

基于多任务学习的词汇约束神经机器翻译方法

叶娜*, 夏宇轩, 张桂平, 杨晨, 王雪妮

(沈阳航空航天大学计算机学院, 辽宁 沈阳 110136)

摘要: [目的] 神经机器翻译(NMT)是指利用神经网络模型将源语言翻译为正确的目标语言,而词汇约束翻译(LCT)要求模型在生成的译文中包含预先指定的目标语言片段.由于NMT模型采用端到端的训练方式,在学习过程中更侧重于序列的整体优化而非个别词汇的强制匹配,这导致现有的LCT方法难以在译文质量与约束准确性间实现平衡. [方法] 为了解决这一问题,本文提出一种基于多任务学习的LCT方法,利用不同子任务的归纳偏置,引导模型在训练过程中获得词汇约束场景下的翻译能力.该方法将LCT作为主任务,将常规机器翻译(MT)、目标端单语文本生成(TMTG)和源端词元类型标注(STTL)作为辅助任务进行多任务学习. [结果] 实验结果表明,本文提出的LCT方法与当前最先进的基线方法相比, BLEU 分数提升 2.28 个百分点, BLEURT 分数提升 0.62 个百分点, 窗口重叠分数提升 1.50. [结论] 本文提出的基于多任务学习的词汇约束 NMT 方法,通过任务间的信息交换与协同作用,可以促进跨任务的知识迁移,在准确生成约束词的同时,保持较高的译文质量和流畅性.

关键词: 神经机器翻译; 词汇约束; 多任务学习; 知识融合

中图分类号: TP 391

文献标志码: A

文章编号: 0438-0479(2025)06-0970-13

A lexically constrained neural machine translation method based on multi-task learning

YE Na*, XIA Yuxuan, ZHANG Guiping, YANG Chen, WANG Xueni

(School of Computer, Shenyang Aerospace University, Shenyang 110136, China)

Abstract: [Objective] Neural machine translation (NMT) refers to using neural network models to translate source language into accurate target language, whereas lexically constrained translation (LCT) requires the generated translation to include pre-specified segments in the target language. The end-to-end training approach employed by NMT models prioritizes overall sequence optimization over the forced alignment of specific lexical items, hence making it challenging for existing LCT methods to strike a balance between translation quality and constraint accuracy. [Method] To address this issue, we propose a multi-task learning-based LCT method that leverages inductive biases from multiple tasks to guide the model in developing translation capabilities for lexical constraint scenarios. This method designates LCT as the primary task and incorporates auxiliary tasks such as NMT, target-side monolingual text generation (TMTG), and source-side token type labeling (STTL) for the multi-task learning. [Results] Experimental results demonstrate that the proposed method achieves improvements over the current state-of-the-art baseline methods, with a BLEU score increase of 2.28, a BLEURT score improvement of 0.62, and a Window-Overlap score increase of 1.50. [Conclusion] By leveraging information exchange and collaborative interactions among tasks, the proposed multi-task learning-based lexical constrained neural machine translation method enhances cross-task knowledge transfer, thus ensuring the accurate generation of constrained terms while maintaining high translation quality and fluency.

收稿日期: 2024-12-11 录用日期: 2025-03-25

基金项目: 国家自然科学基金(U1908216)

* 通信作者: yena_1@126.com

引文格式: 叶娜, 夏宇轩, 张桂平, 等. 基于多任务学习的词汇约束神经机器翻译方法[J]. 厦门大学学报(自然科学版), 2025, 64(6): 970-982.

Citation: YE N, XIA Y X, ZHANG G P, et al. A lexically constrained neural machine translation method based on multi-task learning[J]. J Xiamen Univ Nat Sci, 2025, 64(6): 970-982. (in Chinese)



Keywords: neural machine translation (NMT); lexical constraint; multi-task learning; knowledge fusion

神经机器翻译(neural machine translation, NMT)是指利用神经网络模型将源语言翻译为正确的目标语言,而词汇约束翻译(lexically constrained translation, LCT)要求翻译模型在输出中包含预先指定的单词和短语,这在实际应用中具有重要意义^[1]. 例如,在领域适应机器翻译(MT)中,需要对术语进行约束^[2-4],在交互式 MT 中,需要对用户指定的片段进行约束^[5-6]. 开发适用于 NMT 模型的 LCT 方法,可以使其更好地适应真实场景的翻译任务.

与具有显式对齐和可控规则的统计 MT(statistical MT, SMT)^[7]模型不同,NMT 模型由于其端到端的训练方式和隐式表达^[8-9],在生成过程中直接对翻译模型施加约束变得更为困难.

为了解决这一问题,一些研究者选择修改解码算法,在解码过程中强制选择目标片段,对模型施加硬约束^[1,10-12]. 尽管硬约束方法可以确保输出中包含目标约束,但存在解码速度慢以及译文质量不佳的问题^[6,13-14].

另一些研究者选择对模型施加软约束(soft constraints),通过构建合成数据或修改模型架构,使模型在训练过程中获得词汇约束场景下的翻译能力^[2,15-19]. 使用软约束虽然可以加快解码速度,但在某些情况下生成译文的约束准确性较低,这表明使用单一的软约束方法并不是 LCT 的最佳选择^[13].

最近的研究通过修改模型的输入输出形式,将 LCT 任务转变为基于模板的填充任务^[6,20]. 这类方法虽然可以降低模型的学习难度,但在某些情况下会损伤译文的流畅性与自然度.

现有方法难以在译文质量和约束准确性间实现平衡,为了解决这一问题,本文提出一种基于多任务学习的 LCT 方法. 多任务学习通过联合训练多个相关任务,模型可以在训练过程中同时优化多个目标,实现任务间的知识迁移. 通过融合不同任务的知识,模型在生成译文的过程中可以避免生硬地插入约束片段,实现译文质量与约束准确性间的平衡.

因此,本文在训练 LCT 任务的同时,训练 NMT 任务以提升翻译质量,训练目标端单语文本生成(target-side monolingual text generation, TMTG)任务以提升约束准确性和译文流畅性,训练源端词元类型标注(source-side token type labeling, STTL)任务以增强模型对于双语约束对的理解. 训练阶段,使用以 Transformer 模型^[9]为基础的多任务学习模型联合训练所有子任务,通过参数共享机制,模型在训练过

程中能够建立双语句子间的语义联系、双语约束间的准确对齐,以及双语句对与双语约束对间的交叉关系. 这种知识融合策略不仅增强翻译模型对双语文本的理解,还强化了其利用给定双语约束片段进行精准翻译的能力.

本文的主要贡献如下:

1) 将多任务学习策略应用于 LCT 任务,提出一种基于多任务学习的 LCT 方法,通过参数共享机制实现主任务与辅助任务的知识融合,在提升模型约束准确性的同时,保持较高的译文质量和流畅性.

2) 联合训练 LCT 任务、常规 MT 任务、TMTG 与 STTL 任务,引导模型建立双语句对与双语约束对间的关系,增强了模型对源语言与约束片段的利用能力.

1 相关研究

1.1 词汇约束 NMT

由于 NMT 模型具有端到端的训练特性与隐层表示的不可解释性^[8-9],直接对其施加约束相对困难. 为了解决这一问题,现阶段研究主要集中在下述 3 种策略.

一些研究者选择对模型施加硬约束,通过修改模型的解码算法^[11],在推理过程中强制选择目标约束,实现 LCT. Hokamp 等^[10]提出网格束搜索,通过在解码过程中将约束片段纳入搜索网格,模型能够生成包含指定约束的译文. 然而,GBS 的计算复杂度与受约束片段的数量成线性关系. 为了降低计算开销,Post 等^[1]提出动态束分配(dynamic beam allocation, DBA),通过将固定大小的束分配给满足相同数量约束的候选词,在解码过程中实现束宽度的动态调整,加速解码过程. 为了进一步加快解码速度,Hu 等^[12]提出矢量化 DBA(vectorized dynamic beam allocation, VDBA),通过矢量化操作可以提高解码效率,但与原始的束搜索相比,仍然非常缓慢^[6].

