北京大学学报(自然科学版) 第 58 卷 第 1 期 2022 年 1 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 58, No. 1 (Jan. 2022)
doi: 10.13209/j.0479-8023.2021.109

基于跨语种预训练语言模型XLM-R的神经机器翻译方法

王倩 李茂西† 吴水秀 王明文

江西师范大学计算机信息工程学院, 南昌 330022; † 通信作者, E-mail: mosesli@jxnu.edu.cn

摘要 探索将 XLM-R 跨语种预训练语言模型应用在神经机器翻译的源语言端、目标语言端和两端, 提高机器翻译的质量。提出 3 种网络模型, 分别在 Transformer 神经网络模型的编码器、解码器以及两端同时引入预训练的 XLM-R 多语种词语表示。在 WMT 英语–德语、IWSLT 英语–葡萄牙语以及英语–越南语等翻译中的实验结果表明, 对双语平行语料资源丰富的翻译任务, 引入 XLM-R 可以很好地对源语言句子进行编码, 从而提高翻译质量; 对双语平行语料资源匮乏的翻译任务, 引入 XLM-R 不仅可以很好地对源语言句子进行编码, 还可以对源语言端和目标语言端的知识同时进行补充, 提高翻译质量。

关键词 跨语种预训练语言模型; 神经机器翻译; Transformer 网络模型; XLM-R 模型; 微调

Neural Machine Translation Based on XLM-R Cross-lingual Pre-training Language Model

WANG Qian, LI Maoxi†, WU Shuixiu, WANG Mingwen

School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022;
† Corresponding author, E-mail: mosesli@jxnu.edu.cn

Abstract The authors explore the application of XLM-R cross-lingual pre-training language model into the source language, into the target language and into both of them to improve the quality of machine translation, and propose three neural network models, which integrate pre-trained XLM-R multilingual word representation into the Transformer encoder, into the Transformer decoder and into both of them respectively. The experimental results on WMT English-German, IWSLT English-Portuguese and English-Vietnamese machine translation benchmarks show that integrating XLM-R model into Transformer encoder can effectively encode the source sentences and improve the system performance for resource-rich translation task. For resource-poor translation task, integrating XLM-R model can not only encode the source sentences well, but also supplement the source language knowledge and target language knowledge at the same time, thus improve the translation quality.

Key words cross-lingual pre-training language model; neural machine translation; Transformer neural network; XLM-R model; fine-tuning

近年来, 预训练上下文语言模型(如 ELMo[1]、BERT[2]和 GPT[3–4]等)在自然语言处理领域引起越来越多的关注。它们在大量未标注的语料上进行预训练, 获得通用的语言表示, 然后应用到下游任务中, 并根据任务的特点进行微调[5]。这种预训练加微调的方式不仅极大地提升下游任务的性能, 而且大幅度地降低下游任务所需标注语料的规模[6]。

通常, 有监督的机器翻译仅利用双语平行语料进行训练, 会导致大规模的单语语料不能被充分利用。为了将在大规模单语语料上训练获取的通用语言知识应用于机器翻译中, 一些学者提出利用微调或知识蒸馏等方法, 将 BERT 预训练上下文语言模

国家自然科学基金(61662031)资助
收稿日期: 2021–06–12; 修回日期: 2021–08–09

型应用于神经机器翻译源语言端辅助源语言句子编码, 或应用于目标语言端指导译文生成[7−14]。但是, 这些方法仅在神经机器翻译模型的一端(源语言端或目标语言端)使用BERT(或 mBERT[15])预训练上下文语言模型, 而未在两端同时使用。近年, 跨语种预训练语言模型XLM[16]和XLM-R[17]蓬勃发展。与BERT(mBERT)相比, XLM和XLM-R模型在多种语言间共享词表, 在同一嵌入空间对多种语言的词语进行编码, 并针对多语言环境进行优化, 在多项多语言理解任务中的应用刷新了相应任务的最好性能记录。

受上述工作启发, 本文尝试将XLM-R跨语种预训练语言模型引入机器翻译模型中, 进一步提高翻译的质量。本文提出3种网络模型, 将XLM-R模型应用在当前主流的神经机器翻译框架Transformer[18]中。这3种网络模型如下: 1) 在源语言端引入XLM-R模型, 对待翻译的句子进行编码, 替代Transformer编码器; 2) 在目标语言端引入XLM-R模型, 通过额外的解码器模块(包括注意力机制和前馈神经网络), 与源语言端信息进行关联; 3) 在源语言端和目标语言端同步引入XLM-R模型。本文还对比3种模型优化策略对系统性能的影响, 包括冻结XLM-R模型参数的训练方法、在冻结XLM-R模型参数训练的基础上再进行微调的方法以及直接优化整个模型参数的方法。

