软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(2):762−783 [doi: 10.13328/j.cnki.jos.007463] [CSTR: 32375.14.jos.007463]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

依存句法信息增强的完全非自回归翻译*

张建新, 郭 沛, 李俊涛, 张 民

(苏州大学 计算机科学与技术学院, 江苏 苏州 215008)

通信作者: 李俊涛, E-mail: ljt@suda.edu.cn

摘 要: 完全非自回归翻译 (fully non-autoregressive translation, Fully NAT) 的主要挑战在于, 如何在保持解码速度优势的同时, 达到与自回归翻译 (autoregressive translation, AT) 相当的翻译质量. 这是因为并行解码的特性使得 Fully NAT 方法难以捕捉目标端的依赖信息, 从而导致翻译质量下降. 因此, 利用源端的依赖信息来增强模型能力显得十分自然, 尤其是在句法信息已被证明能够有效提升 AT 方法效果的背景下. 尽管近年来这一领域取得了显著进展, 但关于在 Fully NAT 中应用句法信息的研究仍然有限. 通过在 5 个翻译基准 (如 workshop on machine translation, WMT) 上的实验发现, 依存语法信息对 Fully NAT 方法非常有帮助, 可以显著提升翻译表现, 同时解码速度的损失成本也在可接受范围内. 代码开源地址 https://github.com/tianxiexiaozhu77/syngec.

关键词: 非自回归翻译 (NAT); 句法信息; 依存关系; 机器翻译

中图法分类号: TP18

中文引用格式: 张建新, 郭沛, 李俊涛, 张民. 依存句法信息增强的完全非自回归翻译. 软件学报, 2026, 37(2): 762–783. http://www.jos.org.cn/1000-9825/7463.htm

英文引用格式: Zhang JX, Guo P, Li JT, Zhang M. Dependency-syntax-information-enhanced Fully Non-autoregressive Translation. Ruan Jian Xue Bao/Journal of Software, 2026, 37(2): 762–783 (in Chinese). http://www.jos.org.cn/1000-9825/7463.htm

Dependency-syntax-information-enhanced Fully Non-autoregressive Translation

ZHANG Jian-Xin, GUO Pei, LI Jun-Tao, ZHANG Min

(School of Computer Science and Technology, Soochow University, Suzhou 215008, China)

Abstract: The main challenge of fully non-autoregressive translation (Fully NAT) lies in maintaining translation quality comparable to autoregressive translation (AT) while preserving the decoding speed advantage. This challenge arises from the parallel decoding nature, which prevents Fully NAT methods from capturing dependency information on the target side, leading to degraded translation quality. Therefore, enhancing the model's ability to capture dependency information from the source side is a natural approach, especially given that syntactic information has proven effective in improving AT methods. Although significant progress has been made in this area in recent years, there has been limited research on incorporating syntactic information in Fully NAT. Through experiments on five translation benchmarks (e.g., workshop on machine translation, WMT), it is found that dependency syntax information is highly beneficial for fully NAT methods, significantly improving translation performance with a decoding speed cost that remains within an acceptable range. The code has been released at https://github.com/tianxiexiaozhu77/syngec.

Key words: non-autoregressive translation (NAT); syntactic information; dependency relation; machine translation

近年来, 神经机器翻译 (neural machine translation, NMT) 在基于 Transformer[1]的自回归模型推动下取得了显著进展, 尤其是在大语言模型 (large language model, LLM) 不断涌现和发展的背景下[2,3]. 然而, 这些模型主要采用自回归模型 (autoregressive model, AR) 解码模式, 即严格按照从左到右的顺序逐一生成词元. 因此, 它们的推理效率较低, 且随着模型参数数量的增加, 这一效率问题变得愈发严重[4]. 在实际应用中, 这一效率瓶颈已逐渐成为主

* 基金项目: 国家自然科学基金 (62206194); 江苏省自然科学基金 (BK20220488)
收稿时间: 2024-11-20; 修改时间: 2025-02-15, 2025-04-21; 采用时间: 2025-04-28; jos 在线出版时间: 2025-12-10
CNKI 网络首发时间: 2025-12-12

要挑战. 尤其是在需要高效推理的场景中 (如实时翻译、对话机器人). Medusa 框架[5]重新审视了非自回归生成[6], 并提出在大语言模型中添加少量完全非自回归生成头, 以加速推理过程, 该方案已成为提升大语言模型推理速度的有效且具有代表性的加速技术之一. 鉴于此, 即便在大语言模型的时代, 探索完全非自回归翻译 (fully non-autoregressive translation, Fully NAT) 的优化方法依然具有重要的研究价值. 由于 Fully NAT 的并行解码特性, 模型依然难以有效捕捉目标语言的内部依赖关系, 导致在翻译质量上的损失. 句法信息增强有助于更好地捕捉语言内部的依赖关系, 进而减少翻译质量上的损失.

Fully NAT 方法的性能不如 AT 方法的主要原因在于难以捕捉目标端的内部依赖信息. Fully NAT 方法通常假设目标词在给定源句子的条件下是独立的, 从而忽略了目标词之间的相互依赖关系. 一些研究尝试通过非自回归翻译 (non-autoregressive translation, NAT) 模型在保持并行解码的同时捕捉目标词之间的依赖关系[6–8]. 迭代的非自回归翻译 (iterative NAT) 模型[9–11]通过多次迭代来优化翻译结果. 尽管 Iterative NAT 模型在翻译质量上具有一定优势, 但其多次解码过程削弱了 NAT 模型的速度优势[8]. 因此, 在提升翻译质量的同时保持速度优势的核心问题在于如何在保持 Fully NAT 推理效率优势的同时, 利用更多的目标端的依赖信息. 考虑到 NMT 是一项语言对齐任务, 一些现有的 AT 方法通过从源端捕捉依赖信息, 实现了显著的性能提升[12–14]. 受到 AT 方法成功利用源端依存句法信息的启发 (文献 [15] 也揭示了编码器对最终翻译性能的更直接影响), 我们重新审视了将源端依存句法信息融入 Fully NAT 中的有效性. 具体来说, 使用 SuPar[16]获取源句子的依存句法信息. GCN[17]擅长处理图结构, 可以捕捉依存句法结构[14,18], 因此本文利用 GCN 编码源句的依存信息. 编码后的依存句法信息随后通过加权求和与句子编码器获得的源句子嵌入相结合. 经过这一过程后, 所得信息将直接输入到解码器中. 通过将源端的句法结构与 NAT 模型的句子编码器输出结合, 我们的方法能够有效地在并行解码过程中利用依赖信息, 从而提升翻译质量.

我们工作的贡献包括重新审视了在 Fully NAT 中引入依存句法信息的方法, 进一步拓展了依存句法信息在 Fully NAT 中作用的研究. 提出了 SynNAT 模型, 该模型将编码后的依存句法信息与句子信息结合, 作为 SynNAT 编码器的最终隐藏状态. 在多个基准测试上的实验表明, 我们的方法能够超越现有的 Fully NAT 研究. 具体而言, 在 WMTEN→RO 数据集上, 我们的方法在 GLAT+NPD (m=7) 模型上实现了最高 1.47 的 BLEU 分数提升.

1 相关工作

1.1 Fully NAT 模型

为了建模 NAT 中输出词语之间的依赖关系, 许多研究引入了各种形式的潜在变量[6,19,20]. 文献 [21] 首先生成可能的翻译输出, 然后对这些结果进行单轮精炼. 此外, 一些研究探讨了从自回归模型向非自回归模型传递知识以提升性能[22]. 部分研究集中在使用不同的训练目标[23]或添加正则化项来提高非自回归模型的有效性[24,25]. 文献 [26] 提出通过预训练短语表将源语言直接映射为目标端短语, 作为解码器输入, 以提供粗粒度的目标端依赖信息. 利用回译策略将解码器内部输出到 Softmax 层之前的隐状态重建为源语言, 增强解码器对源语言语义的传递能力. 文献 [27] 提出通过门控机制动态融合原文本词向量与高层语义特征信息, 以增强解码器输入. 该方法还使用基于课程学习的训练策略, 能够根据模型当前表现动态调整学习任务难度, 从而优化 Fully NAT. 文献 [28] 提出一种基于课程学习的非自回归翻译方法, 并在此基础上提出一种基于模型蒸馏与增强网络的翻译增强方法. 为保持方法高效性, 通过模型蒸馏训练用于生成初步翻译表征的浅翻译模型, 为使增强网络具备提升翻译表现的能力, 通过预训练向增强网络融入条件上下文知识与错误恢复能力. GLAT[29]在非自回归翻译任务中表现出色, 且推理速度高效. 其创新的 glancing sampling 机制有效捕捉源句的依赖关系, 同时通过并行解码提高了翻译效率. 此外, 该模型的设计兼顾了降低计算复杂度并保持翻译质量. 因此, 本文的工作基于 GLAT 展开.

1.2 Iterative NAT 模型

迭代优化是一种通过多轮解码来提升输出质量的策略. 文献 [30] 提出了基于去噪自编码器的迭代优化方法.