另一些研究选择对模型施加软约束,通过构建合成数据或修改模型架构,引导模型在训练过程中获得词汇约束场景下的翻译能力. Song 等^[15]使用目标约束片段直接替换对应的源语言片段,并使用指针网络增强模型对目标约束片段的复制能力. Dinu 等^[16]将目标约束片段作为源语言句子的内联注释,通过插入目标约束片段的方法构建合成数据,增强了约束片段与对应源语言片段的联系. Chen 等^[21]使用分隔符将

多个约束片段附加到整个源语言句子之后,构建一个包含约束的训练数据集. Jon 等^[22]也使用类似的方法,通过减少一致性错误改善术语翻译的效果. Song 等^[23]提出利用词对齐施加约束,通过引入额外的对齐模块,增强了双语约束片段间的联系. Chen 等^[13]利用输入端与输出端的注意力权重推断对齐信息,通过引入显式对齐模块,实现 LCT. Wang 等^[17]通过矢量化约束片段,将目标约束与对应的源语言片段添加到注意力模块以集成约束. Zhang 等^[18]结合占位符与代码切换,在训练时使用约束标签简化预测序列,在推理过程中将标签替换为目标约束片段,提升 LCT 的鲁棒性. Hauhio 等^[19]将硬约束与软约束结合,使用基于非确定性有限状态转换机的约束识别算法强制使用目标约束片段,并利用回译替换技术调整源语言句子以引导模型生成更准确的翻译. 软约束方法相较于硬约束方法虽然能够加快解码速度,但在某些情况下生成目标约束的成功率较低^[13].

最近的研究选择改变翻译模型的输入输出结构,通过修改任务形式,实现基于模板的 LCT. Xiao 等^[6]受到文本填充任务的启发^[24],通过构建译文模板将 LCT 任务转变为双语文本填充任务,但文本填充的局限性在于约束片段必须依据译文顺序提供. Wang 等^[20]将约束片段与未约束片段抽象为非终结符模板,将 LCT 任务转变为模板生成任务,以此降低模型的学习难度. 构建模板的方法虽然解决了双语文本填充必须依据译文顺序提供约束片段的局限性,但在某些情况下会损伤译文的流畅性.

为了在译文质量与约束准确性间实现平衡,模型应在学习词汇约束能力的同时,增强翻译能力与上下文感知能力. 使用多任务学习方法,模型可以同时学习多个相关任务,利用不同任务间的知识迁移,同时增强词汇约束能力、翻译能力与上下文感知能力.

1.2 多任务学习

多任务学习是机器学习中的一种学习范式,其目的是利用多个相关任务中包含的有用信息来提高每个任务的泛化能力^[25]. 作为一种迁移学习方法,多任务学习已成为 NMT 领域一种常见的方法.

早期多任务学习的主要动机是缓解数据稀疏问题,Johnson 等^[26]通过在多语言翻译模型中使用联合词表及共享参数,保证每种语言的充分表示并增强低资源语言的翻译性能. Imankulova 等^[27]将多任务学习与数据增强技术结合,在极低资源的日-俄翻译任务上产生明显提升.

最近的研究表明,多任务学习模型在某些情况下

可以比单任务模型取得更好的性能^[28]. 一个应用多任务学习的思路是将 NMT 任务设置为主任务,同时添加若干相关的辅助任务,所有任务都可以受益于共享知识^[29],通过多任务学习方法,可以增强 NMT 模型对于语言知识的捕获能力^[30]. 例如, Dong 等^[31]使用单编码器-多解码器结构的多语言 NMT 模型增强编码器的表示能力,提升了模型整体的翻译性能.

虽然多任务学习方法在常规的 MT 任务中已有应用,但在 LCT 任务中尚未得到实践. 受到上述工作的启发,为了提升模型在词汇约束场景下的译文流畅性与约束准确性,本文提出基于多任务学习的 LCT 方法,通过多任务学习的参数共享机制可以帮助模型理解任务间的共性,取得译文质量与约束准确性间的平衡.

2 方法

2.1 任务定义

LCT 任务要求模型根据源语言句子与一组双语约束对生成包含指定约束片段的正确译文. 任务示例如图 1 所示.

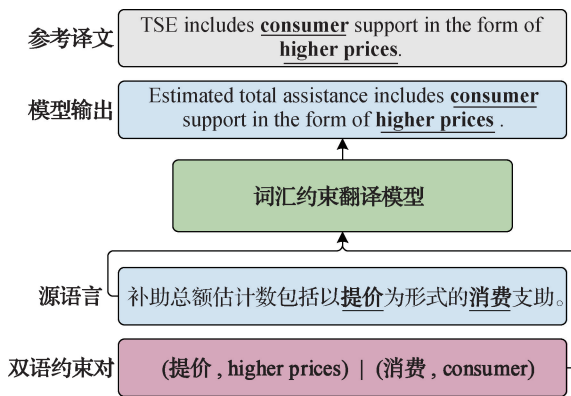


图 1 LCT 任务示例

Fig. 1 An example of LCT task

形式上,给定包含 M 个词元(token)的源语言句子 $\mathbf{x} = [x_1, x_2, \dots, x_M]$ 、包含 N 个词元的目标语言句子 $\mathbf{y} = [y_1, y_2, \dots, y_N]$ 与包含 K 个约束对的集合 $C = \{(s_1, T_1), (s_2, T_2), \dots, (s_K, T_K)\}$, 其中每个约束对 (s_k, T_k) 的源语言片段 s_k 是 \mathbf{x} 中的连续词元,目标语言片段 T_k 是 \mathbf{y} 中与 s_k 对应的连续词元,任务目标是训练翻译模型预测条件概率 $P(\mathbf{y} | \mathbf{x}, C; \theta)$. 条件概率的计算如式(1)所示.

$$\begin{aligned}
 \mathbf{y} &= \operatorname{argmax}_{\mathbf{y}} P(\mathbf{y} | \mathbf{x}, C; \theta), \\
 P(\mathbf{y} | \mathbf{x}, C; \theta) &= \prod_{n=1}^N P(y_n | \mathbf{x}, \mathbf{y}_{<n}, C; \theta).
 \end{aligned}
 \tag{1}$$

其中: θ 是需要优化的参数集, $y_{<n}$ 是第 n 步之前解码的部分译文, y_n 是第 n 步解码的文本.

2.2 多任务学习模型

LCT 不仅需要模型具有较高的翻译能力, 还需要模型理解双语约束片段间的关联性. 为了实现这一目标, 本文提出一种基于多任务学习的 LCT 方法. 具体的模型架构如图 2 所示.

该模型由 3 个任务子网、常规 MT 子网、一个共享子网以及两组 Transformer 编码器和解码器组成, 所有子网均由 3 层堆叠的全连接层构成. 编码器间使用硬参数共享策略, 以促进跨任务的知识迁移, 同时通过任务子网维护任务间的独立性, 确保特异化的学习能力. 解码器则使用独立的参数配置, 以适应不同任务类型的解码需求.

此外, 每个任务均通过共享子网利用部分通用参数, 以增强模型的泛化能力. 训练时, 源文本与标签序列同时输入编码器中, 由编码器层提取源文本的上下文信息和标签序列的语义信息. 随后, 解码器将结合编码器输出与历史解码信息生成目标序列.

多任务学习模型利用不同任务特有的学习信号作为引导, 形成有益的归纳偏置, 实现高效的知识迁移, 提升模型的泛化能力. 通过联合训练 LCT 任务、

常规 MT 任务、TMTG 任务与 STTL 任务, 可以进行跨任务的知识共享. 4 种任务的具体细节与数据集构建方法将在 2.3 节中详细描述.

生成任务(LCT、NMT、TMTG)间的相关性与共享表示不仅增强了模型对 LCT 和常规 MT 间共性的理解, 也提升了模型对目标约束片段的利用能力. 生成任务的框架如图 3 所示.

