技术的相关研究工作。

神经机器翻译系统作为深度神经网络模型包含千万级别的参数，而训练这些参数通常需要大量的训练数据，在低资源的场景下是非常容易出现过拟合现象的，正则化是有效防止过拟合的方法。下面介绍三种在机器翻译中常用的正则化技术：

Dropout Srivastava 等人[7]在 2014 年提出了 dropout 技术，通过随机地把网络中的部分神经元的输出置为 0 来简化网络，从而减少模型对于某些特征的依赖，使得模型的泛化性更强。现在已经成为训练深度网络的通用技术。

Label smoothing Christian 等人[8]在 2016 年提出标签平滑的方法，通过对多分类任务中的监督标签进行加噪，减少真实标签类别在计算损失函数时的权重，这样模型就不会过度地向正向标签和负向标签插值最大的反向学习，尤其是训练数据较少的情况下，是能够有效抑制过拟合的。

正则化技术的主要思想就是通过干扰策略来简化模型或者限制模型对于训练数据的过度学习。传统的正则化技术则更多的是通过给模型增加约束，在考虑模型性能的同时，尽可能选择简单的模型，即对模型的复杂度进行了约束，比如 L2 正则化项，简单模型的参数矩阵应该符合稀疏性，比如 dropout，通过随机地将部分神经元的输出置为 0 得到相比于原来复杂网络的简单网络结构，从而防止网络过度拟合数据，提升模型的泛化能力。这样的方法往往更具通用性，但是还有另外一类正则化技术，在不削弱网络本身建模能力的前提下通过干扰模型的输入数据或者监督信号来抑制模型过度拟合训练数据的方法，如上文提到的标签平滑技术，Cheng 等人[9]也在 2018 年提出的在模型输入上添加小扰动来进行对抗稳定训练；Wang 等人[10]也提出通过对模型源端和目标端输入句子的单词位置采样后进行随机单词替换的方法来进行扰动的方法与本文研究的词级别正则化技术类似，但是相比于随机单词替换的扰动，本文对此有更多的考虑并提出了三种扰动策略，具体细节见第 3 节。

本文研究的词级别的正则化技术是扰动 NMT 的编码器和解码器两端输入句子的方法，考虑到文本数据离散的特性，句子中的每个单词都具有其语义角色，相比于把句子看作一个整体进行句子的整体干扰，单词级别的干扰粒度更小，扰动更加灵活，并且扰动更有针对性。其次，词级别的扰动对于两端的影响各不相同。对于编码端而言，对输入长度为n的句子进行随机的单词采样扰动，则模型会见到2^n个不同的句子，句子也因为扰动导致信息细节的减少；对于解码端而言，解码端的输入句子在训练阶段是作为监督信号而存在的，即 teacher forcing[12]的强约束，模型预测的单词必须严格对照目标句子的每一个单词，这个约束可以保证模型的性能并且加快模型的收敛，但是也驱使模型过度地拟合目标句子而导致对测试数据的泛化不够好，所以对编码端输入句子的干扰可以削弱这一约束，这一点和标签平滑使用软标签来代替硬标签的思想是相似的。

3 词正则化

本节介绍词级别正则化的整体框架和三种扰动策略：采样噪声干扰(Sample Noise Perturbation，SNP)，采样相似词替换(Sample Synonyms Replacement, SSR)，采样遮罩(Sample Unk Mask，SUM)。词级别正则化的目标是通过扰动策略对 NMT 编码端和解码端的输入句子造成干扰，其总体的框架示意图如图 2 所示。

对于模型的输入句子$X = x_1, x_2, \dots, x_n$，定义一个句子长度n的概率向量r服从概率为p的多元伯努利分布，跟句子长度是无关。在训练过程中，如果单词对应的r_w为 1，则对该单词进行正则化操作，如果为 0，则不执行任何操作，解码端输入Z的操作同理。计算过程如下公式(5)和公式(6)：

$$r_w \sim Bernoulli(p) \tag{5}$$

$$\hat{X} = WR(x_i, r_w) = \begin{cases} x_i, & if\ r_w = 0 \\ \hat{x}_i, & if\ r_w = 1 \end{cases} \quad (6)$$