文献 [10] 在推理过程中通过插入和删除操作优化翻译输出. 文献 [9] 提出了掩蔽语言模型, 通过迭代替换被掩蔽的词元生成新的输出. 文献 [11] 引入了自适应掩蔽策略, 以增强解码器的优化能力, 并简化编码器的优化过程. 文献 [31] 提出了时间步感知条件掩蔽语言模型, 将离散前向扩散过程与条件掩蔽语言模型相结合. 该方法有效提升了模型训练的鲁棒性和翻译质量. 文献 [26] 提出了采用轻量级的接合器网络, 对 XLM 预训练语言表征与 NAT 模型进行深层次 XLM-Fused 特征融合. 此外, 通过结合掩码-预测算法, 实现预训练与下游任务推理的两阶段训练范式统一. 文献 [27] 提出在 CMLM 的基础上增加两个优化模型的训练策略: N-gram 掩码和一致性学习正则化. 文献 [32] 提出了一种通过明确引入周围词元信息而改进 NAT 模型局部性能力的方法, 在编码器和解码器两个方向上引入了混合分组线性变换, 以获得更具局部感知性的表示. 近年来, 文献 [7] 建议增强预测目标词的表示独特性, 并提出了观察邻居方法. 文献 [33] 提出了掩蔽生成序列建模方法, 该方法直接作用于多个音频标记流. 文献 [34] 提出了一种创新且高效的领域自适应方法 Bi-kNN, 专为非自回归模型定制的 k 近邻算法. 尽管这些方法在准确性上具有明显优势, 但多次解码显著降低了非自回归模型的推理效率.

1.3 源端句法信息增强神经机器翻译

在 NMT 领域, 研究人员探讨了如何有效地将源端依存句法信息融入模型中, 以提升翻译质量. 尽管传统的序列到序列 (seq2seq) NMT 模型在一定程度上可以从源语言句子中学习依存句法信息, 但这种学习通常是有限的, 难以充分利用依存句法结构来改善翻译效果. 为了解决这一问题, 研究人员提出了多种方法, 旨在将依存句法信息融入 NMT 模型中[35]. 文献 [36] 通过图卷积网络在编码器中引入依存树信息, 使得单词的表示能够捕捉其语法邻域信息, 从而改善神经机器翻译的性能. 文献 [37] 提出了双向树状编码器, 利用句法树增强 NMT 的编码过程, 同时提出树覆盖注意力机制, 使得模型在解码过程中能更有效地利用句法结构. Recurrent Graph Encoder 模型[38]通过显式建模句法依存关系, 同时保持词序信息, 以提升序列建模能力. 文献 [39] 扩展了局部注意力 (local attention) 机制, 引入句法距离约束, 使得注意力机制能够优先关注与当前目标词依存关系更紧密的源词. 文献 [40] 探讨了在 Transformer 中引入源句句法信息以提升神经机器翻译的方法, 比较了基于位置编码和输入嵌入的两种方式来利用源句的依存句法信息, 同时保持 Transformer 结构不变, 从而保证其计算效率. 文献 [41] 提出了一种将句法信息融入复数值编码器-解码器 (complex-valued encoder-decoder) 架构的方法, 该方法通过注意力机制联合学习从源端到目标端的词级和句法级注意力分数, 其不依赖于特定的网络架构, 可直接集成到任意序列到序列框架中. 文献 [35] 将源句的解析树线性化以获得结构标签序列. 在此基础上, 提出了 3 种不同的编码器将源句法信息融入 NMT: 并行 RNN (同步学习单词与标签)、分层 RNN (两级结构处理单词和标签)、混合 RNN (拼接单词与标签序列). 所提出的 3 个句法编码器都能提高翻译准确度. 文献 [42] 提出利用一个经过充分训练的端到端依存句法解析器的中间隐藏表示, 称为句法感知词表示 (syntax-aware word representation, SAWR). 然后将 SAWR 直接与标准词嵌入拼接, 以增强基本 NMT 模型的表示能力. 文献 [43] 尝试同时利用源端和目标端的依存树, 在解码过程中构建目标端依存结构, 以帮助模型更精确地预测目标词的结构关系. 文献 [44] 基于依存树将源端长距离依赖关系纳入 NMT, 丰富了每个源状态的全局依存结构, 这可以更好地捕捉源句的固有句法结构. 这些研究清楚地表明, 结合依存句法信息的模型在多个翻译任务上均能显著提高 BLEU 分数, 验证了依存句法信息对 NMT 的有效性.

此外, 少数研究关注通过句法信息增强非自回归翻译 (NAT). 其中, SynST[45]利用句法结构来增强 NAT. 该方法首先基于自回归解码器生成结构化的句法树解析块. 然后, 利用这些解析块引导解码器进行目标词的非自回归生成. 然而, 这种方法降低了解码速度, 失去了纯粹非自回归的优势. SNAT 模型 (syntactic and semantic structure-aware non-autoregressive Transformer model)[46]提出了在非自回归 Transformer 中整合词性和命名实体等句法与语义结构的方法. 与这些方法不同, 文献 [47] 提出了一种基于源语言依存关系的非自回归解码模型. 该模型采用全局与局部注意力相结合的方式, 对源端依存关系进行显式建模, 赋予核心词权重并结合全局注意力. 文献 [47] 同时提出融入源语言依存标签的非自回归机器翻译模型. 该模型通过更细粒度的依存关系标签间接融入目标词的上下文, 从而减少语法结构错误. 在这些启发的基础上, 我们开展了关于使用句法信息来增强非自回归翻译模型的研究. 与现存模型不同的是, 我们综合考虑依存句法树中节点的输入弧、输出弧和依存关系标签来增强 Fully NAT

的翻译性能, 显著提升了翻译质量.

2 基于依存句法信息增强的 Fully NAT

2.1 符号说明

给定一个翻译数据集 D, 该数据集包含多个句子对, 每个句子对表示从源语言句子 $X=(x_1,x_2,\ldots,x_N)$ 到目标语言句子 $Y=(y_1,y_2,\ldots,y_M)$ 的翻译关系. 其中, N 表示源语言句子的长度, M 表示目标语言句子的长度.

源语言句子 X 通过依存句法分析器进行分析, 生成依存句法树 G_x=(V_x, E_x), 其中, V_x 表示一系列节点, 每个单词作为一个节点. E_x 表示一系列边, 表示单词之间的依存关系. 通常, 每个句子有一个虚拟根节点, 根节点连接到句子的核心词 (例如谓词). 边的标签通常表示依存关系, 如主语、宾语、修饰语等. 图 1 展示了对应于句子的依存句法树.

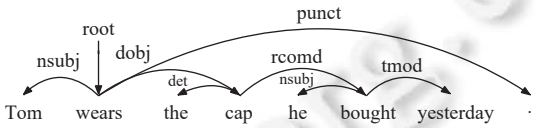


图 1 句子及其对应的依存句法树

我们将 R 表示为依存关系标签的集合. 依存句法树中的每一条边通过三元组 $(i,j,r_{i,j})$ 来表示, 该三元组能够有效且精确地描述给定句子中单词之间的依存关系. 父节点是连接的起始点, 边标签表示具体的连接关系类型, 而子节点是连接的终点. 例如, 三元组 $(i,j,r_{i,j})$ 表示从父节点 v_i 到子节点 v_j 的依存关系, 边的关系标签为 $r_{i,j}\in R$. 如图 1 所示, 三元组 (wears, Tom, nsubj) 具体表示父节点 wears 到子节点 Tom 的依存关系, 边的关系标签为 nsubj. 同时, 连接某个节点的弧分为输入弧和输出弧, 比如, 节点 wears 具有来自节点 root 的输入弧和指向节点 Tom 的输出弧.

2.2 句子编码器

如后文图 2 所示, SynNAT 模型采用 Transformer 编码器作为句子编码器, 对输入句子进行编码. 句子编码器采用 L_1 个 Transformer 编码器块, 每个块包含多头注意力机制和前馈全连接层, 以获得源句子中每个词元的上下文表示:

$$H^{L_1}=\mathrm{Enc}_{0:L_1-1}(emb(X)) \tag{1}$$

其中, $\mathrm{Enc}_{0:L_1-1}$ 表示句子编码器, $emb(\cdot)$ 为离散词元的嵌入函数, 将其映射为向量表示: $H^{L_1}=\left(h_1^{L_1},h_2^{L_1},\ldots,h_N^{L_1}\right)$ 为源句子 X 在句子编码器的最后一层输出的词元表示. 我们还在句子编码器以及接下来要介绍的句法编码器和解码器的每个子层周围使用了残差连接[48], 随后进行层归一化[49].