生成任务的训练遵循下述过程. 首先将文本序列及其对应的标签序列同时输入嵌入层, 然后合并两者的嵌入向量后传入相应的任务子网. 经过文本编码器编码处理后, 由共享子网提取部分特征, 最后使用文本解码器输出预测结果. 生成任务隐藏状态的计算如式(2)所示.

$$\begin{aligned}
y_n &= \text{TextDec}(y_{<n}, \mathbf{H}_{\text{shared}}), \\
\mathbf{H}_{\text{shared}} &= \text{SharedSubnet}(\mathbf{E}_{\text{text}}), \\
\text{Enc}_{\text{text}} &= \text{TextEnc}(\mathbf{H}_{\text{task}}), \\
\mathbf{H}_{\text{task}} &= \text{TaskSubnet}(\mathbf{V}_x + \mathbf{V}_t), \\
\mathbf{V}_x &= \text{Embed}(x), \mathbf{V}_t = \text{Embed}(t).
\end{aligned}
\tag{2}$$

其中: x 是输入序列, \mathbf{V}_x 是输入序列的嵌入, t 是标签序列, \mathbf{V}_t 是标签序列的嵌入, \mathbf{H}_{task} 是任务子网的隐藏状态, Enc_{text} 是文本编码器的隐藏状态, $\mathbf{H}_{\text{shared}}$ 是共享子网的隐藏状态.

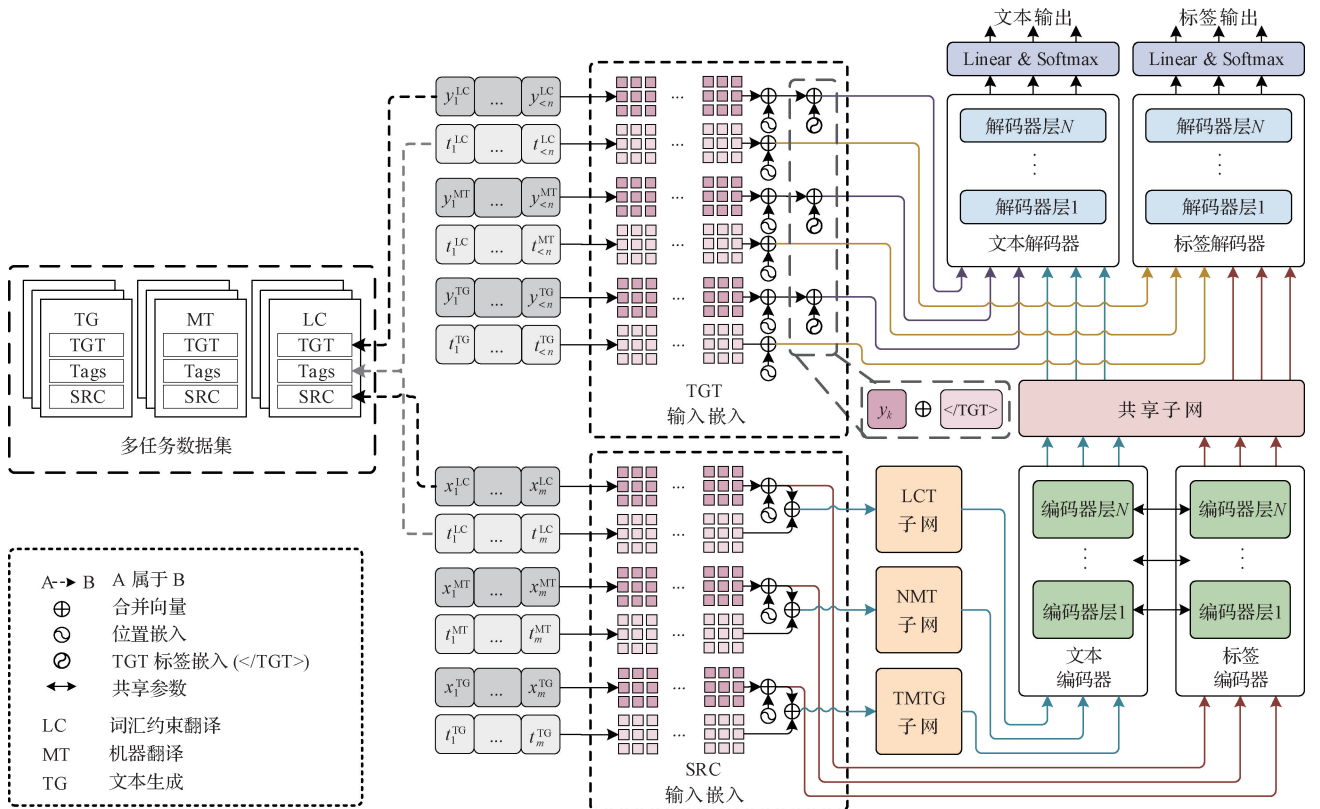


图 2 多任务模型架构

Fig. 2 Multi-task model architecture

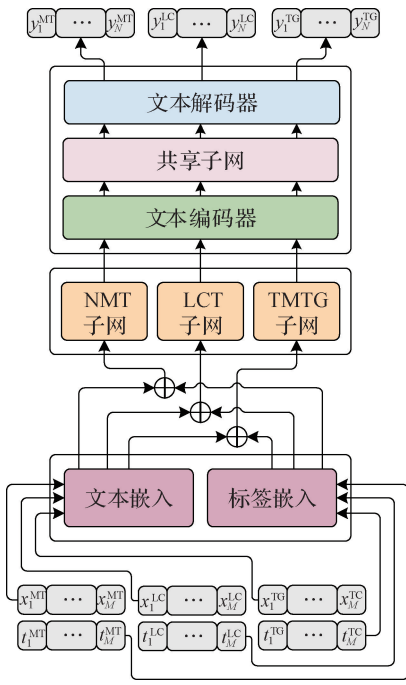


图 3 生成任务框架

Fig. 3 Generation task framework

STTL 任务则帮助模型理解双语约束对与源语言句子间的关联性。STTL 任务的框架如图 4 所示。

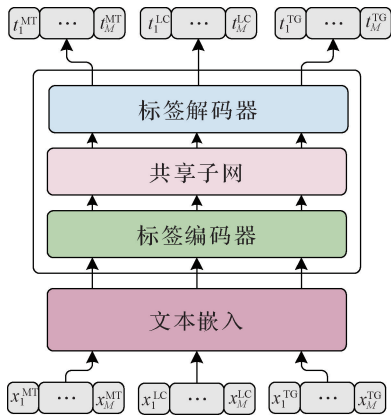


图 4 STTL 任务框架

Fig. 4 Source-side token type labeling task framework

STTL 任务的训练遵循下述过程。将文本序列输入嵌入层，在得到词嵌入向量后，直接传入标签编码器中处理。由共享子网提取部分特征后，使用标签解码器输出最终的预测结果，最后使用文本解码器输出预测结果。STTL 任务隐藏状态的计算如式(3)所示。

$$\begin{aligned}
 t_k &= \text{TagDec}(t_{<k}, \mathbf{H}_{\text{shared}}), \\
 \mathbf{H}_{\text{shared}} &= \text{SharedSubnet}(\mathbf{E}_{\text{tag}}), \\
 \mathbf{E}_{\text{tag}} &= \text{TagE}(\mathbf{V}_x), \\
 \mathbf{V}_x &= \text{Embed}(x).
 \end{aligned} \tag{3}$$

其中： \mathbf{E}_{tag} 是标签编码器的隐藏状态， $t_{<k}$ 是第 k 步之前解码的部分标签序列， t_k 是第 k 步解码的标签。

通过联合训练，多任务学习模型能够捕获上述 4 类任务的共性表示，学习相关任务间有益于词汇约束场景的知识，提升译文质量。第 4 节将详细分析多任务学习对实验结果的影响。