其中$WR(.,.)$根据r_w取值对单词x_i进行扰动。值得注意的是，我们的方法仅在训练阶段对输入文本进行正则化，而在预测阶段没有任何改动。下面具体介绍三种产生扰动单词$\hat{x}_i$的扰动策略。

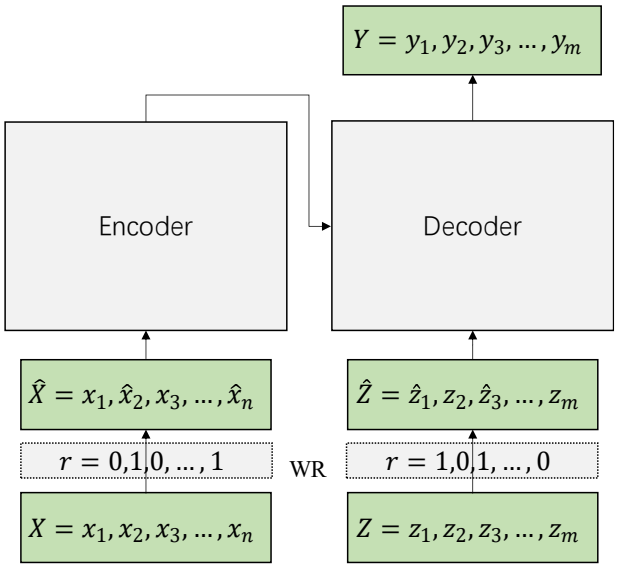

图 2 词正则化示意图

Fig.2 Structure diagram of WR

3.1 SNP

大量文献表明在神经网络中加入随机噪音是减轻过拟合、提升泛化能力的有效方法[13],[14],[15]。由于本文关注的是词级别的正则化方法，因此我们考虑在输入单词的词嵌入上加入噪音来模拟输入扰动：

$$e(\hat{x}_i) = e(x_i) + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I) \quad (7)$$

其中$e(x_i)$代表单词x_i的词嵌入，向量ϵ是以均值为0，标准差为σ采样的高斯噪音，$e(\hat{x}_i)$是加噪后的词嵌入。使用这个方法既可以限制输入文本的信息量，又可以保留大部分语义信息。但是由于深度神经网络强大的建模能力，其本身具有很强的抗干扰能力，因此加入高斯噪音对模型产生的扰动较小。

3.2 SSR

第二种产生扰动单词的方法是选择原单词的相似词进行替换。不同于利用相似词替换扩充语料的方法，我们在训练的过程中动态地进行采样和替换，替换候选词也随着参数更新的过程不断变化。给定单词x_i，我们计算x_i与词表中其它单词的余弦距离作为相似度：

$$\hat{x}_i = Sample_{uniform}(Top\_k(Similarity(x_i))) \quad (8)$$

$$Similarity(x_i) = \frac{exp\{cos(e(x_i),\ e(x))\}}{\sum_{x \in V \setminus x_i} exp\{cos(e(x_i),\ e(x))\}} \quad (9)$$

其中$cos(e(x_i),\ e(x))$衡量x_i与x之间余弦相似度，$V \setminus x_i$是去除x_i的词表，由于词表大小一般是几万，在这样大的空间内采样相似词不确定性很高，因此我们计算相似度最高的前k个候选词，然后根据均匀分布采样一个候选词进行替换。SSR 采样与候选词列表如表 1 和表 2 所示：

表 1 SSR 示例

Tab.1 Example of SSR

原句:his car was still running in the dri@@ ve@@ way
采样:his car was still running in the dri@@ ve@@ way
替换:my car is still running in the dri@@ ve@@ ways

表 2 候选词列表示例

Tab.2 Example of candidates list

采样单词	his	was	way
候选词 ($k = 9$)	my	were	manner
	our	been	path
	their	is	road
	him	came	ways
	your	became	process
	its	wasn	paths
	he	had	avenue
	gement	remains	method
	Her	does	course

可以看出通过本方法计算的相似词与原词的关联性较高，替换之后不会对句子的句法结构造成很大影响。但是会出现个别不相关的干扰词，如：候选词 gement 与 his 完全不相似。