2.3 句法编码器

我们使用 DepGCN[50]来编码依存句法树. DepGCN 由多个相同的堆叠块组成, 每个块包括一个 GCN[17]子层和一个前馈全连接子层. 在句法编码器中, 我们使用了 3 个块. 对于 GCN 子层, 考虑到输入弧和输出弧在节点中的作用可能不同, 我们同时引入了依存输入弧、依存输出弧和依存关系标签.

$$H^{L_2}=\mathrm{SynEnc}_{0:L_2-1}\left(H^{L_1},A,e\right) \tag{2}$$

其中, $A\in R^{n\times n}$ 为依存关系标签的邻接矩阵, n 为源句子中的节点数. $e\in R^d$, 表示邻接关系标签的嵌入, 其中 d 是嵌入维度. $\mathrm{SynEnc}_{0:L_2-1}$ 表示由 DepGCN 模块组成的句法编码器. $H^{L_2}=\left(h_1^{L_2},h_2^{L_2},\ldots,h_N^{L_2}\right)$ 表示从句法编码器最终层获得的源句子的标记表示. 我们采用 GCN 来利用源句子 X 的依存树 G_x 中的依存句法知识. 然而, GCN 原本设计用于无标签无向图, 而依存树是一个有标签的有向图. 因此, 我们也将依存关系视为节点, 通过聚合邻近关系来获得每个节点的表示.

l-th GCN 中第 i 个标记的向量表示 h_i^l 的计算见公式 (3).

$$h_i^l = \sum_{j=1}^{n} \left(A_{ij} U_{\text{in}} + A_{ij} U_{\text{out}}^{\text{T}} \right) \tag{3}$$

其中, U_{in} 和 U_{out} 分别表示第 i 个标记的入度和出度表示, T 表示转置.

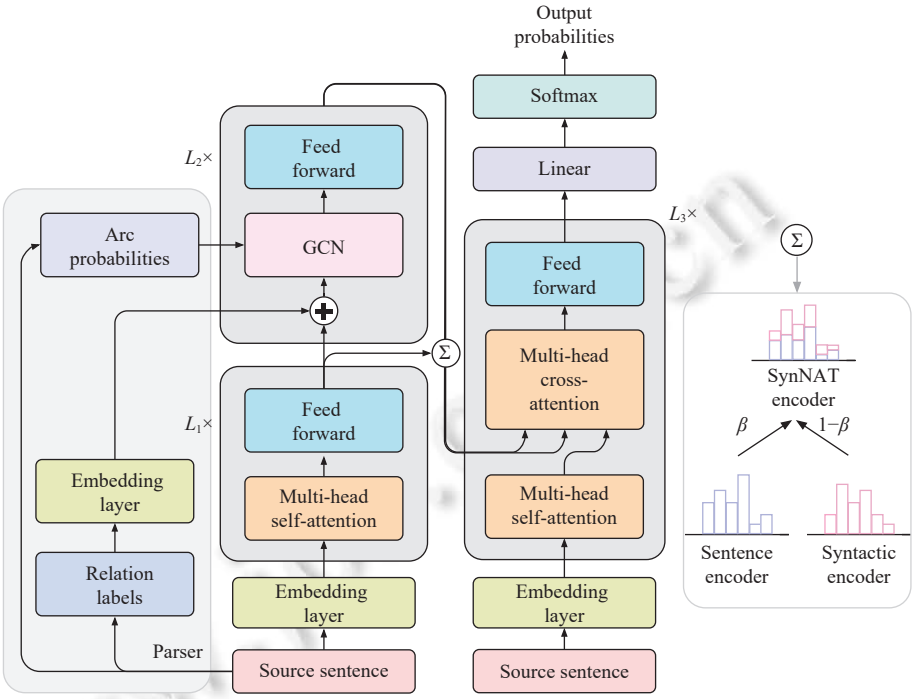


图 2 SynNAT 模型图

通过关系三元组 $(i, j, r_{i,j})$, 我们获取从节点 v_i 到 v_j 的输出表示. 其中 $r_{i,j}$ 是可学习的关系标签. 因此, 从节点 v_i 到 v_j 的关系 $(i, j, r_{i,j})$ 的表示计算如下:

$$U_{\text{out}} = ReLU\left(W_{\text{out}} \left(h_i^{l-1} \oplus e_{i,j} \right) + b_{\text{out}} \right) \tag{4}$$

其中, $e_{i,j}$ 表示从节点 i 到节点 j 的邻接标签关系 $r_{i,j}$ 的嵌入表示. 其中, $W_{\text{out}} \in R^{d \times 2d}$ 和 $b_{\text{out}} \in R^d$ 为模型的参数. $ReLU$ 是激活函数, $\oplus$ 表示向量拼接操作符.

同理, 从节点 v_j 到节点 v_i 的输入关系 $(j, i, r_{j,i})$ 的表示计算如下:

$$U_{\text{in}} = ReLU\left(W_{\text{in}} \left(h_i^{l-1} \oplus e_{j,i} \right) + b_{\text{in}} \right) \tag{5}$$

为了减少误差传播, 受到文献 [51] 的启发, 我们使用从 SuPar[16]获得的弧概率矩阵作为邻接矩阵 A, 该矩阵提供了更丰富的依存句法结构信息. 对于 $e_{i,j}$, 使用最佳头节点 v_j 的最佳关系标签. 然后, 将 GCN 子层的输出输入到前馈 (FF) 子层, 该子层由两个线性变换组成, 并在中间使用 $ReLU$ 激活函数, 如公式 (6) 所示:

$$FF\left(h_i^l \right) = ReLU\left(W_1 h_i^l + b_1 \right) W_2 + b_2 \tag{6}$$

其中, $W_1, W_2 \in R^{d \times 2d}$, $b_1, b_2 \in R^d$ 是模型参数. $ReLU$ 是激活函数.

2.4 SynNAT 编码器

考虑到我们有两个表示, 一个是来自句子编码器的上下文表示 H^{L_1}, 另一个是来自句法编码器的表示 H^{L_2}. 为了平衡句法编码器中的句法感知的词元表示 $(h_i^{L_2})$ 和句子编码器中的词元表示 $(h_i^{L_1})$ 的贡献, 我们使用它们的加权和作为最终表示, 然后将其输入到 Transformer 解码器中:

$$h_i^{\text{en}} = \beta h_i^{L_1} + (1 - \beta) h_i^{L_2} \tag{7}$$

其中, $\beta \in [0,1]$ 是一个超参数, 称为融合因子, h_i^{en} 表示来自 SynNAT 编码器的第 i 个词元的最终输出向量. 操作符∑表示向量的加权求和如图 2 右侧所示.

2.5 SynNAT 解码器

解码器及其训练策略与 GLAT[29]所描述的方法一致. NAT 模型基于 Transformer 框架. NAT 模型的目标是学习一个映射函数 f_1: $X \to Y$, 将源句子映射到目标句子, 从而估计未知的条件分布 $P(Y|X;\theta)$, 其中, θ 表示 NAT 模型的参数集合. 在训练过程中, Fully NAT 模型通过条件独立性分解进行预测, 目标是最大化:

$$\mathcal{L}_{\text{NAT}} = \sum_{t=1}^{T} \log P(y_t|H;\theta) \tag{8}$$

其中, T 表示目标句子的长度. 训练过程中, $T=M$, M 为真实目标句子的长度, 而在推理过程中, $T=P_L(X)$, 由长度预测模块 P_L 决定. 与自回归模型相比, NAT 模型省略了条件词元 $y<t$, 从而允许并行翻译, 并加速推理过程. 其中, H 的计算方式如下:

$$H = f_2(emb(X)) \tag{9}$$

其中, f_2 是复制函数, H 是经过 $emb(\cdot)$ 处理后的词元通过 UniformCopy 或 SoftCopy[22]获得的嵌入矩阵. 这些嵌入随后作为 SynNAT 模型解码器的输入.

我们的 SynNAT 采用与 GLAT[29]相同的解码器. GLAT 仅改变了训练过程, 推理过程保持完全并行, 并只需一次解码. GLAT 与标准 NAT 的区别在于, GLAT 在训练过程中使用扫视语言模型 (glancing language model, GLM), 显式地鼓励词元间的相互依赖. 为了实现自适应扫视采样, GLM 在训练过程中进行两次解码. 第 1 次解码类似于标准的 NAT, 用于评估预测准确性, 以判断当前参考词是否难以拟合. 在第 2 次解码时, GLM 基于第 1 次解码进行扫视采样, 以获得采样的参考词元, 并将这些参考词元提前透露给解码器, 学习预测剩余的未采样词元. 需要注意的是, 只有第 2 次解码会更新模型参数. GLAT 只修改训练过程, 其推理过程完全并行, 且仅需 1 次解码. 它的目标是最大化以下目标:

$$\mathcal{L}_{\text{GLM}} = \sum_{y_t \in \overline{GS\left(Y,\hat{Y}\right)}} \log P\left(y_t|GS\left(Y,\hat{Y}\right),X;\theta\right) \tag{10}$$

其中, $y_t \in \overline{GS\left(Y,\hat{Y}\right)}$ 表示未被选中的词元, 扫视采样策略 $GS\left(Y,\hat{Y}\right)$ 通过将初始预测与真实标签进行比较, 从目标句子中选择词元. 如果网络的初始预测不够准确, 它将选择更多的词元, 并在第 2 次解码时将它们输入解码器. 扫视采样可以分为两个步骤: 首先确定样本的数量 S, 然后从参考句子 Y 中随机选择 S 个词元. 对于给定的输入 X 的预测句子 $\hat{Y}$ 和参考句子 Y, 扫视采样函数 $GS\left(Y,\hat{Y}\right)$ 的目标是从 Y 中获得一个子集, 该子集由从 Y 中采样的词元组成:

$$GS\left(Y,\hat{Y}\right) = Random\left(Y,S\left(Y,\hat{Y}\right)\right) \tag{11}$$

其中, $Random(Y,S)$ 表示从目标句子 Y 中随机选择 S 个词元, 该 S 值的确定采用了与 GLAT[29]描述的相同策略, 是通过比较 Y 和 $\hat{Y}$ 之间的差异来计算得到的, 公式为: $S\left(Y,\hat{Y}\right) = \lambda \cdot d\left(Y,\hat{Y}\right)$. 采样比例 $\lambda \in [0,1]$ 是一个超参数, 用于灵活地控制采样词元的数量. $d\left(Y,\hat{Y}\right)$ 是衡量 Y 和 $\hat{Y}$ 之间差异的度量. 我们采用了汉明距离[52]作为度量方法, 计算公式为: $d\left(Y,\hat{Y}\right) = \sum_{t=1}^{T}\left(y_t \neq \hat{y}_t\right)$. 通过 $d\left(Y,\hat{Y}\right)$, 可以根据当前训练模型的预测能力自适应地决定采样数量.