2.3 任务细节

本文联合训练 4 种任务，每种任务的细节如下。4 种任务的输入和输出示例将在 2.4 节中详细描述。

1) LCT 任务：多任务学习的主任务，模型需要根据给定的源语言文本以及双语约束片段，输出包含指定目标语言片段的正确译文。为了让模型同时具有约束准确性和译文流畅性，本文使用合成数据与额外嵌入在训练过程中引导模型获得词汇约束场景下的翻译能力。具体地，合成数据由插入内联注释和约束标签的双语句对构建，额外嵌入的序列是与模型输入长度相同的一串符号序列，用于标识输入序列中的词元类型，强化双语约束片段间的关联性。

给定源语言句子 $\mathbf{x} = [x_1, x_2, \dots, x_M]$ 、目标语言句子 $\mathbf{y} = [y_1, y_2, \dots, y_M]$ 以及约束对集合 $C = \{(s_1, T_1), (s_2, T_2), \dots, (s_K, T_K)\}$ 。对于 \mathbf{x} 中的任意片段 $s_Q = \{x_i, x_{i+1}, \dots, x_j\}, 1 \leq i \leq j \leq M$ ，如果 s_Q 在约束对集合 C 中存在相同的源语言约束片段 s_k ，则将其转换为带有约束标签的内联注释形式 $s_Q' = \langle c \rangle s_Q T_k \langle /c \rangle$ ，LCT 任务的输入 $\mathbf{x}^{\text{LC}} = [s_1', s_2', \dots, s_Q']$ 。 s_Q' 的定义如式(4)所示。

$$s_Q' = \begin{cases} \langle c \rangle s_Q T_k \langle /c \rangle, & (s_Q, T_k) \in C, \\ s_Q, & \text{其他}. \end{cases} \tag{4}$$

其中， $\langle c \rangle$ 与 $\langle /c \rangle$ 为约束标签，用于标记双语约束对在双语句对中的边界。

对于 \mathbf{y} 中的任意片段 $T_Q = [y_i, y_{i+1}, \dots, y_j], 1 \leq i \leq j \leq N$ ，如果 T_Q 在约束对集合 C 中存在相同的目标语言约束片段 T_k ，则将其转换为带有约束标签的形式 $T_Q' = \langle c \rangle T_Q \langle /c \rangle$ ，LCT 任务的输出 $\mathbf{y}^{\text{LC}} = [T_1', T_2', \dots, T_Q']$ 。 T_Q' 的定义如式(5)所示。

$$T_Q' = \begin{cases} \langle c \rangle T_Q \langle /c \rangle, & (s_k, T_Q) \in C, \\ T_Q, & \text{其他}. \end{cases} \tag{5}$$

内联注释与约束标签不仅为源语言句子添加额外的上下文信息，还使目标约束以相同的形式出现在双语句对中，这可以增强模型复制相同片段的能力。

给定表示双语句子词元类型的符号集合 $C^{\text{TAG}} = \{\langle /SRC \rangle, \langle /TGT \rangle, \langle /AS \rangle, \langle /AT \rangle\}$ 。其中， $\langle /SRC \rangle$ 和 $\langle /TGT \rangle$ 分别指示不属于约束对的

源语言与目标语言词元, $\langle /AS \rangle$ 和 $\langle /AT \rangle$ 分别指示属于约束对的源语言与目标语言词元. 作为源语言句子额外嵌入的标签序列 $\mathbf{t}^{LC} = [t_1^{LC}, t_2^{LC}, \dots, t_M^{LC}]$, $t_m^{LC} \in C^{TAG}$ 中的符号与源语言句子中的词元逐一匹配, 用于提示模型目标约束片段与源语言句子的哪些部分存在更显著的相关性.

使用上述策略构建的双语句对与标签序列, 可以帮助模型建模 s 和 T 间的对应关系. 给定训练集 $D^{LC} = \{\mathbf{x}^{LC}, \mathbf{y}^{LC}, \mathbf{t}^{LC}\}$, LCT 任务的训练目标是使最小化式(6)的损失函数.

$$L(D^{LC}, \theta^{shared}, \theta^{LC}) = -\sum_{D^{LC}} \log P(\mathbf{y}^{LC} | \mathbf{y}^{LC}, \mathbf{t}^{LC}; \theta^{shared}, \theta^{LC}), \quad (6)$$

其中: θ^{shared} 是多任务学习模型的共享参数集, θ^{LC} 是 LCT 任务的独立参数集.

2) NMT 任务: 将源语言句子与标签序列同时输入模型, 训练模型输出目标语言句子, 用于增强模型的翻译能力. 形式上, 给定源语言句子 $\mathbf{x}^{MT} = [x_1^{MT}, x_2^{MT}, \dots, x_M^{MT}]$ 、目标语言句子 $\mathbf{y}^{MT} = [y_1^{MT}, y_2^{MT}, \dots, y_N^{MT}]$ 以及标签序列 $\mathbf{t}^{MT} = [t_1^{MT}, t_2^{MT}, \dots, t_M^{MT}]$, NMT 任务的训练集 $D^{MT} = \{\mathbf{x}^{MT}, \mathbf{y}^{MT}, \mathbf{t}^{MT}\}$, 训练目标是使最小化式(7)的损失函数.

$$L(D^{MT}, \theta^{shared}, \theta^{MT}) = -\sum_{D^{MT}} \log P(\mathbf{y}^{MT} | \mathbf{x}^{MT}, \mathbf{t}^{MT}; \theta^{shared}, \theta^{MT}), \quad (7)$$

其中 θ^{MT} 是 NMT 任务的独立参数集. 联合训练 NMT 与 LCT 任务, 模型能捕获不同翻译任务间的共性与差异, 提升主任务的翻译能力.

3) TMTG 任务: 将目标约束片段与标签序列同时输入模型, 训练模型输出完整的目标语言句子, 用于增强模型利用目标约束片段的能力, 提升约束准确性与译文流畅性. 给定约束对集合 $C = \{(s_1, T_1), (s_2, T_2), \dots, (s_K, T_K)\}$ 和标签序列 $\mathbf{t}^{TG} = [t_1^{TG}, t_2^{TG}, \dots, t_K^{TG}]$, 模型输入 $\mathbf{x}^{TG} = [\langle c \rangle T_1 \langle /c \rangle, \dots, \langle c \rangle T_K \langle /c \rangle]$, 模型输出 $\mathbf{y}^{TG} = \mathbf{y}^{LC}$, TMTG 任务的训练集 $D^{TG} = \{\mathbf{x}^{TG}, \mathbf{y}^{TG}, \mathbf{t}^{TG}\}$, 训练目标是使最小化式(8)的损失函数.

$$L(D^{TG}, \theta^{shared}, \theta^{TG}) = -\sum_{D^{TG}} \log P(\mathbf{y}^{TG} | \mathbf{x}^{TG}, \mathbf{t}^{TG}; \theta^{shared}, \theta^{TG}), \quad (8)$$

其中 θ^{TG} 是 TMTG 任务的独立参数集. 联合训练 TMTG 与 LCT 任务, 模型可以有效利用目标约束信息, 捕获约束片段与完整译文间的联系, 并构建合适的上下文. 这不仅可以提升主任务的约束准确性, 还

可以提升译文的流畅性.

4) STTL 任务: 将上述 3 类任务的源文本输入模型后, 在不提供标签序列的情况下, 训练模型使用 C^{TAG} 中的元素对文本序列进行标注, 用于增强模型对双语约束对与源语言句子间关系的理解. 给定 LCT 任务的源文本 \mathbf{x}^{LC} 和标签序列 \mathbf{t}^{LC} 、常规 MT 任务的源文本 \mathbf{x}^{MT} 和标签序列 \mathbf{t}^{MT} 以及 TMTG 任务的源文本 \mathbf{x}^{TG} 和标签序列 \mathbf{t}^{TG} , STTL 任务的训练集 $D^{SL} = \{\mathbf{x}^{SL}, \mathbf{t}^{SL}\}$, 训练目标是使最小化式(9)的损失函数.

$$L(D^{SL}, \theta^{shared}) = -\sum_{D^{SL}} \log P(\mathbf{t}^{SL} | \mathbf{x}^{SL}; \theta^{shared}). \quad (9)$$

其中: 模型输入 $\mathbf{x}^{SL} = \{\mathbf{x}^{LC}, \mathbf{x}^{MT}, \mathbf{x}^{TG}\}$, 模型输出 $\mathbf{t}^{SL} = \{\mathbf{t}^{LC}, \mathbf{t}^{MT}, \mathbf{t}^{TG}\}$.