3.3 SUM

第三种产生扰动单词的方法是用$< unk >$标识对原单词进行掩码。高斯噪音和相似词替换都

面临着搜索空间大、扰动不确定性高的问题，因此我们考虑了一种更为软性的扰动策略：

$$\hat{x}_i = Replace(x_i, < unk >) \quad (10)$$

其中$< unk >$是机器翻译模型用于替换未登录词的特殊符号，我们使用该符号替换被采样的单词。该方法受到掩码语言模型[16],[17]的启发，即充分利用句子的上下文信息来表征单词，并且由于$< unk >$作为词表里的单词是一个可学习向量，编码器根据上下文对其进行编码得到中间状态，能够很好地表示句子的信息。不同于掩码语言模型的是，我们掩码输入句子中的部分单词，限制模型对于一些信息过于依赖，达到了词正则化的目的。SUM 策略示例如表所示：

表 3 SUM 示例

Tab.3 Example of SUM

原句:his car was still running in the dri@@ ve@@ way
采样: his car was still running in the dri@@ ve@@ way
掩码:his car was <unk> running in the <unk> ve@@ way

3.4 训练优化目标

词正则化对编码器和解码器的输入进行干扰，减轻过拟合，但是由于干扰程度的随机性和不确定性，容易破坏句子的语义信息。为了使模型学习到更好的句子表征，我们引入对抗的思想，使用一个线性判别器C，对中间状态$H(x)$进行判别是否被正则化，其中H是生成器，生成模型输入对应的中间状态，对应本文中的编码器和解码器。判别器C的目的是区分被正则化和未被正则化的单词，而生成器H的目的则是制造难以被C区分的中间状态。对抗损失的计算方法如公式(11)：

$$L_{adv}(\boldsymbol{x}, \hat{\boldsymbol{x}}; \theta_{mt}, \theta_C) = \sum_{x_i \in \boldsymbol{x}} -\log\ C\big(H(x_i)\big) + \sum_{\hat{x}_i \in \hat{\boldsymbol{x}}} -\log(1 - C\big(H(\hat{x}_i)\big)) \quad (11)$$

其中θ_C是判别器的参数，$\boldsymbol{x}$与$\hat{\boldsymbol{x}}$分别代表原始输入单词和被正则化单词的集合。在训练过程中判别器和生成器可以相互得到提升，这样 NMT 模型可以获得更好的表征能力，并且即使被干扰，生成的中间表示依然能够保留原句的大部分信息。最终的损失函数是翻译和对抗目标的线性组合，使用参数λ控制两个损失比例，如公式(12)。框架示意图如图 3 所示。

$$L(\theta_{mt}, \theta_C) = L_{mt}(\theta_{mt}) + \lambda\, L_{adv}(\theta_{mt}, \theta_C) \quad (12)$$

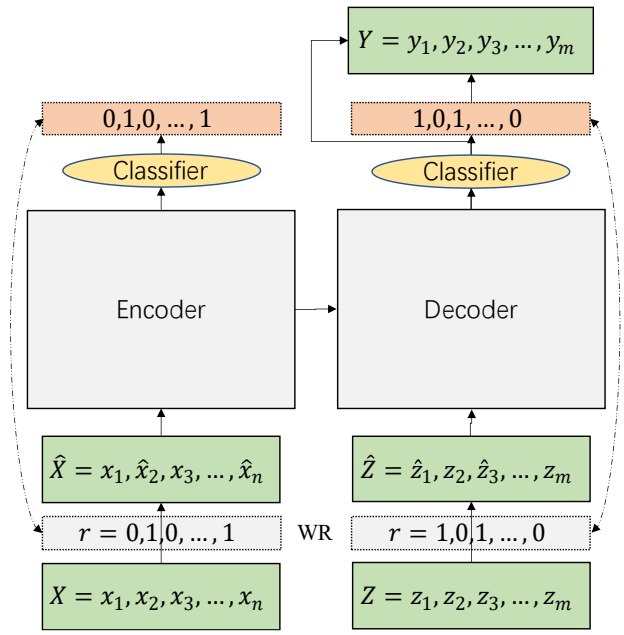

图 3 对抗优化目标示意图

Fig.3 Structure diagram of adversarial optimization

4 实验和结果

为了验证词正则化方法的有效性，我们选择标准的 Transformer 模型在低资源数据集（2018 Third Conference on Machine Translation，WMT'18[1]）上进行英语-土耳其语（简称英-土）和土耳其语-英语（简称土-英）的实验，以及标准数据集语言数据联盟（Linguistic Data Consortium, LDC）上进行中文-英语（简称中-英）的实验；此外，本文也在 12 层的 Transformer 模型上进行了词正则化方法的有效性验证。