3 实 验

3.1 数据集

我们在 5 个常用的机器翻译基准数据集上评估了 SynNAT 模型, 包括 WMT14 EN↔DE、WMT16 En↔RO 以及 IWSLT14 DE→EN, 数据集大小及划分见表 1. 在处理 WMT14 EN↔DE 数据集时, 我们遵循了文献 [1] 中详

细描述的方法. 对于 WMT16 En↔RO 数据集, 我们采用了文献 [30] 中描述的数据处理技术. 至于 IWSLT14 DE→EN 数据集, 我们严格按照文献 [25] 中提出的程序进行处理.

表 1 实验所用数据集大小及划分

数据集名称	训练集	验证集	测试集
WMT14 EN↔DE	4 508 785	3 003	3 000
WMT16 EN↔RO	610 320	1 999	1 999
IWSLT14 DE→EN	160 239	7 280	6 750

3.2 知识蒸馏

我们对所有数据集应用了序列级知识蒸馏[53], 具体方法如先前的研究所述[6,24,25]. NAT 模型基于条件独立性假设进行预测, 这意味着每个预测的词元不依赖于其他词元. 因此, 对于同一输入可能会产生多个输出. 例如, 德语的“Danke schön”和“Vielen Dank”都可以对应英语输入“Thank you”. 序列级知识蒸馏通过提供唯一的目标序列, 帮助模型减少输出的不确定性, 从而获得更一致的结果. 例如, “Thank you”可统一选择“Danke schön”或“Vielen Dank”作为标准译文.

首先, 我们在原始数据上训练一个标准的 Transformer 模型 (对于 WMT14 EN↔DE 使用 Transformer-large, 对于 WMT16 En↔RO 和 IWSLT14 DE→EN 使用 Transformer-base), 然后使用该模型生成的输出作为我们的训练数据. 随后, 我们使用经过蒸馏处理的数据训练我们的模型.

3.3 评估指标

实验在 FAIRSEQ[54]框架上实现. 为了评估 SynNAT 模型, 我们报告了广泛使用的 BLEU 分数[55], 这一指标用于衡量翻译任务的性能表现. 加速比 *Speedup* 的评估方法保持与之前的研究一致[56–58], 表示模型在解码过程中的速度提升倍数. 具体而言, 我们在测试集上生成目标句子, 使用批大小为 1 的设定, 并评估我们的模型相对于标准 Transformer[1]在单张 NVIDIA 3090 显卡上的推理延迟. 具体公式如下:

$$Speedup = \frac{BMT}{OMT} \tag{12}$$

其中, *BMT* 表示 baseline model (Transformer-base) 解码时间, *OMT* 表示本文模型解码时间.

3.4 实验设置

在训练过程中, 我们参考并主要使用文献 [29, 57] 中的超参数设置. 对于 WMT (workshop on machine translation) 数据集, 我们使用基本的 Transformer 配置, 包括 6 层堆叠的编码器和解码器, 每层 8 个注意力头、512 维模型和 2 048 维隐藏层, 并使用暖启动学习率调度[1], 在 4 000 个训练步骤内将学习率提升至 5E–4[1]. 由于 IWSLT 数据集较小, 我们使用一个更小的 Transformer 模型, 包括 6 层堆叠的编码器和解码器, 每层 4 个注意力头、512 维模型和 1 024 维隐藏层. 我们在 WMT/IWSLT 数据集上分别使用 64 000/8 000 个词元的批次大小训练模型, 并为 SynNAT 使用 Adam 优化器[59], β 值为 (0.9, 0.999). 所有模型训练 600 000 个训练步骤内, 最终的推理模型通过对 5 个最佳检查点进行平均生成, 检查点的选择依据是验证集上的 BLEU 分数.

3.5 对比模型

由于 SynNAT 模型是对 GLAT[29]的改进, 因此我们将 SynNAT 与强大的 GLAT[29]进行比较. 为了评估 SynNAT 的有效性, 遵循文献 [29] 的做法, 我们还将 SynNAT 集成到其他两个强大的 NAT 模型: GLTA+NPD[29] 和 GLAT+CTC[29], 这些模型采用了更复杂的机制来优化输出长度预测. 对于 NPD (noisy parallel decoding)[6], 我们首先预测 m 个目标长度候选, 然后对每个目标长度候选使用 argmax 解码生成输出序列. 接着, 利用预训练的 Transformer 对这些序列进行排序, 并选择最佳的整体输出作为最终输出. 对于 CTC (connectionist temporal classification)[60], 根据文献 [23], 我们首先将最大输出长度设置为源输入的两倍, 并在生成后去除空白和重复的标记. 对于自回归基线模型, 我们将方法与蒸馏数据训练的标准 Transformer 进行比较.

表 2 不同模型在 WMT 任务上的结果展示

	模型	I_{dec}	WMT14		WMT16		*Speedup*
			EN→DE	DE→EN	EN→RO	RO→EN	
AT Models	Transformer[1]	N	27.30	—	—	—	—
	Transformer-base[29]	N	27.48	31.27	33.70	34.05	1.0×
	Transformer-base w/KD	N	28.47	31.95	33.70	33.75	1.0×
Iterative NAT	NAT-IR[30]	10	21.61	25.48	29.32	30.19	1.5×
	LaNMT[61]	4	26.30	—	—	29.10	5.7×
	LevT[10]	6+	27.27	—	—	33.26	4.0×
	Mask-Predict[9]	10	27.03	30.53	33.08	33.31	1.7×
	CMLM+HGLT[32]	—	27.46	31.03	33.08	33.12	2.6×
	JM-NAT[15]	10	27.69	32.24	33.52	33.72	—
	DisCO[62]	Adv	27.34	31.31	33.22	33.25	3.5×
	Multi-Task NAT[63]	10	27.98	31.27	33.80	33.60	1.7×
	RewriteNAT[64]	Adv	27.83	31.52	33.63	34.09	—
	CMLMC[65]	10	27.63	31.17	33.88	34.09	1.75×
	Isotropy-Enhanced CMLMC[7]	10	26.68	30.50	34.01	33.83	1.7
	XlmF-NAT[26]	10	27.63	31.17	33.88	34.09	1.75×
	CMLM+N-gram 掩码+一致性学习[27]	10	27.32	31.08	33.69	34.02	—
	AMOM[11]	10	27.57	31.67	34.62	34.82	2.3
	CMLMC+EECR[66]	10	28.04	31.65	34.33	34.32	3.77
Fully NAT	NAT-FT[6]	1	17.69	21.47	27.29	29.06	15.6×
	Mask-Predict[9]	1	18.05	21.83	27.32	28.2	—
	SynST[45]	1	20.74	25.50	—	—	4.96×
	BoN+源语言依存标签[47]	1	21.42	25.15	29.41	30.01	9.25×
	imit-NAT[22]	1	22.44	25.67	28.61	28.90	18.6×
	BoN+源语言依存标签[47]	1	—	—	29.17	30.24	9.23×
	Flowseq[19]	1	23.72	28.39	29.73	30.72	1.1×
	NAT-HINT[67]	1	25.80	28.40	32.30	31.70	18.6×
	NAT-DCRF[68]	1	25.80	28.40	32.30	31.70	18.6×
	ReorderNAT[69]	1	22.79	27.28	29.30	29.50	16.1×
	SNAT[46]	1	24.64	28.42	32.87	32.21	22.6×
	NAT+门控增强输入, 课程学习 (NPD, k=3)[27]	1	25.68	30.45	32.00	32.71	—
	AlignNAT[70]	1	26.40	30.40	32.50	33.10	11.3×
	Fully-NAT+KD+CTC[57]	1	26.51	30.46	33.41	34.07	16.5×
	OAXE-NAT[71]	1	26.10	30.20	32.40	33.30	15.3×
	GLAT+DePA[72]	1	26.43	30.42	33.07	33.82	15.1×
	GLAT+RenewNAT[21]	1	25.75	29.67	32.53	32.93	12.5×
	NAT[26]	1	25.02	29.51	33.58	33.6	18.9×
	NAT 课程学习+增强网络[28]	1	27.08	—	33.05	—	15.2×
	PCFG-NAT[73]	1	27.02	31.29	32.72	33.07	12.6×
	DA-Transformer+BeamSearch+5-gram LM[74]	1	27.91	31.95	—	—	7.0×
w/CTC	NAT-CTC[23]	1	16.56	18.64	19.54	24.67	—
	Imputer[75]	1	25.80	28.40	32.30	31.70	18.6×