1 相关工作

如何将预训练模型整合到机器翻译中, 前人的工作主要分为两类。

一类是探索如何充分利用BERT预训练上下文语言模型, 辅助机器翻译。Imamura等[7]直接使用BERT模型作为神经机器翻译的编码器, 并提出两阶段训练策略来减轻预训练模型的灾难性遗忘问题。Weng等[10]、Yang等[11]和Chen等[12]提出使用知识蒸馏技术, 将BERT模型预训练知识迁移到神经机器翻译的编码器或者解码器中。Zhu等[13]提出BERT融合模型, 先使用BERT模型提取输入句子的表示, 然后通过额外的注意力模块, 将BERT模型表示与机器翻译系统中编码器和解码器的每一层融合。Guo等[14]提出并设计不同的轻量级神经网络组件, 插入BERT模型的每一层(如前馈神经网络模块和注意力模块等), 将预训练参数和特定任务的参数解耦, 从而绕过灾难性遗忘问题, 同时引入并行序列解码算法Mask-Predict, 以便充分利用BERT模型, 保持训练和解码过程的一致性。

由于预训练上下文语言模型通常针对语言理解任务而设计——使用遮挡语言模型进行建模, 与机器翻译自回归方式(从一端逐步生成目标语言词语的下一词)预测任务存在差异, 因此第二种方法旨在设计适用于机器翻译的自回归式预训练模型。Song等[19]提出MASS预训练模型, 它是一个基于Transformer的序列到序列单语预训练框架, 其中编码器将带有随机遮挡单词(几个连续标记)的句子作为输入, 解码器则根据编码器的表示来预测这些被遮挡单词, 其输入是编码器中被遮挡的单词, 该模型显著地提升了无监督机器翻译的性能。Lewis等[20]提出BART预训练模型, 其架构与MASS相同, 但训练方式有所不同, 编码器输入被破坏的文本(使用5种噪声函数对文本进行破坏), 解码器根据编码器的表示来恢复原始文本, 该模型在语言理解和文本生成任务中都取得较好的结果。Liu等[21]提出mBART多语言预训练模型, 旨在将BART应用于多种语言的大规模单语语料库, 其模型架构和预训练方式与BART相同, 该模型能够在句子级和文档级别上显著地改善有监督和无监督的机器翻译。

本文与上述工作不同, 我们分别在Transformer的编码器、解码器以及两端同时引入最新的XLM-R跨语种预训练上下文语言模型, 通过XLM-R语言模型初始化表示源语言句子或目标语言句子中的词语, 使用适用的网络结构提高机器翻译的质量。

2 背景知识

2.1 Transformer网络模型

Transformer模型采用编码器–解码器架构(Encoder-Decoder), 其中编码器和解码器均由6个堆叠的编码器层和解码器层组成。编码器将输入序列$X=(x_1, x_2, \ldots, x_n)$抽象成源语言句子的中间表示张量$\boldsymbol{Z}=(z_1, z_2, \ldots, z_n)$, 解码器根据$\boldsymbol{Z}$, 以自回归的方式从左向右逐步生成目标语言句子$Y=(y_1, y_2, \ldots, y_m)$, 计算公式如下:

$$P(Y|X;\theta)=\prod_{i=1}^{m}P(y_i|X,y_{<i};\theta), \tag{1}$$

其中, θ为模型的未知参数, 在双语平行语料上训练获取。

2.2 XLM-R跨语种预训练语言模型

XLM-R跨语种预训练语言模型是在Common

Crawl大型语料上过滤的2.5 TB文本数据上训练形成，支持100种语言。其网络上层采用Transformer编码器架构(层数为12或24)，因此它与Transformer模型具有天然的兼容性，可以方便地引入神经机器翻译中。

XLM-R模型的架构如图1所示，与一般预训练上下文语言模型的差异表现在以下3个方面。1) 它的输入是任意数量的句子组成的文本流(同种语言)，而不是两个句子组成的文本对(如BERT模型)；2) 训练时，每一步涵盖所有语言，每种语言为一个批次；3) 它的训练目标是多语种遮挡语言模型，根据当前词的上下文预测当前词，类似完型填空任务，与机器翻译任务中目标语言句子词语的从左向右自回归生成方式不同。