上述 3 类任务与 STTL 任务的额外嵌入可以形成互补关系. 标签序列作为附加信息, 指示源文本的词元类型, 仅将其作为额外嵌入对模型的提示作用不明显, 当与 STTL 任务协同作用时, 模型可以更好地捕捉并理解这些隐含的分类规则, 提升主任务的约束准确性.

本文联合训练上述 4 类任务, 训练数据集 $D = \{D^{LC}, D^{MT}, D^{TG}, D^{SL}\}$. 参考 Wang 等^[32]的研究, 由于本文的每个任务都具有相同的重要性, 所以直接根据任务数量 N_{tasks} 缩放每个任务目标的权重. 最终的训练目标是使最小化式(10)的损失函数.

$$L(D, \theta) = \frac{L_{LC} + L_{MT} + L_{TG} + L_{SL}}{N_{tasks}}, \quad (10)$$

$$L_{LC} = L(D^{LC}, \theta^{shared}, \theta^{LC}),$$

$$L_{MT} = L(D^{MT}, \theta^{shared}, \theta^{MT}),$$

$$L_{TG} = L(D^{TG}, \theta^{shared}, \theta^{TG}),$$

$$L_{SL} = L(D^{SL}, \theta^{shared}).$$

其中 $\theta = \{\theta^{shared}, \theta^{LC}, \theta^{MT}, \theta^{TG}\}$.

本文的任务目标是实现 LCT, 在训练完成后, 模型将舍弃标签编码器与标签解码器, 仅使用词汇约束任务子网、共享子网、本文编码器与文本解码器来预测最终的翻译结果, 模型将通过最大后验概率 (maximum a posteriori, MAP) 对译文进行解码. 译文解码的计算如式(11)所示.

$$\mathbf{y}^{LC} = \arg \max_{\mathbf{y}^{LC}} P(\mathbf{y}^{LC} | \mathbf{x}^{LC}, \mathbf{t}^{LC}; \theta^{shared}, \theta^{LC}). \quad (11)$$

2.4 任务数据

本文联合训练上述 4 种任务, 每种任务的输入和输出示例如表 1 所示.

① <https://github.com/lemaoliu/translation-datasets-taslp>

表 1 4 种任务的输入和输出示例
Tab. 1 Examples of inputs and outputs for the four tasks

数据	任务	示例
MT 数据	SRC	我的 职责 便是 打@@ 探 消息, 夫人.
	S-TAG	</SRC> </SRC> </SRC> </SRC> </SRC> </SRC> </SRC> </SRC> </SRC> </SRC>
	TGT	Knowledge is my trade,my Lad@@ y.
	T-TAG	</TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT>
LCT 数据	SRC	我的 <c> 职责 trade </c> 便是 打@@ 探 消息,<c> 夫人 Lad@@ y </c>.
	S-TAG	</SRC> </SRC> </AT> </AS> </AT> </AT> </SRC> </SRC> </SRC> </SRC> </SRC> </AT> </AS> </AT> </AT> </AT> </SRC>
	TGT	Knowledge is my <c> trade </c>,my <c> Lad@@ y </c>.
	T-TAG	</TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT>
TMTG 数据	SRC	<c> trade </c> <c> Lad@@ y </c>
	S-TAG	</AT> </AT> </AT> </AT> </AT> </AT> </AT>
	TGT	Knowledge is my <c> trade </c>,my <c> Lad@@ y </c>.
	T-TAG	</TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT> </TGT>

其中:词汇约束句对的 SRC 和 S-TAG 为 LCT 任务的输入,TGT 为输出. MT 句对的 SRC 和 S-TAG 为常规 MT 任务的输入,TGT 为输出. 单语生成句对的 SRC 和 S-TAG 为 TMTG 任务的输入,TGT 为输出. 三类句对的 SRC 为 STTL 的输入,S-TAG 为输出. 三类句对的 T-TAG 为目标语言句子额外嵌入的标签序列.

3 数据与实验结果

3.1 数据集

在实际应用中,约束片段通常由人工翻译提供. 为了模拟真实场景,本文参考 Chen 等^[13]的研究,从词对齐模型的标注信息中抽取双语对齐片段作为双语约束对. 具体地,本文使用预训练的 Cross-align 模型自动标注对齐信息^[33],并从中抽取约束对. 每个句子中约束对的数量在 1 至 5 间随机采样,每个目标约束片段的词元数量在 1 至 3 间随机采样,所有约束对的长度之和必须小于句长的 30%.

本文在汉-英 LCT 任务上开展实验. 实验使用的数据集与 Li 等^[34]一致,该数据集包含从多个在线新闻网站收集的约 200 万条双语句对^①. 首先,使用 2.3 节的方法构建 4 类任务数据,所有任务的数据规模相同. 然后,对数据集应用 BPE(byte pair encoding)分词^[35],联合合并次数为 32 000. 训练集与验证集的信

息如表 2 所示,其中 Cons_x 表示每个句子约束对数为 x 的情形.

表 2 训练集与验证集信息

Tab. 2 Information on training set and validation set

数据类型	训练集句对数/10 ⁶	验证集句对数
Sentence	2.00	1 999
Cons_1	0.48	514
Cons_2	0.46	467
Cons_3	0.41	423
Cons_4	0.35	337
Cons_5	0.28	258

本文使用 5 个测试集,分别测试模型接受不同数量约束对时的译文质量. 测试集信息如表 3 所示.

表 3 测试集信息

Tab. 3 Information on test sets

测试集	约束句对	句子数
1	1	1 999
2	2	1 877
3	3	1 727
4	4	1 571
5	5	1 378

3.2 实验设置

本文使用 Fairseq 工具包构建模型^[36],并进行如下设置.具体地,模型共享全部嵌入参数,并使用预层归一化(pre-layer normalization)策略,编码器和解码器的层数 $n_{\text{layers}} = 6$,编码器和解码器的嵌入维度 $d_{\text{model}} = 512$,编码器和解码器的隐藏层维度 $d_{\text{hidden}} = 2\ 048$,所有子网的嵌入维度 $d_{\text{subnet}}^{\text{model}} = 512$,所有子网的隐藏层维度 $d_{\text{hidden}}^{\text{subnet}} = 2\ 048$,每个多头注意力模块的头数 $n_{\text{heads}} = 8$,激活函数为 GLEU^[37].本文使用 Adam 优化模型参数^[38],其中 $\beta_1 = 0.9, \beta_2 = 0.98, \delta = 10^{-9}$,学习率为 3^{-4} ,随机失活率(dropout)为 0.1.本文在 2 张 NVIDIA Tesla T4 Tensor Core GPU 上训练模型,每个批次(batch)处理的最大词元数量 $n_{\text{batch}} = 8\ 192$,梯度累积频率 $n_{\text{freq}} = 4$,预热步数为 4 000,当训练步数 $n_{\text{step}} = 400\ 000$ 或 10 轮(epoch)内没有性能提升时训练停止,用于测试的检查点根据模型在验证集上的表现进行选择.解码阶段,所有实验的束搜索宽度(beam size) $n_{\text{beam}} = 10$.

3.3 基线方法

为了验证本文方法的有效性,选取 6 种具有代表性的模型作为对比实验的基线方法,具体如下.