表 2 不同模型在 WMT 任务上的结果展示 (续)

	模型	I_{dec}	WMT14		WMT16		*Speedup*
			EN→DE	DE→EN	EN→RO	RO→EN	
w/NPD	NAT-FT+NPD (*m*=100)	1	19.17	23.20	29.79	31.44	2.4×
	imit-NAT+NPD (*m*=7)	1	24.15	27.28	31.45	31.81	9.7×
	NAT-HINT+NPD (*m*=9)	1	25.20	29.52	—	—	—
	Flowseq+NPD (*m*=30)	1	25.31	30.68	32.20	32.84	—
	NAT-DCRF+NPD (*m*=9)	1	26.07	29.68	—	—	6.1×
	RenewNAT+NPD (*m*=5)	1	26.54	30.55	32.86	33.35	10.7×
Ours	GLAT[29]	1	25.21	29.84	31.19	32.04	15.3×
	SynNAT	1	25.96	29.97	33.10	33.84	13.27×
	GLAT+CTC	1	26.39	29.54	32.79	33.84	14.6×
	SynNAT+CTC	1	26.85	30.36	33.62	34.47	12.37×
	GLAT+NPD (*m*=7)	1	26.55	31.02	32.87	33.51	7.9×
	SynNAT+NPD (*m*=5)	1	27.06	31.13	34.34	34.67	9.84×

3.6 实验结果

表 2 所示使用不同的基础模型在 WMT 的 4 个数据集上评估了我们的方法, I_{dec} 表示解码器迭代生成的次数.

实验结果表明, 我们的方法在所有数据集上显著提升了强大的 Fully NAT 模型 GLAT 和 GLAT+CTC 的 BLEU 分数. 尤其是在 WMT16 EN→RO 数据集上, GLAT+NPD (*m*=7) 的 BLEU 分数提升了 1.47. 不同基础模型上的一致性改进进一步验证了方法的广泛适用性和有效性. 此外, 与其他方法相比, 我们的模型不仅提升了翻译质量, 同时解码速度的损失有限, 这表明我们提出的方法能够有效提升 NAT 模型的整体性能. 这些结果不仅凸显了方法在多种翻译任务中的优势, 也为非自回归模型的进一步改进提供了重要参考. 此外, 在比较先进的模型中, PCFG-NAT[73]采用了基于概率上下文无关文法的目标端建模方式来增强 NAT 模型捕捉输出端复杂依赖关系的能力, 而我们的方法则通过句法树显式建模源端信息来优化解码过程. DA-Transformer[74]提出了一种新的方法有向无环 Transformer, 该方法在有向无环图 (DAG) 中表示隐藏状态, 其中 DAG 的每条路径对应于一种特定的翻译. 整个 DAG 结构同时捕捉多种翻译, 并在非自回归方式下实现快速预测. 与我们的方法相比, DA-Transformer 在 En↔De 数据集上取得了更高的 BLEU 分数. 然而, 由于其基于 DAG 结构的表示方式引入了较高的计算复杂度, 导致其解码速度明显较慢. 相比之下, 我们的方法在翻译质量与推理效率之间保持了更好的平衡, 使其在实时翻译场景中更具优势.

4 消融实验

本节主要探讨不同因素对模型的影响, 包括原始数据集、邻接矩阵与邻接标签嵌入的效果、源句长度、GCN 子层数量、两种编码器获得的隐藏状态的融合顺序、融合因子 β 的影响、与自回归模型结合的可行性分析、不同距离度量方式的影响以及对源语言的句法建模深入探讨等消融实验.

4.1 原始数据的影响

除在蒸馏数据上进行了相关实验外, 我们希望了解原始数据对模型的影响. 在未经蒸馏的 IWSLT14 DE→EN 数据集和蒸馏后的 IWSLT14 DE→EN 数据集上进行了实验, 结果如图 3 所示. 可以看出, 我们的模型在 IWSLT14 DE→EN 测试集上的 BLEU 分数优于相应的基线模型, 表明了框架的稳健性. 我们还展示了蒸馏后 IWSLT DE→EN 测试集的案例研究. 如表 3 中从 GLAT 和 SynNAT 模型在 IWSLT DE→EN 测试集上蒸馏的两个翻译示例. 从示例 1 中可以看出, GLAT 使用了 she 这一词, 暗示主语是女性, 而原文中的 sie 并不一定指代某个特定的人或性别 (尤其是在没有明确上下文的情况下). 而 SynNAT 则使用了 it, 这一表达更加中性, 且与 sie 可能指代非人类的含义更为契合. 在 SynNAT 生成的句子, and to images in my imagination 更符合原文 zu bildern in meiner vorstellung 的意思. 相比之下, GLAT 生成的句子 and images to pictures in my mind 显得有些重复, 并且缺乏逻辑性, 特

别是在 images to pictures 这一短语的使用上. 示例 2 中, 我们观察到 GLAT 和 SynNAT 都存在重复的问题, 但 GLAT 还出现了较为严重的拼写错误, 如 insiders information 和 obviouslyly. GLAT 和 SynNAT 都属于非自回归翻译模型的范畴. 在非自回归架构中, 模型通常是并行生成所有目标词. 与自回归模型相比, NAT 模型更容易出现词与词之间的上下文建模不足, 从而导致词语冗余或重复的现象. 这是非自回归翻译模型中普遍存在的多模态问题[76]. IWSLT14 DE→EN 数据集的规模相对较小, 模型并没有在大规模语料上进行充分的上下文学习. 在此情况下, 一旦面对较为复杂的源句, 模型的泛化能力不足, 更容易出现重复或者其他歧义现象.

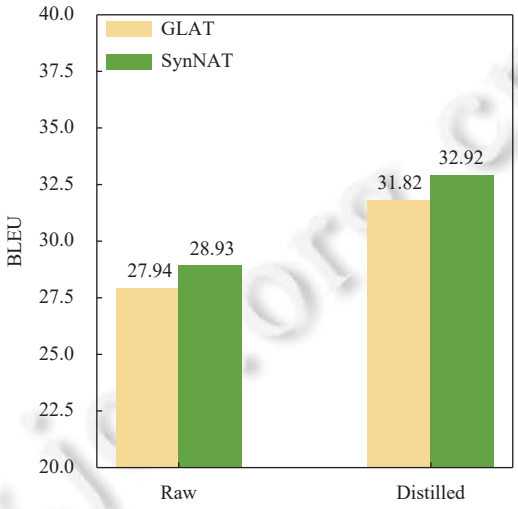


图 3 IWSLT DE→EN 任务中原始数据和蒸馏数据使用 beam=5 时的 BLEU 分数

表 3 GLAT 和 SynNAT 在蒸馏后的 IWSLT14 DE→ EN 测试集上的翻译示例

模型	示例 1	示例 2
Source	sie wurde ein goldener zugang zu einer welt voller piraten und verunglückter schiffe und zu bildern in meiner vorstellung.	und uns fiel auch auf, dass sie sehr professionell konstruiert waren, von leuten, die offensichtlich alle insider- informationen zur verfügung hatten.
Reference	it became a gilded gateway into a world full of pirates and shipwrecks and images in my imagination.	and we also saw that they are very professionally engineered by people who obviously had all insider information.
GLAT	she became a golden access to a world full of pirates and unfortunate ships and images to pictures in my mind in my mind.	and we also noticed that they were very professionally constructed by people who obviouslyly had all the insiders information information.
SynNAT	it became a golden access to a world full of pirates and unfortunate ships and to images in my imagination.	and we also noticed that they were very professionally constructed by people who obviously had all all the insider information.

香农熵 (Shannon entropy)[77]用于度量随机变量的不确定性或信息量的期望值. 熵越大, 表示不确定性越大. 我们使用香农熵来衡量“模型对词的预测不确定度” (但并不足以可靠地反映句子的歧义程度). 具体而言, 每个单词进行一次 [MASK] 替换, 然后通过 BERT (bert-base-uncased)[78]的 Masked LM 得到在该位置上所有可能词的概率分布 (Softmax 后的分布). 接着计算该分布的熵, 再将所有词的熵取平均, 得到句子级平均熵. 该方法主要衡量的是 BERT 自身对词汇预测的不确定性, 不能够完全代表真实语言环境或人工判断下的歧义程度. 此外, 我们采用 N-gram 重复率作为衡量句子潜在歧义性的一种辅助指标, 以测量句子中重复单词的比例, 采用大模型 GPT-4o-mini[2]评估、人工评估作为主观与客观结合的综合评估手段 (详见附录 A), 以进一步确保歧义检测的可靠性. 我们在没有蒸馏的 IWSLT14 DE→EN 数据上进行测试, 结果如表 4 所示. 从表 4 中可见, 给 GLAT 加入 Syntax (SynNAT) 后, 句子层面的香农熵 (entropy) 和 N-gram 重复率 (rep) 均有所下降, 意味着模型对词的预测不确定性、句子中重复单词

的比例都减少. 同时, 大模型 (GPT-4o-mini) 和人工 (human) 对译文歧义性的评分也随之降低, 表明在引入语法信息后, 译文的歧义程度得到一定程度的改善, 但是没有完全消解译文的歧义现象.