3 引入XLM-R知识的Transformer网络模型

为了引入XLM-R模型在多种语言文本的大规模语料上训练获取的单语知识，本文提出3种方式改进传统的Transformer模型，在编码端、解码端以及两端逐步引入源语言句子的XLM-R模型和目标语言句子的XLM-R模型，并引入源语言句子和目标语言句子的XLM-R模型，分别简称为XLM-R_ENC模型、XLM-R_DEC模型和XLM-R_ENC&DEC模型。

3.1 XLM-R_ENC模型

XLM-R模型采用Transformer编码器的结构对文本进行抽象表示，其输入文本和输出张量格式与Transformer编码器相同。为了将源语言端预训练的XLM-R模型引入Transformer编码器，我们尝试过两种方式：1) 将XLM-R模型作为特征提取器放在Transformer编码器的底部，用来初始化表示源语言句子中的词语；2) 用XLM-R模型替代Transfomer编码器。第一种方式不仅扩大了模型的规模，增加训练成本，且容易造成预训练知识的灾难性遗忘。因此，本文采用第二种方式，改进的编码器结构如图2左侧所示，解码器采用原始的Transformer解码器结构，改进的编码器形式化表示如下：

$$E^0 = \mathrm{XEmbeddings}(X), \tag{2}$$

$$H^t_{\mathrm{xe}} = \mathrm{LN}(E^{t-1} + \mathrm{MH}(E^{t-1}, E^{t-1}, E^{t-1}))\ (t \geq 1), \tag{3}$$

$$E^t = \mathrm{LN}(H^t_{\mathrm{xe}} + \mathrm{FFN}(H^t_{\mathrm{xe}})), \tag{4}$$

其中，XEmbeddings(·)表示XLM-R模型的嵌入函数，包含词嵌入和位置嵌入，X是源语言句子经过子词切分后的输入；MH(·)表示多头自注意力函数；E^t表示XLM-R模型第t层的输出(t≥1)；LN(·)表示层归一化函数；FFN(·)表示前馈神经网络；H^t_{xe}表示XLM-R模型第t层多头自注意力层的输出。通过式(2)~(4)得到XLM-R模型的最终输出$\boldsymbol{E}^N$，Transformer解码器根据源句子的中间表示张量$\boldsymbol{E}^N$，以自回归的方式解码生成目标语言句子。

XLM-R_ENC模型的编码器与原始Transformer编码器的主要区别在于，XLM-R_ENC模型使用预先训练好的XLM-R模型作为编码器，可提供额外的通用知识，并且所有语言统一采用基于一元文法语言模型的子词切分方法[22]对多语种文本进行切分，以便在多语种文本间共享词表。因此，在将XLM-R模型应用于编码端时，使用相同的子词切分方法对源语言句子进行子词切分。

3.2 XLM-R_DEC模型

为了将目标语言端的预训练知识引入神经机器翻译，本文探索将目标语言XLM-R模型引入Transformer解码端。XLM-R模型使用多语种遮挡语言模型进行训练，其多头注意力中的词语遮挡矩