[1] http://www.statmt.org/wmt18/

4.1 实验数据

对于中-英翻译系统，我们使用 LDC 的中-英平行语对，其中训练集包含 125 万句，我们使用 NIST06（1664 句）作为验证集，使用 NIST02、NIST03、NIST04、NIST05、NIST08（分别包含平行句对 878、919、1788、1082、1357 句）作为测试集。为了获得较强的基准系统，我们采用了双字节编码（Byte Pair Encoding，BPE）技术限制词表的大小，其中中文词表为 4.2 万，英文词表为 3.1 万。

对于英-土和土-英翻译系统，我们采用了 WMT'18 土-英平行语对，其中训练集包括 21 万句，验证集和测试集分别是 newstest2016 和 newstest2017。语料同样使用 BPE 处理，此外由于英语和土耳其语是相近语言，所以我们使用联合词表进行训练，词表大小为 4.9 万。

4.2 实验参数

实验采用的是基于 Pytorch 实现的 *fairseq*[18] 框架，使用 Transformer 作为我们的基准系统，其结构包含 6 层编码器和 6 层解码器，前馈层和中间层分别为 512 维和 2048 维。对于正则化技术，我们使用 dropout = 0.3 以及使用标签平滑$ls = 0.1$。对于本文提出的词级别正则化方法，源端和目标端采样概率分别为$p_s = 0.1$和$p_t = 0.3$，SNP 策略使用高斯噪音的标准差$\sigma = 1$，SSR 策略的候选词数量$k = 20$。我们在三种正则化策略上都应用了对抗训练，如 3.4 节所示，其中 SNP 与 SSR 的$\lambda = 1$，SUM 的$\lambda = 0.01$。

LDC 中英的测试集包含四个参考译文，因此本文使用*multibleu.pl*测试其双语评估（Bilingual Evaluation Understudy, BLEU）分数，而对于 WMT'18 土-英，则使用 SacreBLEU 计算 BLEU 分数。在解码时，集束搜索的大小均设为 10。

4.3 主要结果

在小数据集英语和土耳其语的实验中，模型在应用不同的正则化策略后都有不同程度的提升，其中 SUM 策略的提升最为明显，分别在英-土上和土-英上分别有分 0.98 分和 1.55 分的提升，其结果如表 4 所示。而在中英任务上，三种策略同样给系统带来了不同程度的提升，尤其是 SUM，获得了 1.42 分的提升，其结果如表 5 所示。因此可以得出结论：三种正则化策略对于低资源或者标准资源场景下都可以给模型带来不同程度的提升，其中 SUM 和 SSR 策略的提升最为明显。

表 4 不同正则化策略在英-土和土-英任务上的 BLEU 值

Tab.4 BLEU scores on EN-TR and TR-EN using different Word Regularization strategies

实验系统	英-土	土-英
Baseline	22.35	19.49
SNP	22.43	20.08
SSR	22.64	20.56
SUM	**23.33**	**21.04**

而在 12 层的 Transformer 模型上的实验结果如表 6 所示，相比于 6 层的 Transformer 模型，需要训练的参数更多，但是数据是小数据集，所以提升没有 6 层 Transformer 明显，但是 SUM 策略依然能够带来一些提升这是很惊喜的。

表 5 不同正则化策略在中-英任务上的 BLEU 值

Tab.5 BLEU score on ZH-EN task using different WR strategies

实验系统	nist02	nist03	nist04	nist05	nist08	AVG
Baseline	47.78	46.74	47.70	47.18	38.11	45.50
SNP	47.40	46.92	48.17	47.65	38.57	45.74
SSR	48.68	**48.29**	48.87	48.71	**39.90**	46.89
SUM	**48.69**	48.14	**49.16**	**48.74**	39.86	**46.92**