表 4　原始 IWSLT14 DE→EN 测试集上 GLAT 及其加入 Syntax 对各指标的影响

原始数据	香农熵	重复率 (%)	GPT-4o-mini评估	人工评估
GLAT	3.89	4.39	2.23	2.48
SynNAT	3.72	3.72	2.16	2.35

4.2　邻接矩阵与邻接标签嵌入的影响

我们在蒸馏后的 IWSLT14 DE→EN 数据集上测试了输入邻接矩阵、输出邻接矩阵与邻接标签嵌入对实验结果的影响, 结果如表 5 所示. 可以看到, 我们的模型在 IWSLT DE→EN 数据集上的 BLEU 分数优于 GLAT. 具体而言, 仅加入依存标签邻接矩阵 (w/probs) 嵌入的模型相比原始 GLAT 提高了 0.83 个 BLEU 分数; 加入依存输入邻接矩阵 (in_arc) 和依存输出邻接矩阵 (out_arc) 模型相比原始 GLAT 提高了 0.95 个 BLEU 分数. 结合依存输入邻接矩阵、依存输出邻接矩阵以及依存标签邻接矩阵的模型, 相较于原始 GLAT, 取得了最高的 BLEU 分数提升.

表 5　邻接矩阵与邻接标签嵌入的影响

模型	BLEU
GLAT	31.82
SynNAT	32.92
w/ in_arc & probs	32.9
w/ out_arc & probs	32.74
w/ probs	32.65
w/ in_arc & out_arc	32.77

4.3　源句子长度的影响

我们还探讨了句子长度对模型的影响, 将蒸馏后的 IWSLT14 DE→EN 测试集中源句子按 BPE 操作后的长度分为 5 个区间, 并计算每个区间的 BLEU 分数. 如图 4 所示, 我们的模型在每个区间的表现均优于对应的基线模型. 当源句子长度位于 [0, 10) 范围内时, SynNAT 实现了显著的提升. 因此, 我们的模型在不同句子长度的场景中均能提升性能.

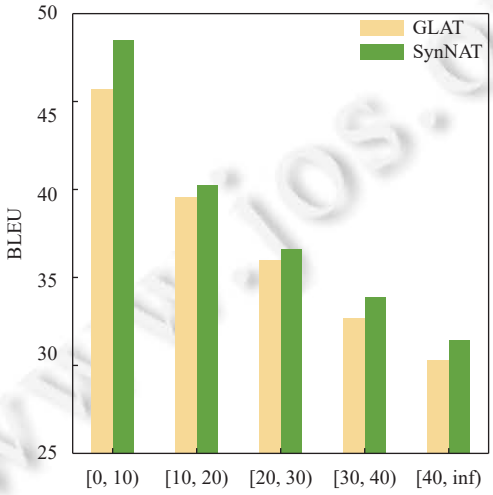


图 4　在 IWSLT14 DE→EN 任务中不同源句子长度在 beam=5 时的 BLEU 分数

4.4 GCN 子层数量的影响

我们还探讨了 GCN 子层数量 K 对模型的影响, 在蒸馏后的 IWSLT14 DE→EN 测试集上测试了不同数量的 GCN 子层, 并计算了不同 GCN 子层数量下的 BLEU 分数. 如图 5 所示, BLEU 分数在 5 层以内随着 GCN 子层数量的增加而提升, 其中 3 层相较于两层表现出最显著的提升. 但在 3 层之后, BLEU 分数的增长趋于缓慢.

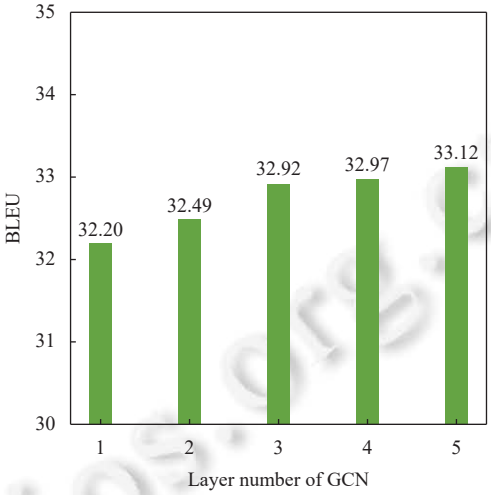


图 5 在 IWSLT14 DE→EN 任务中, GCN 使用不同层数 K 的表现

4.5 融合顺序的影响

我们还探讨了句子编码器和句法编码器的融合顺序. 图 6(a) 表示我们方法中句法编码器和句子编码器的顺序, 而图 6(b) 表示将句法编码器和句子编码器的顺序对调的方法. 两种方法的解码器保持一致. 我们使用未蒸馏和蒸馏后的 IWSLT14 DE→EN 测试集评估了两种方法的 BLEU 分数. 如图 7 所示, 先对句子进行编码再对句法进行编码的方法在模型中始终获得更高的 BLEU 分数. 不同的混合方式见公式 (13)、(14).

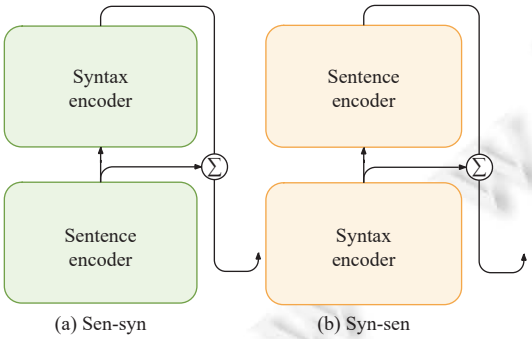


图 6 句子编码器和语法编码器的混合模式

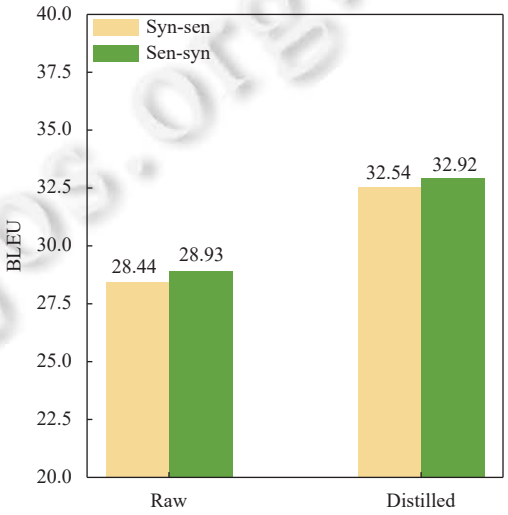


图 7 不同融合方法在 IWSLT14 DE→EN 任务上的表现

$$H_{\text{sen-syn}} = \text{SynEnc}\left(H^{L_1}, A, e\right) \tag{13}$$

$$H_{\text{syn-sen}} = \text{Enc}\left(\text{SynEnc}\left(emb(X), A, e\right)\right) \tag{14}$$

4.6 融合因子 β 的影响

在我们的方法中, 融合因子 β 用于平衡源句嵌入和依存句法信息嵌入的权重. 在蒸馏的 IWSLT14 DE→EN 验证集上, 我们对 β 进行了网格搜索 (范围 0.1–0.9, 步长 0.1) 确定 β 的最优值. 实验结果见表 6, β=0.5 时模型在验证集中达到最优性能 (BLEU=32.96), 显著优于其他设置 (如 β=0.2 时 BLEU=32.04, β=0.7 时 BLEU=30.96). 过低 (β<0.3) 或过高 (β>0.6) 的 β 值均会导致性能下降, 说明句法信息与语义信息的平衡对模型至关重要. 具体而言, β 过低时模型难以捕捉长距离依存关系, 而 β 过高时则可能过度依赖句法结构, 忽略语义上下文.

表 6 蒸馏的 IWSLT14 DE→EN 验证集上 β 对 BLEU 的影响 (beam=5)

β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
BLEU	31.97	32.04	32.55	32.87	32.96	32.90	30.96	30.79	30.20

4.7 与自回归模型结合的可行性

我们将 Syntax Encoder 与 AT 模型 (Transformer-base[1]) 结合, 使用蒸馏的 IWSLT14 DE→EN 训练集训练后, 发现在测试集上 BLEU 提升幅度 (+0.57) 显著低于其在 NAT 中的表现 (+1.1), 见表 7. 证明句法信息对非自回归场景增益更显著. 我们猜测原因有: (1) AT 通过自回归生成天然建模了目标端依赖关系, 其性能瓶颈更多在于源端语义理解而非句法建模. (2) Fully NAT 因完全并行解码, 无法显式捕捉目标端依赖, 因此更依赖源端提供的结构化信息 (如句法) 补偿这一缺陷. 我们方法在 AR 模型中依然带来了性能提升, 说明其作用机制不仅适用于 NAT 模型, 也从更普适的层面增强了源语言结构的表示与理解能力, 具有较强的通用性.