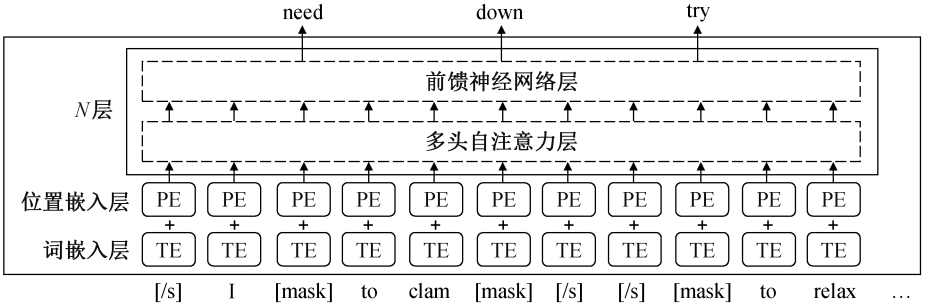

图1　XLM-R模型架构

Fig. 1　Model architecture of XLM-R

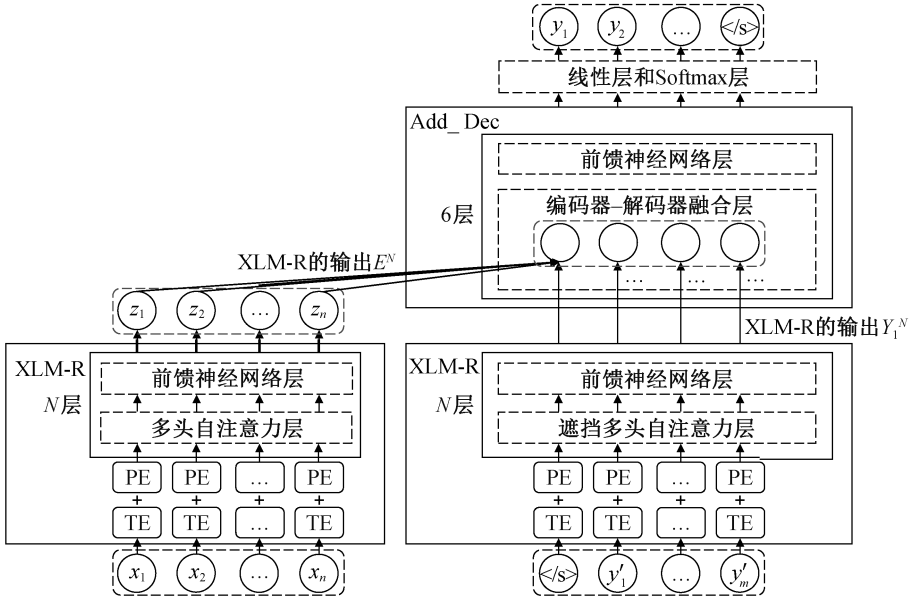

图 2 XLM-R_ENC&DEC 模型架构

Fig. 2 Model architecture of XLM-R_ENC&DEC

(a) XLM-R的遮挡方式 (b) 适用于翻译的遮挡方式

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1

图 3 两种不同的遮挡方式

Fig. 3 Two different masked methods

阵如图 3(a)所示。我们用数字 1 表示信息可见，0 表示信息不可见。在神经机器翻译中，翻译当前词时只能看到前面已经翻译的词语，不能看到未翻译的词语，因此对 XLM-R 模型中的遮挡矩阵进行修改(图 3(b))，以便模拟翻译时从左向右自回归的生成译文中词语。

在模型架构方面，我们尝试直接使用改进遮挡方式的 XLM-R 模型作为解码器；或者在其基础上引入源语言信息的 XLM-R 模型作为解码器，如图 2 右侧所示，在 XLM-R 模型顶部构建额外的 6 层解码器子网络 Add_Dec，包括编码器–解码器融合层和前馈神经网络层，以便将目标语言句子知识与源语言句子知识关联。初步实验结果表明，在解码时关联源语言信息可以更好地生成译文，故采用第二种方式改进解码器结构，编码器则采用原始的 Transformer 编码器结构。改进的解码器形式化表示如下：

$$Y_1^0 = \mathrm{XEmbeddings}(Y'), \tag{5}$$

$$H_{\mathrm{xd}}^t = \mathrm{LN}(Y_1^{t-1} + \mathrm{MMH}(Y_1^{t-1}, Y_1^{t-1}, Y_1^{t-1})) \ (t \geq 1), \tag{6}$$

$$Y_1^t = \mathrm{LN}(H_{\mathrm{xd}}^t + \mathrm{FFN}(H_{\mathrm{xd}}^t)), \tag{7}$$

$$Y^0 = Y_1^N, \tag{8}$$

$$H_{\mathrm{xed}}^t = \mathrm{LN}(Y^{t-1} + \mathrm{MH}(Y^{t-1}, T^N, T^N)) \ (t \geq 1), \tag{9}$$

$$Y^t = \mathrm{LN}(H_{\mathrm{xed}}^t + \mathrm{FFN}(H_{\mathrm{xed}}^t)), \tag{10}$$

其中，MMH(·)表示遮挡多头自注意力函数，Y'是目标语言句子的子词切分结果，H_{xd}^t表示 XLM-R 模型第 t 层多头自注意力层的输出，Y_1^t 表示 XLM-R 模型第 t 层的输出，Y^t 是 Add_Dec 子网络第 t 层的输出，式(10)中 Y^N 是解码器的最终输出。根据式(5)~ (7)，得到 XLM-R 模型的最终输出 Y_1^N，我们将它作为 Add_Dec 子网络的输入。通过式(9)(即 Add_Dec 子网络中的编码器–解码器融合层)，将 Transformer 编码器编码过的源语言句子 T^N 与目标句子相关联，得到 H_{xed}^t。

3.3 XLM-R_ENC&DEC 模型

为了在源语言端和目标语言端同步引入 XLM-R 模型，我们联合 XLM-R_ENC 模型以及 XLM-R_DEC 模型，同时改进 Transformer 编码器和解码器，模型的整体结构如图 2 所示。Add_Dec 子网络第一个子层编码器–解码器融合层会将经过 XLM-R 模型