表 6 不同正则化策略在 12 层 Transformer 的 BLEU 值

Tab.6 BLEU scores on Transformer with 12 layer using different Word Regularization strategies

实验系统	英-土	土-英
Baseline	22.09	19.46
SNP	22.62	18.72
SSR	22.37	19.99
SUM	**23.06**	**20.50**

5 模型分析

5.1 消融分析

本文旨在设计一种如 Dropout 和 Label smoothing(LS)一样简单且通用的正则化方法，为此本文就这三种正则化方法对模型的影响进行了讨论，在英-土/土-英数据集上的结果如表 8 所示。首先，正则化方法对于模型的性能是必须的，其次，不同正则化方法对模型的性能都有所提升，其中 Dropout 起到最主要的作用，最后，词正则化方法也能够在一定程度上对模型的训练过程起到积极的影响。

5.2 模型泛化分析

困惑度（Perplexity, ppl）是衡量语言模型收敛情况以及模型好坏的指标之一。它的主要思想是通过一句话中所有单词的联合概率来估计这句话的合理性。计算公式如下：

$$ppl(S) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{p(w_i|w_1 w_2 \dots w_{i-1})}} \quad (13)$$

其中$S = w_1, w_2, \dots, w_N$代表一个句子，N是句子长度。模型在给定测试集上的句子能够获得概率越大，则说明模型对于测试集的结果越准确，相应的 ppl 也是越小的，可以有效地反映模型是否出现过拟合。图 4(a)(b)(c)分别展示的就是基准系统和采用词正则化后的模型在训练过程中的损失变化曲线，验证集的 ppl 变化曲线图，以及 BLEU 变化曲线图。相比于 baseline 系统，采用词正则化的模型在训练过程中的损失不会下降到 baseline 水平并且在验证集上获得更低的 ppl，其中 SUM 策略和 SSR 策略随着训练进程的推进并

表 7 Dropout、Label Smoothing、WR 消融实验 BLEU 值

Tab.7 BLEU scores of Dropout, Label Smoothing, WR Ablation experiment

实验系统	英-土	土-英
Baseline	14.88	13.01
+LS	15.66	13.59
+Dropout	20.58	18.31
+Dropout & LS	**22.11**	**19.18**
+WR	17.28	15.56

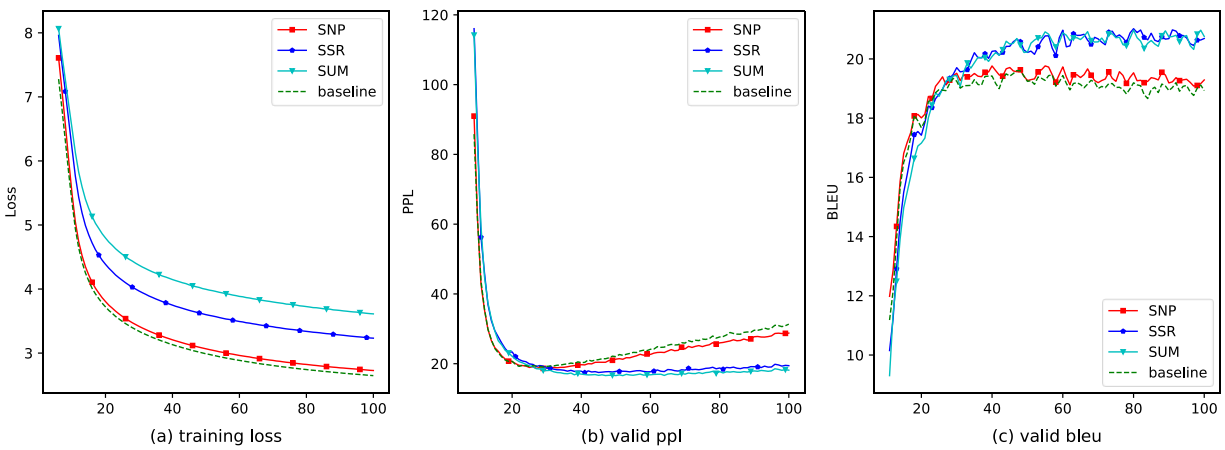

图 4 训练过程中损失 loss、验证集 ppl、验证集 bleu 变化曲线图

Fig.4 Loss, ppl, and BLEU scores over epochs on the TR-EN

没有出现和 baseline 一样的回升趋势，并且 BLEU 值也比高于 baseline，证明我们的模型更不容易出现过拟合的情况，以及模型的泛化能力得到了提升。