表 7 蒸馏的 IWSLT14 DE→EN 训练集上 Syntax+AT 对 BLEU 的影响 (beam=5)

模型	BLEU	模型	BLEU
GLAT	31.82	Transformer-base	34.69
SynNAT	32.92	+Syntax Encoder	35.26

4.8 不同距离度量方式的影响

本文使用了与 GLAT 相同距离度量的计算方式, 采用了汉明距离作为评估模型初始输出与参考目标序列之间差异的度量标准. 此外, 我们也认识到, 除了汉明距离外, 还有其他度量方法也适用于评估序列间的差异, 例如 Levenshtein 距离 (编辑距离)[79], 计算将一个序列转换为另一个序列所需的最小编辑操作次数, 包括插入、删除和替换. Damerau-Levenshtein 距离[80], 在 Levenshtein 距离的基础上, 增加了交换相邻字符的操作. 因此我们对这 3 种距离度量方式做了相关消融实验, 结果如表 8 所示. 不同的距离度量方式对 BLEU 分数的影响较小, 其中 Damerau-Levenshtein 距离的 BLEU 最高 (32.95), 略高于汉明距离 (32.92), 而 Levenshtein 距离的 BLEU 略低 (32.88). 这一结果表明, 考虑额外的编辑操作 (字符交换) 可能对翻译质量有所帮助, 但整体影响较为有限.

表 8 蒸馏的 IWSLT14 DE→EN 测试集上不同的距离度量方式对 BLEU 的影响 (beam=5)

度量方式	BLEU
汉明距离	32.92
Levenshtein距离	32.88
Damerau-Levenshtein距离	32.95

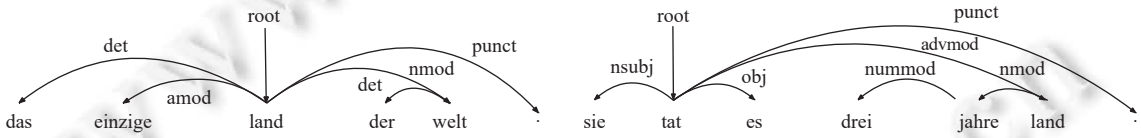
4.9 对源语言的句法建模深入探讨

本文以具体示例探讨为何对源端句法的建模能够提升目标端句法的正确性. 以蒸馏的 IWSLT14 DE→EN 测

试样本为例, 见表 9. 对其使用 SuPar 进行解析后得到句法树如图 8 所示. 两个向量的内积表示这两个向量的相关性[81], 我们计算了 GLAT 和 SynNAT 的编码器输出的不同词元的向量 h_i^{en} 的内积, 并画出热力图, 如图 9 所示, 颜色越深表示越相关. 图 10 为这两个句子的使用 GLAT 和 SynNAT 生成的目标句子的信息熵图. 我们发现, 图 9(b)、(d) 中颜色较深的方格中有比图 9(a)、9(c) 中更多的图 8 中的某些结构. 比如, 图 9(b) 中 land→welt、land→einzige 等 (在热力图中的表现为颜色更深), 然而像 land→·这样的依存关系仍未在热力图中获得足够突出, 这说明我们的方法在有效融合句法结构方面仍有改进空间. 在图 9(a) 中, land→welt 的相关程度较低 (表现为颜色更浅), 在图 10(a) 中, GLAT 没有了 welt 这个词所对应的翻译结果. 取而代之的是翻译了两次 land (land 的翻译结果为 country). 我们使用香农熵来衡量它们对词汇预测的不确定性. 熵值越大表示模型的预测越不确定, 即其概率分布较为均匀, 意味着该词的置信度较低; 相反, 熵值越小表示模型的预测较为集中, 置信度较高. 我们通过颜色深浅来可视化这种不确定性, 颜色越深表示熵越高 (不确定性越大), 颜色越浅表示熵越低 (不确定性越小). 通过信息熵可以看到, GLAT 模型在第 2 次翻译 country 时是更不确定的 (表现为颜色更深). 在图 9(c) 中, 我们也观察到了图 8(b) 中的某些结构. 比如 lang→jahre、jahre→drei 等. 在图 9(d) 中, 还发现了 tat→es. 而在图 9(c) 中 tat→es 的相关程度较低. 在图 10(b) 中, GLAT 没有了 es 这个词所对应的翻译结果. es 应翻译成 it 的地方翻译成了 for, 其对应的信息熵更高, 表示模型对于该地方翻译成 for 是更不确定的.

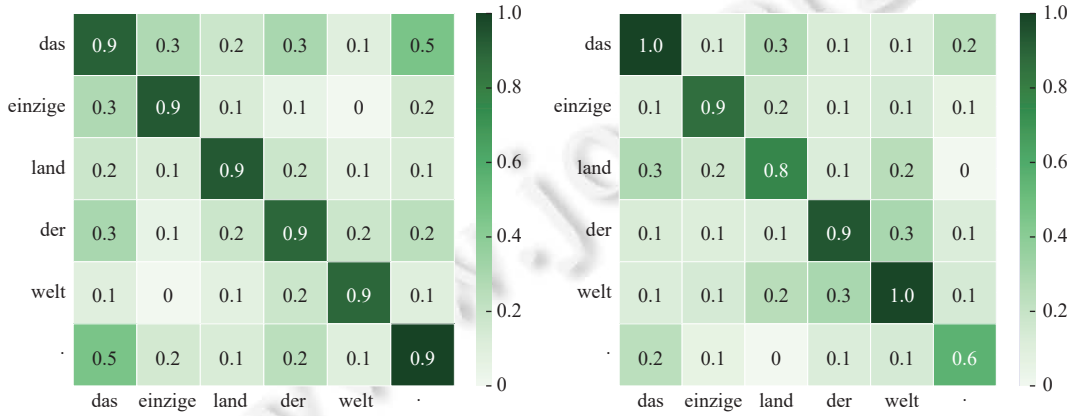
表 9 蒸馏的 IWSLT14 DE→EN 测试样本研究

模型	示例 1	示例 2
source	das einzige land der welt.	sie tat es drei jahre lang.
reference	the only country in the world.	she did it for three years.
GLAT	the only country country.	she did for three years.
SynNAT	the only country in the world.	she did it for three years.



(a) 表9中示例1的源句子对应依存句法树 (b) 表9中示例2的源句子对应依存句法树

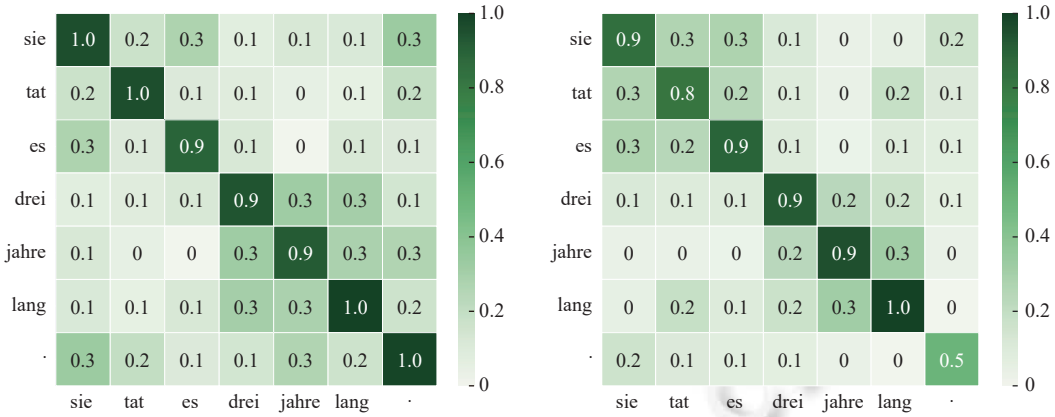
图 8 表 9 中源句子对应依存句法树



(a) GLAT encoder对表9中的示例 1 编码后的词元之间的内积

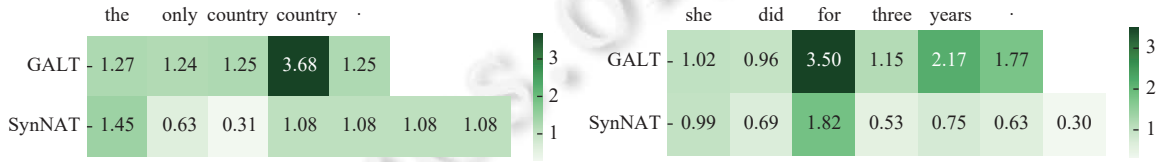
(b) SynNAT encoder 对表9中的示例 1 编码后的词元之间的内积

图 9 译文不同情况下不同编码器输出的词元之间的内积案例分析



(c) GLAT encoder 对表9中的示例 2 编码后的词元之间的内积　(d) SynNAT encoder 对表9中的示例 2 编码后的词元之间的内积

图 9　译文不同情况下不同编码器输出的词元之间的内积案例分析 (续)



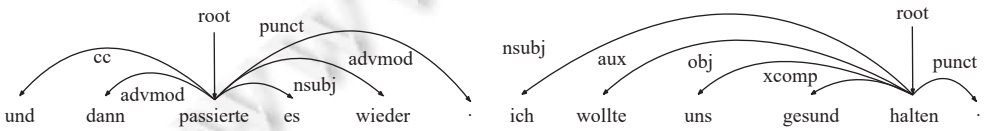
(a) 对应图9(a)、(b)的译文词汇预测信息熵　(b) 对应图9(c)、(d)的译文词汇预测信息熵