- 1) Transformer^[9]:使用机器翻译句对训练的标准 Transformer 模型;
- 2) VDBA^[12]:硬约束的代表方法,通过修改解码算法实现 LCT;
- 3) AttnVector^[17]:软约束的代表方法,通过矢量化约束片段,将目标约束与对应的源语言片段添加到注意力模块以集成约束;
- 4) Template^[20]:一种基于模板的词汇约束翻译方法,将 LCT 任务转变为模板生成任务;
- 5) BiTIIMT^[6]:另一种基于模板的 LCT 方法,将 LCT 任务转变为双语文本填充任务;
- 6) GPT-4o(OpenAI, 2024)^①:最具代表性的大语言模型,通过设计提示模板可以执行 LCT 任务.

除 GPT-4o 外,所有基线方法均在 2 张 NVIDIA Tesla T4 Tensor Core GPU 上进行训练,每个批次处理的最大词元数量 n_{batch} 、梯度累积频率 n_{freq} 、训练步数 n_{step} 、束搜索宽度 n_{beam} 以及测试检查点的选择方案与本文方法保持一致,其他超参数均采用相应论文的推荐设置. GPT-4o 通过 API 进行推理,随机种子设为 42,其他参数保持默认值,执行 LCT 任务的提示模板

遵循 Moslem 等^[39]的方法进行设计.

3.4 评价指标

本文遵循 Wang 等^[20]的观点,使用下述 4 个指标对所涉及的方法进行全面比较.

- 1) BLEU^[40]:通过计算机器译文与参考译文间的 n-gram 匹配度来评估翻译质量;
- 2) BLEURT^[41]:上下文感知评估指标,通过评估句子的语义,衡量机器译文的流畅性和准确性;
- 3) 精确匹配(exact match)分数:用于衡量机器译文中目标约束的翻译准确率;
- 4) 窗口重叠(window overlap)分数:通过计算机器译文中目标约束窗口与参考答案中目标约束窗口间的重叠率,衡量目标约束是否置于合适的上下文中,窗口大小设置为 2.

3.5 实验结果

表 4 展示了本文方法与基线方法在不同评价指标上的比较结果,其中平均得分表示模型在 5 个测试集上的平均结果,最高分数加粗表示,次高分加下划线表示,Ours 为本文提出的方法.

由表 5 可知与其他基线方法相比,本文的方法可以在 LCT 任务上取得更好的翻译效果.相较于最佳基线,平均 BLEU 分数提升 2.28 个百分点,平均 BLEURT 分数提升 0.62 个百分点,平均窗口重叠分数提升 1.50 个百分点.

本文提出的方法在 5 种约束条件下均实现了 BLEU 和 BLEURT 指标的最优表现,且随着每个句子中约束对数量的增加,性能不断提升,这表明本文提出的方法能够在不同约束条件下充分利用源语言与约束信息,生成更加准确且忠实于源文本的译文.当约束片段少于 3 个时,VDBA 的 BLEU 分数达到次高,当约束片段更多时,BiTIIMT 的 BLEU 分数达到次高.此外,在 5 种约束条件下 BiTIIMT 的 BLEURT 分数均能达到次高.这是因为硬约束方法的强制插入策略在约束片段较少时可以有效满足约束条件而不会显著影响译文流畅性,随着约束片段的增多,其对译文质量的损伤也会加剧.相反,双语文本填充方法利用预构建的模板集成约束,随着约束片段增加,模型可利用的上下文更多,译文质量随之增强.Template 在 BLEU 分数与 BLEURT 分数上略低于 BiTIIMT 但高于 AttnVector 和 VDBA,这表明单一的硬约束或软约束方法更容易损害译文质量.相较于

① <https://openai.com/index/gpt-4o-system-card/>

表 4 测试集上的实验结果

Tab. 4 Results on test sets

%

评价指标	系统	测试集 1	测试集 2	测试集 3	测试集 4	测试集 5	平均得分
BLEU	Transformer	46.66	46.81	47.18	47.57	47.74	47.19
	VDBA	<u>48.07</u>	<u>48.77</u>	49.50	49.89	49.89	49.22
	AttnVector	45.27	45.89	46.57	47.02	46.87	46.32
	Template	46.62	48.20	49.56	50.94	51.28	49.32
	BiTIIMT	46.38	48.08	<u>49.69</u>	<u>51.50</u>	<u>52.08</u>	<u>49.55</u>
	GPT-4o	32.17	33.33	34.18	35.40	35.59	34.13
	Ours	49.38	50.38	51.84	53.28	54.28	51.83
BLEURT	Transformer	72.51	73.00	73.72	73.91	73.77	73.38
	VDBA	73.13	73.32	73.17	72.99	72.11	72.94
	AttnVector	72.23	72.39	72.84	72.51	72.36	72.47
	Template	73.19	73.92	74.71	74.94	74.83	74.32
	BiTIIMT	<u>73.29</u>	<u>74.05</u>	<u>74.82</u>	<u>75.27</u>	<u>75.15</u>	<u>74.52</u>
	GPT-4o	71.17	71.9	72.12	72.34	72.08	71.92
	Ours	73.94	74.45	75.43	75.82	76.04	75.14
Exact Match	Transformer	67.03	68.62	70.00	70.19	70.89	69.35
	VDBA	99.49	<u>99.62</u>	99.42	<u>99.53</u>	99.56	<u>99.52</u>
	AttnVector	92.69	91.76	92.47	91.50	91.17	91.92
	Template	100.00	100.00	100.00	99.98	<u>99.97</u>	99.99
	BiTIIMT	100.00	100.00	100.00	99.98	100.00	99.99
	GPT-4o	84.02	84.39	84.59	85.00	84.45	84.49
	Ours	<u>99.59</u>	99.54	<u>99.49</u>	99.41	99.07	99.42
Window Overlap	Transformer	42.51	54.84	61.30	62.39	62.17	56.64
	VDBA	57.21	58.71	60.03	59.59	58.78	58.86
	AttnVector	53.62	57.82	59.66	59.46	58.94	57.90
	Template	<u>59.36</u>	60.68	62.54	62.28	62.34	61.44
	BiTIIMT	58.51	<u>60.93</u>	<u>63.25</u>	<u>63.86</u>	<u>63.70</u>	<u>62.05</u>
	GPT-4o	44.87	50.62	52.64	53.37	53.04	50.91
	Ours	60.84	62.45	64.39	64.71	65.34	63.55

Template, BiTIIMT 能够获得更高的 BLEU 与 BLEURT 分数,可能归因于约束片段按照正确译文的顺序提供,模型在生成过程中能够更自然地连接不同片段,从而提高译文的连贯性。

在对不同方法约束准确性的评估中,本文提出的方法虽然平均精确匹配分数略低于最佳基线 0.57 个百分点,但整体得分仍高于其他最优结果 AttnVector 7.50 个百分点,并在特定约束条件下达到次高分,这种差异也说明模型在优化过程中,为了提升文本流畅度

与上下文适应性,灵活调整了个别表达。BiTIIMT 和 Template 的精确匹配分数接近 100,因为前者的译文模板能够覆盖所有约束片段,后者在解码前缀中嵌入约束片段,确保生成的模板自然包含所有必要的约束信息。

在对不同方法上下文适应性的评估中,本文提出的方法在 5 种约束条件下均实现了窗口重叠指标的最优表现,这表明本文提出的方法能够充分理解双语约束对间的关系,模型可以在适当的上下文环境中生成更多准确的目标约束片段。BiTIIMT 的平均窗口重

叠分数可以达到次高,因为其译文模板提供了正确的约束片段顺序,机器译文中的约束片段更容易出现在正确的上下文中. Template 的平均窗口重叠分数略低于 BiTIIMT,可能归因于约束片段提供的顺序是随机的. VDBA 和 AttnVector 的窗口重叠分数偏低,这表明单一的硬约束或软约束方法会损伤译文整体的流畅性.

此外,GPT-4o 在各个指标上的得分普遍低于传统模型,因为 GPT-4o 的训练目标是生成具有上下文相关性的自然语言文本,在 LCT 任务中的性能不及专为该任务优化的传统模型.