编码过的源语言句子与经过XLM-R模型编码过的目标语言句子相互关联, 以便更好地软对齐源语言句子中词语与目标语言句子中词语, 最终生成机器译文。

3.4 模型训练

3个模型均采用多分类交叉熵损失函数作为优化目标, 在双语平行语料上进行训练。由于网络部分子结构的参数权值已将XLM-R模型作为初始值。在进行模型整体参数训练时, 我们探索3种网络参数训练策略: 1) 直接微调(DirectFine-tuning), 即所有的模型参数一起更新, 反向传播, 应用于所有层; 2) 固定XLM-R模型参数(Freeze), 将XLM-R模型视为特征提取器, 不参与翻译任务的训练; 3) 先固定, 再微调(+Fine-tuning), 即先固定XLM-R模型参数, 使用双语平行语料训练剩余的未知参数, 直到模型在验证集上损失最小, 再联合微调所有模型, 即同时更新模型中的所有参数。

除非特殊说明, 本文实验中均采用直接微调的方法优化网络整体参数。后续的消融实验中将对3种参数调整策略进行对比, 用于验证直接微调参数优化策略对系统性能的提升幅度最大。

4 实验

4.1 实验设置

我们分别在双语平行语料资源丰富和资源匮乏的翻译任务中评价本文模型。在资源丰富的任务中采用WMT2014英语–德语语料(WMT14 En-De), 使用newstest2013作为验证集, newstest2014作为测试集。在资源匮乏的任务中采用IWSLT2017英语–葡萄牙语(IWSLT17 En-Pt)和IWSLT2015英语–越南语语料(IWSLT15 En-Vi), 分别使用tst2016和tst2012作为验证集, tst2017和tst2013作为测试集。各任务中训练集、验证集和测试集的语料规模见表1。对于WMT14 En-De和IWSLT15 En-Vi翻译任务, 使用来自斯坦福大学的自然语言处理小组(The Stanford NLP Group)预处理后的语料; 对于IWSLT17 En-Pt翻译任务, 使用开源工具包mosesdecoder(https://github.com/moses-smt/mosesdecoder)中的预处理工具, 对句子使用标点符号规范化、移除非打印字符和标记化等预处理, 所有语料均使用基于一元文法语言模型子词切分方法进行子词切分。

表 1　实验语料规模统计

Table 1　Experimental corpus size statistics

语料	训练集	验证集	测试集
WMT14 En-De	4468840	3000	2737
IWSLT17 En-Pt	171032	1155	1124
IWSLT15 En-Vi	133317	1553	1268

利用开源工具包fairseq[23]实现3种基于XLM-R模型的Transformer网络结构。XLM-R模型使用XLM-Roberta-Base预训练模型, 层数为12, 注意力头数为12, 隐藏层大小为768, 前馈神经网络内置隐藏层大小为3072; Transformer模型和Add_Dec子网络均只使用6层, 隐藏层大小、注意力头数和前馈神经网络内置隐藏层的参数设置与XLM-R模型相同。

对比的基线模型包括Transformer base模型、Transformer big模型[18]和NMT with BERT模型[7]。其中, Transformer base模型的层数为6, 注意力头数为8, 隐藏层大小为512, 前馈神经网络内置隐藏层大小为2048; Transformer big模型层数为6, 注意力头数为16, 隐藏层大小为1024, 前馈神经网络内置隐藏层大小为4096; NMT with BERT模型通过直接用BERT替换Transformer的编码端来引入预训练知识。

用BLEU[24]作为译文评价指标, 利用开源工具mosesdecoder中的脚本multi-bleu.perl进行打分。打分时, 机器译文均进行符号化(tokenize)处理, 并区分大小写。

4.2 实验结果

4.2.1 3个模型性能对比的实验结果

表2给出本文提出的3个模型和对比的基线系统在WMT英语–德语和IWSLT英语–葡萄牙语、英语–越南语等翻译方向上的实验结果。在所有翻译方向上, XLM-R_ENC模型都优于基线模型, 特别是在资源匮乏的翻译任务中, 引入预训练知识能够大幅度提升模型的翻译性能。再对比Transformer base与Transformer big模型可以看出, 当模型的参数量增大时, 其翻译性能并不一定会提升, 进一步说明是预训练知识提升了翻译的性能。对比NMT with BERT模型, 使用在多种大规模单语语料上预训练获取的通用语言知识, 翻译性能优于使用仅在单语语料上预训练获取的通用语言知识。最后, 我们尝试对XLM-R_ENC模型进行集成, 在开发集上取翻译性能最优的5组模型的参数进行平均, 以期进一步提高模型的翻译性能, 集成的结果见表2中