5.3 采样概率分析

神经机器翻译模型包含编码器和解码器，编码器端和解码器端的采样概率p_s，p_t是影响模型性能的重要因素，所以为了分析两端的采样概率对模型的影响，我们进行了不同采样概率的对比实验。

对于源端采样概率p_s，我们固定p_t=0.3，p_s选择在[0,0.05,0.10,0.15,0.20,0.25]的概率区间内进行实验；对于目标端采样概率p_t，我们固定p_s=0.1，p_t选择在[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7]的概率区间内进行实验；以此来观察模型的性能变化，图 6 是不同源端采样概率下各模型的 BLEU 曲线；图 7 是不同目标端采样概率下各模型的 BLEU 曲线。

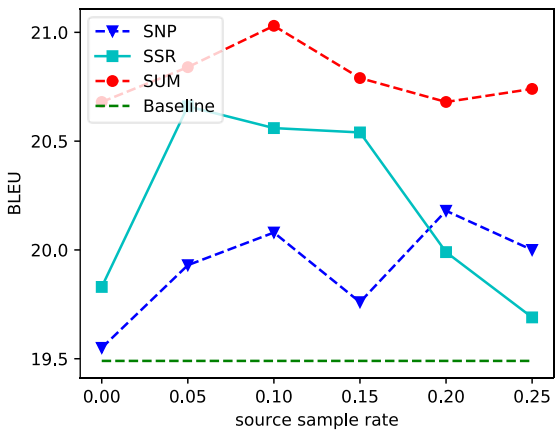

图 6 BLEU 分数随源端采样概率变化曲线

Fig.6 BLEU scores over iterations with different source sample probability

观察图表可以发现：首先，源端对于采样概率的变化相比于目标端更加敏感；其次，采样概率并不是越大越好，模型的性能都随着目标端和源端概率的增加呈现先增后降得趋势；对于每个正则化策略，最佳的采样概率也不一样，三者对于模型的干扰程度也决定了最佳采样概率的峰值，其中 SUM 策略随采样概率增加呈现最明显的下降趋势，所以在三种正则化策略中 SUM 的干扰程度是最大的。

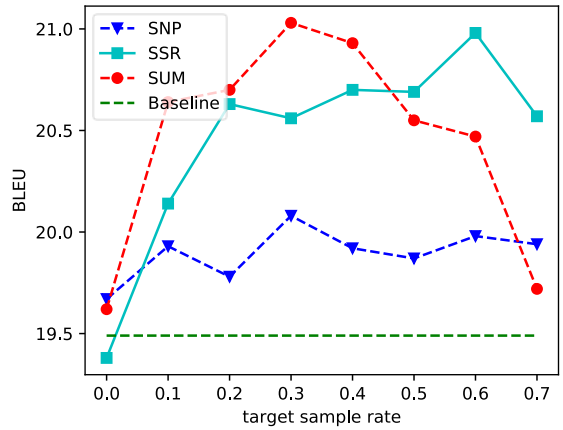

图 7 BLEU 分数随目标端采样概率变化曲线

Fig.7 BLEU scores over iterations with different target sample probability

5.4 优化目标分析

在训练的过程中，本文应用了对抗损失目标，见公式(8)，为了分析对抗损失对模型性能的影响，我们进行了含有对抗目标的模型和不含对抗目标的模型的对比实验，结果如图 8 所示。可以看出在三个正则化策略上应用对抗优化目标都获得了 BLEU 分数的提升，证明了该损失目标的有效性。其中 SSR 受到对抗目标的影响最大，BLEU 分数差值在 0.7 分左右，而 SNP 和 SUM 相较于没有使用对抗损失的模型提升了大约 0.2 BLEU 分数。