通过对比分析可以发现,本文提出的方法在翻译效果上达到了最优水平,虽然在精确匹配分数上略低于最佳基线,但仍能在适当的上下文中生成更加流畅且准确的译文,这表明本文的模型可以在译文质量与约束准确性间取得平衡,充分证明了其在 LCT 任务中的有效性.

3.6 推理速度

为了比较不同 LCT 方法的推理速度,除 GPT-4o 外,所有方法都以 $n_{batch} = 1$ 的设置 在 5 个测试集上进行测试. 推理速度的测试结果如图 5 所示.

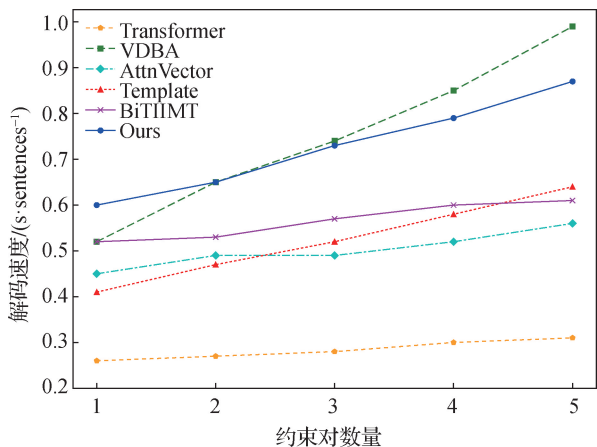


图 5 测试集上推理速度的实验结果

Fig. 5 Results of inference speed on test sets

与 Transformer 相比,所有 LCT 方法均显著增加了推理时间. 其中基于模板的 Template 方法与 AttnVector 的推理速度接近,而 VDBA 的推理速度最慢. 尽管本文提出的方法在约束片段数量为 1 时的推理速度稍逊于其他方法,但随着约束片段数量的增加,推理速度能够超越 VDBA. 这归因于 VDBA 的计算复杂度与受约束片段的数量相关,而其他方法由于未改变模型架构或无需解码完整序列,因此在推理过程中的计算负担较低. 本文提出的方法虽然在前向计算过程中需要合并标签嵌入并通过任务子网融合不

同任务的知识,引入额外开销,但在解码阶段并未增加额外的计算负担.

4 消融实验

为验证本文提出的基于多任务学习的 LCT 方法的有效性,在上述 4 个指标上进行消融实验. 最高分数加粗表示,次高分数加下划线表示,Ours 为本文提出的方法. 实验结果如表 5 所示.

其中: Mix Corpus 表示将原始双语数据、合成双语数据以及单语生成数据合并为一个数据集,不使用多任务学习方法,直接训练标准 Transformer 模型; Inline Comments 表示使用内联注释构建合成双语句对; Type 表示将标签序列作为额外嵌入; MT 表示联合训练 NMT 任务; TG 表示联合训练 TMTG 任务; SL 表示联合训练 STTL 任务; Tag 表示为数据集插入约束标签.

实验结果表明,在所有评价指标上,采用多任务学习策略训练的模型性能均优于未采用该策略的模型. 这表明多任务学习策略可以提升模型在 LCT 任务中的整体性能.

随着联合训练任务数量的增加,模型性能逐渐提升. 当 NMT 纳入训练序列时,模型的平均 BLEU 与 BLEURT 分数可以达到次高,并在特定约束条件下达到最高分数. 这表明联合训练 NMT 任务可以增强模型整体的翻译能力.

当 TMTG 纳入任务序列时,模型的平均窗口重叠分数可以达到次高. 这表明联合训练 TMTG 任务可以增强模型对目标语言的理解,生成具有连贯上下文结构的译文.

当为模型添加额外嵌入并将 STTL 纳入任务序列时,模型在除精确匹配之外的各项指标的平均得分上均达到了最高水平. 这表明额外嵌入与 STTL 任务间存在互补作用,能够共同提高模型对源语言的理解,生成忠实于源文本的机器译文.

此外,在不插入约束标签的情况下,模型生成译文整体质量最佳. 当引入约束标签后,虽然平均 BLEU 分数略有下降(-0.36 个百分点),但平均精确匹配分数取得显著的提升(+2.18 个百分点). 这一结果表明,该方法能够在翻译质量和约束准确性间实现有效平衡. 这种平衡对于 LCT 任务尤为重要,因为它允许模型在保持译文流畅性和可读性的同时,精确地生成特定约束片段.

通过对比分析可以发现,本文提出的多任务学习

表 5 测试集上消融实验的结果
Tab. 5 Results of ablation experiments on test sets

评价指标	系统	测试集 1	测试集 2	测试集 3	测试集 4	测试集 5	平均得分 %
BLEU	Mix Corpus	48.48	49.87	51.13	52.84	53.70	51.20
	Inline Comments	48.83	50.22	51.48	53.32	53.48	51.47
	+Type	48.41	50.11	51.13	52.41	53.75	51.16
	+Type+MT	<u>49.52</u>	<u>51.03</u>	52.31	53.57	54.20	<u>52.13</u>
	+Type+MT+TG	49.24	50.56	<u>52.26</u>	53.29	54.24	51.92
	+Type+MT+TG+SL	49.78	51.18	52.17	<u>53.34</u>	54.52	52.19
	+Type+MT+TG+SL+Tag	49.38	50.38	51.84	53.28	<u>54.28</u>	51.83
BLEURT	Mix Corpus	<u>73.98</u>	74.60	75.50	75.77	75.84	75.14
	Inline Comments	73.63	74.49	75.34	75.80	75.71	74.99
	+Type	73.14	74.07	75.13	75.41	75.92	74.73
	+Type+MT	73.81	<u>74.68</u>	75.70	75.94	76.14	<u>75.25</u>
	+Type+MT+TG	73.79	74.74	<u>75.63</u>	<u>75.92</u>	<u>76.05</u>	75.23
	+Type+MT+TG+SL	74.08	74.74	75.58	75.89	76.03	75.26
	+Type+MT+TG+SL+Tag	73.94	74.45	75.43	75.82	76.04	75.14
Exact Match	Mix Corpus	97.54	97.81	97.62	97.72	97.56	97.65
	Inline Comments	<u>98.54</u>	98.26	<u>97.97</u>	<u>98.10</u>	97.92	<u>98.16</u>
	+Type	98.29	98.50	<u>97.97</u>	97.91	<u>97.95</u>	98.12
	+Type+MT	96.59	96.32	96.23	96.11	96.12	96.27
	+Type+MT+TG	96.94	97.14	97.41	96.84	96.85	97.04
	+Type+MT+TG+SL	97.49	97.38	97.08	97.34	96.89	97.24
	+Type+MT+TG+SL+Tag	99.59	99.54	99.49	99.41	99.07	99.42
Window Overlap	Mix Corpus	60.09	62.20	64.61	64.38	65.09	63.27
	Inline Comments	60.54	62.11	64.24	64.91	64.54	63.27
	+Type	59.74	62.29	64.31	64.27	64.73	63.07
	+Type+MT	59.97	<u>62.85</u>	<u>65.13</u>	65.46	<u>64.82</u>	63.65
	+Type+MT+TG	60.01	62.83	64.96	<u>65.26</u>	65.49	<u>63.71</u>
	+Type+MT+TG+SL	<u>60.80</u>	63.04	65.26	65.15	65.99	64.05
	+Type+MT+TG+SL+Tag	60.84	62.45	64.39	64.71	65.34	63.55

策略可以提升 LCT 的质量. 在所有评估指标上, 多任务学习模型均优于单任务模型, 且随着联合训练任务数量的增加, 模型性能逐渐提升. 并且本文方法能在翻译质量和约束准确性间实现平衡, 在保持译文流畅性和可读性的同时, 精确生成特定约束片段.