表 2 不同模型的翻译性能对比

Table 2 Comparison of translation performance of different models

模型		BLEU		
		WMT14 En-De	IWSLT17 En-Pt	IWSLT15 En-Vi
对比模型	Transformer base	27.22	34.86	26.12
	Transformer big	28.46	34.70	27.73
	NMT with BERT	28.90	36.56	29.57
本文模型	XLM-R_ENC	**29.07**	**39.22**	**31.39**
	XLM-R_DEC	21.50	29.58	23.97
	XLM-R_ENC&DEC	24.51	37.95	30.98
本文集成模型	$\text{XLM-R\_ENC}_{\text{ensemble}}$	**29.09**	**39.28**	31.44
	$\text{XLM-R\_DEC}_{\text{ensemble}}$	21.10	29.55	24.67
	$\text{XLM-R\_ENC\&DEC}_{\text{ensemble}}$	25.09	38.30	**31.75**

说明: 粗体数字表示在该翻译方向上翻译性能最佳, 下同。

$\text{XLM-R_ENC}_{\text{ensemble}}$一行。

对于仅在解码端引入 XLM-R 模型的翻译方法 XLM-R_DEC, 在所有翻译方向上的性能大幅度劣于基线模型, 可能是 XLM-R 模型的多语种遮挡语言模型的训练目标与 Transformer 的自回归训练目标不同所致。Lample 等[16]在解码端的有效尝试, 并未修改解码端的模型架构, 只是用 XLM 模型预训练好的模型参数去初始化 Transformer 解码端相应的模型参数。

对于在编码端和解码端同步引入 XLM-R 模型的 XLM-R_ENC&DEC 方法, 在资源丰富的 WMT 英语–德语翻译任务中, 其性能并没有得到提升, 而在资源匮乏的 IWSLT 英语–葡萄牙语和英语–越南语翻译任务中, 不论是单系统还是集成系统, 其性能均超过基线模型。这表明对于资源匮乏的翻译任务, 在源语言端和目标语言端同步引入 XLM-R 模型也可以提高翻译质量。我们猜测, 在资源匮乏的翻译任务中, 目标语言端引入的额外通用语言知识可以克服 XLM-R 模型与 Transformer 模型训练目标不一致的弊端, 后续的实验分析中将进一步挖掘这种情况产生的原因。

4.2.2 不同训练方式的实验结果

我们在 WMT 英语–德语以及 IWSLT 英语–葡萄牙语和英语–越南语翻译任务中对比不同参数调整策略下的系统性能, 结果如表 3 所示。在 XLM-R_ENC 和 XLM-R_ENC&DEC 方法中, 对于资源丰富的翻译任务, 直接微调的方法(DirectFine-tuning)与先固定再微调的方法(+ Fine-tuning)性能相当; 对于资源匮乏的翻译任务, 直接微调的方法远远优于先固定再微调的方法。在 XLM-R_DEC 方法中, 先固定再微调的方法优于直接微调的方法, 但是两种训练方式都未能提高翻译性能。因此, 本文实验中均采用直接微调的方法优化网络整体参数。

表 3 不同训练方式对翻译性能的影响

Table 3 Impact of different training methods on translation performance

模型	训练方式	BLEU		
		En-De	En-Pt	En-Vi
XLM-R_ENC	DirectFine-tuning	**29.07**	**39.22**	**31.39**
	Freeze	23.22	29.81	24.11
	+ Fine-tuning	28.90	36.69	29.26
XLM-R_DEC	DirectFine-tuning	21.50	29.58	**23.97**
	Freeze	18.84	27.18	16.76
	+ Fine-tuning	**23.21**	**32.47**	23.23
XLM-R_ENC&DEC	DirectFine-tuning	24.51	**37.95**	**30.98**
	Freeze	9.50	17.61	16.71
	+ Fine-tuning	**24.79**	36.99	30.05

4.2.3 不同层数预训练模型的实验结果

为了比较使用不同层预训练模型对翻译性能的影响, 我们对比两种 XLM-R 模型层数使用策略: 1) 在 3 个模型中使用预训练模型 XLM-R 的全部层(12 层)表示张量; 2) 仅使用其底部 6 层表示张量。在 WMT 英语–德语以及 IWSLT 英语–葡萄牙语和英语–越南语翻译任务中的实验结果如表 4 所示。在源语言端使用 XLM-R 模型全部层的表示或在源语言端和目标语言端同时使用 XLM-R 模型全部层的表示优于使用底部 6 层的表示, 仅在目标语言端使用 XLM-R 模型底部 6 层的表示优于使用全部层的表示, 但仍未提高翻译质量。因此, 本文的默认模型