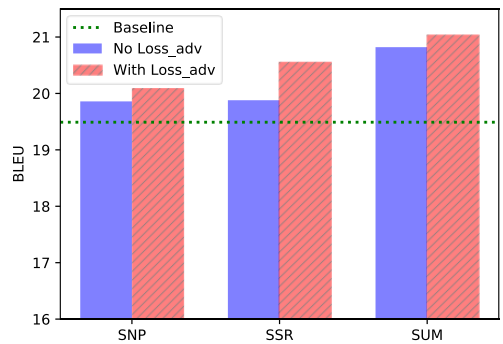

图 8 对抗优化目标的影响

Fig.8 Impact of adversarial objective

6 总结

本文根据神经机器翻译模型对于未知数据的泛化能力不足问题提出了词级别的正则化技术，并在中-英数据集和英语-土耳其语数据集上进行了相关实验。实验结果表明，词正则化能够提升模型的泛化能力，有效防止过拟合，并提升模型性能。在未来的工作中，将更多考虑采样策略以及对特定单词的针对性干扰。

参考文献

[1] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]. Advances in neural information processing systems. 2014: 3104-3112.

[2] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[3] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in Neural Information Processing Systems. 2017: 5998-6008.

[4] 王怡君. 面向有限平行语料资源的神经机器翻译方法研究[D].中国科学技术大学,2019:19-20.

[5] 蔡子龙,杨明明,熊德意.基于数据增强技术的神经机器翻译[J].中文信息学报,2018,32(07):30-36.

[6] Miceli Barone, Antonio Valerio, Haddow Barry, et al. Regularization techniques for fine-tuning in neural machine translation[J]. Association for Computational Linguistics.2017.10.18653/v1/D17-1156: 1489—1494.

[7] Srivastava Nitish, Hinton Geoffrey, Krizhevsky Alex, , et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. The Journal of Machine Learning Research.2014.15:1929-1958.

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, et al. Rethinking the inception architecture for computer vision[C].Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.

[9] Cheng Yong, Tu Zhaopeng, Meng Fandong, et al. Towards robust neural machine translation[J]. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018: 1756—1766.

[10] Xinyi Wang, Hieu Pham, Zihang Dai, Graham Neubig. SwitchOut: an Efficient Data Augmentation Algorithm for Neural Machine Translation[J]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 856–861.

[11] Krogh A, Hertz J A. A simple weight decay can improve generalization[C].Advances in neural information processing systems. 1992: 950-957.

[12] Wen Zhang, Yang Feng, Fandong Meng, et al. Bridging the Gap between Training and Inference for Neural Machine Translation[C]. Association for Computational Linguistics.2019.10.18653/v1/P19-1426:4334-4343.https://www.aclweb.org/anthology/P19-1426

[13] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. Journal of machine learning research, 2010, 11(12).

[14] Neelakantan A, Vilnis L, Le Q V, et al. Adding gradient noise improves learning for very deep networks[J]. arXiv preprint arXiv:1511.06807, 2015.

[15] An G. The effects of adding noise during backpropagation training on a generalization performance[J]. Neural computation, 1996, 8(3): 643-674.

[16] Xipeng Qiu, Tianxiang Sun, Yige Xu, et al. Pre-trained Models for Natural Language Processing: A Survey[J]. arXiv preprint arXiv:2003.08271, 2020.

[17] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[18] Ott M, Edunov S, Baevski A, et al. fairseq: A fast, extensible toolkit for sequence modeling[J]. arXiv preprint arXiv:1904.01038, 2019.

Word level regularization for Neural Machine Translation

QIU Shigui, ZHANG Huaao, DUAN Xiangyu, ZHANG Min

(College of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China)

Abstract: Neural machine translation (NMT) model leverages large scale artificial neural network advanced in superior translation quality. However, NMT model lacks of generalization and prone to overfit especially in low resource scene. For this reason, we propose Word Regularization (WR), which prevents overfitting and improves generalization for unknown data by perturbating input words of sentences. We conduct experiment on LDC Chinese-English and WMT'18 Turkish-English task based on Transformer model. Result shows that our approach is effective on reducing overfitting and gains prominent improvement on translation quality.

Keywords: neural machine translation, generalization, overfitting, regularization