5 结 论

本文针对现有词汇约束 NMT 方法中存在的翻译质量与约束准确性间不平衡的问题, 提出了一种基于多任务学习的 LCT 方法, 通过联合训练 LCT 任务、常

规 MT 任务、TMTG 任务与 STTL 任务, 实现跨任务的知识共享与迁移, 在翻译质量与约束准确性间实现平衡. 在新闻领域汉-英数据集上的实验结果表明, 本文的方法可以同时满足较高的翻译质量与匹配精度, 优于多个具有代表性的基线方法.

在未来的工作中, 我们将探索如何为该模型融入更多辅助任务, 使得模型学习和利用更多相关任务的知识, 进一步提升约束准确率和翻译性能.

参考文献:

[1] POST M, VILAR D. Fast lexically constrained decoding

- with dynamic beam allocation for neural machine translation[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: ACL, 2018: 1314-1324.
- [2] MICHON E, CREGO J, SENELLART J. Integrating domain terminology into neural machine translation[C]//Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: International Committee on Computational Linguistics, 2020: 3925-3937.
- [3] EXEL M, BUSCHBECK B, BRANDT L, et al. Terminology-constrained neural machine translation at SAP [C] // Proceedings of the 22nd Annual Conference of the European Association for Machine Translation. [S. l.]: EAMT, 2020: 271-280.
- [4] AILEM M, LIU J S, QADER R. Encouraging neural machine translation to satisfy terminology constraints[C]//Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Stroudsburg: ACL, 2021: 1450-1455.
- [5] LI H Y, LIU L M, HUANG G P, et al. GWLAN: general word-level autocompletion for computer-aided translation [C] // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Stroudsburg: ACL, 2021: 4792-4802.
- [6] XIAO Y L, LIU L M, HUANG G P, et al. BiTIIMT: a bilingual text-infilling method for interactive machine translation[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Dublin: ACL, 2022: 1958-1969.
- [7] KOEHN P, HOANG H, BIRCH A, et al. Moses: open source toolkit for statistical machine translation [EB/OL]. [2024-12-01]. <https://dl.acm.org/doi/pdf/10.5555/1557769.1557821>.
- [8] BAHDANAU D, CHO K H, BENGIO Y. Neural machine translation by jointly learning to align and translate[EB/OL]. [2024-12-01]. <https://arxiv.org/abs/1409.0473>.
- [9] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [J]. Advances in Neural Information Processing Systems, 2017, 30: 1-15.
- [10] HOKAMP C, LIU Q. Lexically constrained decoding for sequence generation using grid beam search[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017: 1535-1546.
- [11] HASLER E, GISPERT A, IGLESIAS G, et al. Neural machine translation decoding with terminology constraints[C] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans: ACL, 2018: 506-512.
- [12] HU J E, KHAYRALLAH H, CULKIN R, et al. Improved lexically constrained decoding for translation and monolingual rewriting[C]//Proceedings of the 2019 Conference of the North. Minneapolis: ACL, 2019: 839-850.
- [13] CHEN G H, CHEN Y, LI V O K. Lexically constrained neural machine translation with explicit alignment guidance[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(14): 12630-12638.
- [14] ZHANG J C, LUAN H B, SUN M S, et al. Neural machine translation with explicit phrase alignment[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29: 1001-1010.
- [15] SONG K, ZHANG Y, YU H, et al. Code-switching for enhancing[C]//Proceedings of the 2019 Conference of the North. Minneapolis: ACL, 2019: 449-459.
- [16] DINU G, MATHUR P, FEDERICO M, et al. Training neural machine translation to apply terminology constraints[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019: 3063-3068.
- [17] WANG S, TAN Z X, LIU Y. Integrating vectorized lexical constraints for neural machine translation[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Dublin: ACL, 2022: 7063-7073.
- [18] ZHANG H A, WANG Q, QIN B, et al. Understanding and improving the robustness of terminology constraints in neural machine translation[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: ACL, 2023: 6029-6042.
- [19] HAUHIO I, FRIBERG T. Mitra: improving terminologically constrained translation quality with backtranslations and flag diacritics[J]. European Association for Machine Translation Conferences/Workshops, 2024, 1: 100-115.
- [20] WANG S, LI P, TAN Z X, et al. A template-based method for constrained neural machine translation[C]//Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Abu Dhabi: ACL, 2022: 3665-3679.
- [21] CHEN G H, CHEN Y, WANG Y, et al. Lexical-constraint-aware neural machine translation *via* data augmentation[C] // Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence.

- Yokohama: AIO, 2020: 3587-3593.
- [22] JON J, AIRES J P, VARIS D, et al. End-to-end lexically constrained machine translation for morphologically rich languages[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Stroudsburg: ACL, 2021: 4019-4033.
- [23] SONG K, WANG K, YU H, et al. Alignment-enhanced transformer for constraining NMT with pre-specified translations[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(5): 8886-8893.
- [24] ZHU W, HU Z, XING E. Text infilling[EB/OL]. [2024-12-01]. <https://arxiv.org/abs/1901.00158>.
- [25] ZHANG Y, YANG Q. A survey on multi-task learning [J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(12): 5586-5609.
- [26] JOHNSON M, SCHUSTER M, LE Q V, et al. Google's multilingual neural machine translation system: enabling zero-shot translation[J]. Transactions of the Association for Computational Linguistics, 2017, 5: 339-351.
- [27] IMANKULOVA A, SATO T, KOMACHI M. Filtered pseudo-parallel corpus improves low-resource neural machine translation[J]. ACM Transactions on Asian and Low-Resource Language Information Processing, 2020, 19(2): 1-16.
- [28] LUONG M T, LE Q V, SUTSKEVER I, et al. Multi-task sequence to sequence learning[EB/OL]. [2024-12-01]. <https://arxiv.org/abs/1511.06114>.
- [29] HOWARD J, RUDER S. Universal language model fine-tuning for text classification[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: ACL, 2018: 328-339.
- [30] DOMHAN T, HIEBER F. Using target-side monolingual data for neural machine translation through multi-task learning[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: ACL, 2017: 1500-1505.
- [31] DONG D X, WU H, HE W, et al. Multi-task learning for multiple language translation[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing: ACL, 2015: 1723-1732.
- [32] WANG Y R, ZHAI C X, HASSAN H. Multi-task learning for multilingual neural machine translation[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg: ACL, 2020: 1022-1034.
- [33] LAI S Y, YANG Z, MENG F D, et al. Cross-align: modeling deep cross-lingual interactions for word alignment[C]//Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Abu Dhabi: ACL, 2022: 3715-3725.
- [34] LI H Y, HUANG G P, CAI D, et al. Neural machine translation with noisy lexical constraints [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, 28: 1864-1874.
- [35] SENNRICH R, HADDOW B, BIRCH A. Neural machine translation of rare words with subword units [C] // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin: ACL, 2016: 1715-1725.
- [36] OTT M, EDUNOV S, BAEVSKI A, et al. Fairseq: a fast, extensible toolkit for sequence modeling [C] // Proceedings of the 2019 Conference of the North. Minneapolis: ACL, 2019: 48-53.
- [37] HENDRYCKS D, GIMPEL K. Gaussian error linear units (GELUs)[EB/OL]. [2024-12-01]. <https://arxiv.org/pdf/1606.08415>.
- [38] KINGMA D P, BA J. Adam: a method for stochastic optimization[EB/OL]. [2024-12-01]. <https://arxiv.org/abs/1412.6980>.
- [39] MOSLEM Y, HAQUE R, KELLEHER J, et al. Adaptive machine translation with large language models [C]//Proceedings of the 24th Annual Conference of the European Association for Machine Translation. New York: EAMT, 2023: 227-237.
- [40] PAPANENI K, ROUKOS S, WARD T, et al. BLEU: a method for automatic evaluation of machine translation [C]//Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2002: 311-318.
- [41] SELLAM T, DAS D, PARIKH A. BLEURT: learning robust metrics for text generation[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2020: 7881-7892.

(责任编辑:汪 军)