表 4 不同层数预训练模型对翻译性能的影响

Table 4　Impact of different layers of pre-training models on translation performance

模型	BLEU					
	En-De		En-Pt		En-Vi	
	6 层	12 层	6 层	12 层	6 层	12 层
XLM-R_ENC	27.04	**29.07**	37.26	**39.22**	30.58	**31.39**
XLM-R_DEC	**21.77**	21.50	**30.13**	29.58	**24.36**	23.97
XLM-R_ENC&DEC	23.56	**24.51**	36.43	**37.95**	30.60	**30.98**

表 5　Add_Dec 子网络层数对 XLM-R_ENC&DEC 模型性能的影响

Table 5　Impact of Add_Dec sub-network layers on XLM-R_ENC&DEC model performance

翻译方向	BLEU					
	1 层	2 层	3 层	4 层	5 层	6 层
En-Pt	37.57	37.06	37.76	37.33	37.65	**37.95**
En-Vi	30.59	30.84	**31.20**	30.88	31.15	30.98
集成模型 $\text{En-Pt}_{\text{ensemble}}$	38.15	37.56	**38.62**	37.84	38.17	38.30
集成模型 $\text{En-Vi}_{\text{ensemble}}$	31.30	31.26	31.64	31.60	31.40	**31.75**

设置为使用 XLM-R 模型全部层的表示。

4.2.4 Add_Dec 子网络不同层数的实验结果

表 2 列出的实验结果表明，对于双语平行语料资源匮乏的翻译任务，引入 XLM-R 可以很好地对源语言端和目标语言端知识同时进行补充，提高翻译质量。因此，我们在 IWSLT 英语–葡萄牙语和英语–越南语翻译方向上探索不同 Add_Dec 子网络的层数对 XLM-R_ENC&DEC 模型的影响，实验结果如表 5 所示，使用 3 层或 6 层 Add_Dec 子网络的翻译性能最佳。考虑到在性能相差不大的情况下，3 层的模型参数量会更小，训练速度更快，因此建议在 XLM-R_ENC&DEC 模型解码端仅使用 3 层 Add_Dec 子网络。

4.2.5 实验分析

为了证明在资源匮乏的翻译任务中，源语言端和目标语言端同步引入 XLM-R 模型也能提高翻译质量，我们对 3 个模型在 IWSLT 英语–越南语的翻译任务中生成的译文示例进行分析。表 6 给出一个英语源语言句子及其越南语人工参考译文，以及 3 个模型的翻译结果。通过在双语语料上查找，我们发现源语言句子中 rehabilitates 一词在双语平行语料的英语端没有出现过，但是 XLM-R_ENC&DEC 模型能将其正确地翻译成越南语中的词语 phục hồi，说明这个翻译知识是由 XLM-R 模型引入的。在更多的翻译示例中还发现，尽管某个越南语的词语在双语平行语料的目标端没有出现，但在机器译文中有时也能正确地翻译该词语(示例略)，同样说明这个知识是由 XLM-R 模型引入的。上述分析均说明，在资源匮乏的环境下，在源语言端和目标语言端同时引入 XLM-R 模型，可以将双语语料中没有出现的词语正确地翻译成目标语言中词语，提高了翻译质量。

5 结论

本文探索跨语种预训练语言模型 XLM-R 在神经机器翻译系统 Transformer 中的应用，提出并对比 3 种模型来实现在源语言或目标语言中，利用在多种大规模单语语料上预训练获取的通用语言知识。在多个翻译任务中的实验结果表明，对于资源丰富的翻译任务，XLM-R 模型可以更好地对源语言句子进行编码表示，从而提高翻译质量，但由于 XLM-R 模型的多语种遮挡语言模型的训练目标与 Transformer 模型的自回归训练目标不一致，导致其应用在解码端时不能提高翻译质量；对于资源匮乏的翻译任务，目标端引入额外的通用语言知识可以克服两个模型训练不一致的弊端，促使在源语言端和目标语言端同步引入 XLM-R 模型，也能提高翻译质量。

表 6　不同模型的译文示例对比

Table 6　Comparison of translation examples of different models

源语言句子	译文			
	人工参考	XLM-R_ENC&DEC	XLM-R_ENC	XLM-R_DEC
I met him at a shelter where Free the Slaves **rehabilitates** victims of slavery .	Tôi gặp cậu bé ở khu cứu trợ mà tổ chức Giải phóng Nô lệ **phục hồi** các nạn nhân bị nô lệ .	Tôi đã gặp ông tại một nơi trú ẩn , nơi người Do Thái **phục hồi** những nạn nhân của nạn nô lệ .	Tôi đã gặp anh ấy tại một nơi trú ẩn nơi mà FreeSpeech phục những nạn nhân nô lệ .	Tôi gặp anh ấy ở nơi trú ẩn nơi mà Free xâm phạm nạn nhân của mình .

说明：粗体字示意 XLM-R_ENC&DEC 模型可以将双语语料中没有出现的词语正确地翻译成目标语言中词语。

参考文献

[1] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations // Proceedings of the NAACL-HLT. New Orleans, 2018: 2227–2237

[2] Devlin J, Chang M W, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding // Proceedings of the NAACL-HLT. Minneapolis, 2019: 4171–4186

[3] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training [R/OL]. (2018) [2020–11–05]. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[4] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners // Proceedings of the NeurIPS. Vancouver, 2020: 1877–1901

[5] 翟煜锦, 李培芸, 项青宇, 等. 基于QE的机器翻译重排序方法研究. 江西师范大学学报(自然科学版), 2020, 44(1): 46–50

[6] 黄民烈, 唐杰, 文继荣. 超大规模预训练模型的优势、局限与未来趋势. 中国计算机学会通讯, 2021, 17(2): 88–89

[7] Imamura K, Sumita E. Recycling a pre-trained BERT encoder for neural machine translation // Proceedings of the EMNLP & NGT. Hong Kong, 2019: 23–31

[8] Kim Y, Rush A M. Sequence-level knowledge distillation // Proceedings of the EMNLP. Austin, 2016: 1317–1327

[9] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network [EB/OL]. (2015–03–09) [2020–11–05]. https://arxiv.org/abs/1503.02531

[10] Weng R, Yu H, Huang S, et al. Acquiring knowledge from pre-trained model to neural machine translation // Proceedings of the AAAI. New York, 2020: 9266–9273

[11] Yang J, Wang M, Zhou H, et al. Towards making the most of bert in neural machine translation // Proceedings of the AAAI. New York, 2020: 9378–9385

[12] Chen Y C, Gan Z, Cheng Y, et al. Distilling knowledge learned in BERT for text generation // Proceedings of the ACL. Washington, 2020: 7893–7905

[13] Zhu J, Xia Y, Wu L, et al. Incorporating BERT into neural machine translation [C/OL] // Proceedings of the ICLR. (2020–03–11) [2020–10–20]. https://openreview.net/forum?id=Hyl7ygStwB

[14] Guo J, Zhang Z, Xu L, et al. Incorporating BERT into parallel sequence decoding with adapters [EB/OL]. (2020–08–13) [2020–10–20]. https://arxiv.org/abs/2010.06138

[15] Karthikeyan K, Wang Z, Mayhew S, et al. Cross-lingual ability of multilingual BERT: an empirical study [C/OL] // Proceedings of the ICLR. (2020–03–11) [2020–10–20]. https://openreview.net/forum?id=HJeT3yrtDr

[16] Lample G, Conneau A. Cross-lingual language model pretraining // Proceedings of the NeurIPS. Vancouver, 2019: 7059–7069

[17] Conneau A, Khandelwal K, Goyal N, et al. Unsupervised cross-lingual representation learning at scale // Proceedings of the ACL. Washington, 2020: 8440–8451

[18] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need // Proceedings of the NeurIPS. Long Beach, CA, 2017: 6000–6010

[19] Song K, Tan X, Qin T, et al. MASS: masked sequence to sequence pre-training for language generation // Proceedings of the ICML. Long Beach, CA, 2019: 5926–5936

[20] Lewis M, Liu Y, Goyal N, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension // Proceedings of the ACL. Washington, 2020: 7871–7880

[21] Liu Y, Gu J, Goyal N, et al. Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 2020, 8: 726–742

[22] Kudo T. Subword regularization: improving neural network translation models with multiple subword candidates // Proceedings of the ACL. Melbourne, 2018: 66–75

[23] Ott M, Edunov S, Baevski A, et al. Fairseq: a fast, extensible toolkit for sequence modeling // Proceedings of the NAACL. Minneapolis, 2019: 48–53

[24] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation // Proceedings of the ACL. Philadelphia, 2002: 311–318