图 10　词汇预测熵值对比案例分析

此外, 我们给出了 GLAT 和 SynNAT 翻译结果均正确的 2 个例子, 见表 10, 图 11 为其对应的依存句法树, 图 12 为这两个句子分别对应 GLAT 和 SynNAT 生成的相关性矩阵. 图 13 为这两个句子的使用 GLAT 和 SynNAT 生成的目标句子的信息熵图. 图 12(a) 中 passierte→es 没有图 12(b) 中相关程度高. 在图 13(a) 中显示, GLAT 虽然将 es 正确的翻译得到了 it, 但是具有较高的信息熵, 表示模型翻译成 it 时的不确定性较高. 图 12(c) 中 halten→wollte 没有图 12(d) 中相关程度高. 在图 13(b) 中显示, GLAT 虽然将 wollte 正确的翻译得到了 wanted. 但是具有较高的信息熵, 表示模型翻译成 wanted 是更不确定的. 而 SynNAT 模型在翻译 it 和 wanted 的时候信息熵虽然比周围的词元更高, 但相比 GLAT 在此位置上的信息熵相比已经降低. 此外从结果可以看出, 尽管两个模型最终生成的译文完全相同且正确, 我们的方法在大多数单词上的熵值更低 (颜色整体更浅), 表明其在词汇预测上的置信度更高. 因此, 我们认为, 对源端句法结构的建模能够提升存在句法依存关系下的词元表示之间的相关性, 而这种增强的相关性在翻译过程中有助于提高目标端中相应词元生成的准确性, 从而进一步提升目标端句法结构的正确性.

表 10　蒸馏的 IWSLT14 DE→EN 测试样本研究 (两个模型的译文结果与金标结果一致)

模型	示例 1	示例 2
source	und dann passierte es wieder.	ich wollte uns gesund halten.
reference	and then it happened again.	i wanted to keep us healthy.



(a) 表10中示例1的源句子对应依存句法树　(b) 表10中示例2的源句子对应依存句法树

图 11　表 10 中源句子对应依存句法树

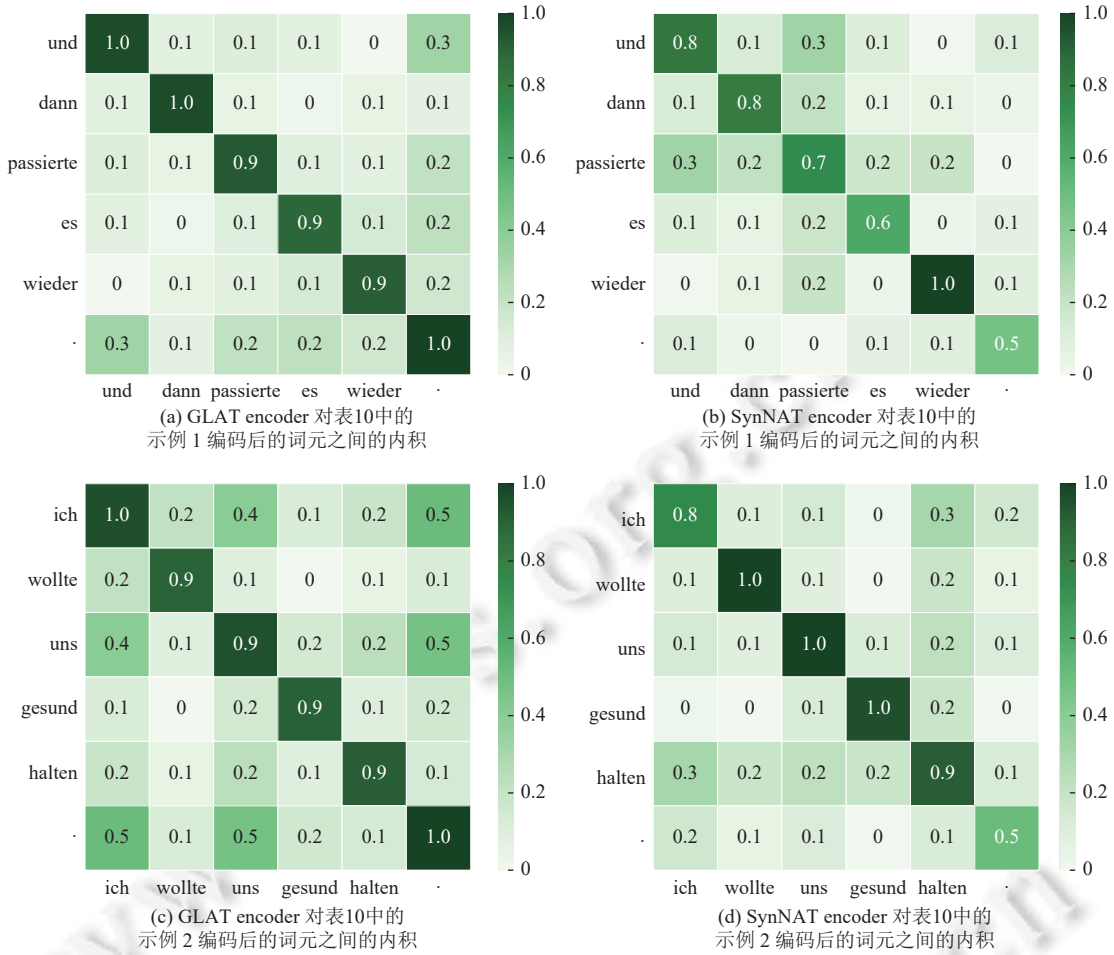


图 12　译文相同情况下不同编码器输出的词元之间的内积案例分析

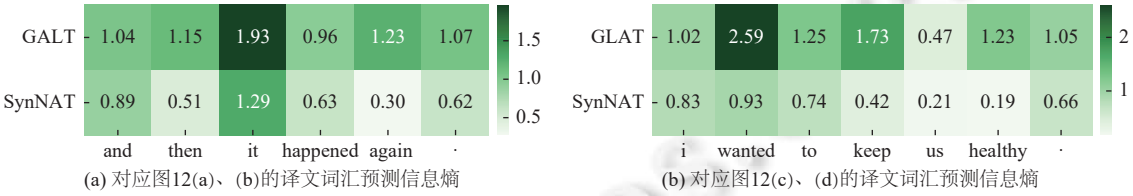


图 13　译文相同情况下词汇预测熵值对比案例分析

此外, 我们还测量了表 9 中 4 个翻译结果与整个 IWSLT14 DE→EN 蒸馏数据集中所有翻译句子的平均句法质量, 使用如下 3 个层面的指标进行评估.

Parse Validity: 该指标衡量生成句子的依存句法树结构是否合法, 具体包括是否存在唯一的根节点, 以及是否出现非法的依存关系标签 (如 dep、orphan 等). 该指标为二值型 (0 或 1), 表示依存结构是否符合基本句法规则. 该评价方式参考了 SpaCy 标注标准及常用依存分析评估规则. 1 表示合法, 0 表示不合法.

Acceptability Score: 该指标通过在 CoLA 数据集[82]上训练的 BERT 分类模型进行判别, 输出句子在语法上被人类接受的概率. 该得分反映了句子的“自然性”或“句法流畅度”, 其值域为 [0, 1], 越高表示句子越符合语法规范和语言习惯[83].

Grammar Error Count: 该指标使用 LanguageTool 工具对句子进行错误检测, 统计其中的错误数量, 主要包括主谓不一致、句子结构不完整、拼写错误等. 该指标以错误数量计, 数值越低表示句子质量越高[84].

上述指标从结构合法性、流畅性判断和细节检测这 3 个层面客观地验证我们方法在提升目标端句法准确性方面的有效性, 且在业界评估实践中均为常用手段. 实验结果如表 11 所示. 从表中最后两行所有数据测试的平均值可见, 在 3 项指标上, SynNAT 生成的译文整体优于 GLAT, 表明源端句法建模在目标句法结构的生成中起到了积极作用.

表 11 蒸馏的 IWSLT14 DE→EN 测试样本评估

模型	句子	Parse Validity	Acceptability Score	Grammar Error Count
GLAT	the only country country.	1	0.7604	3
SynNAT	the only country in the world.	1	0.9482	2
GLAT	she did for three years.	1	0.8599	2
SynNAT	she did it for three years.	1	0.9817	2
GLAT	all data avg	0.4824	0.4804	7.4690
SynNAT	all data avg	0.4934	0.4926	7.3787

5 结 论

本研究重新探讨了依存句法信息增强源端是否能够提升 Fully NAT 的性能. 通过将源句的丰富依存句法结构显式编码到 Fully NAT 模型中, 我们提出的 SynNAT 模型显著提高了翻译质量, 同时不会产生明显的解码速度损失. 实验结果表明, 该方法在多个翻译基准测试中表现出色, 显著提升了 Fully NAT 模型的翻译性能, 验证了利用 GCN 对依存句法信息进行编码的有效性和潜力. 该创新方法不仅为 NAT 模型的发展提供了宝贵的新见解, 也为实现高效且高质量的非自回归翻译奠定了基础.

References:

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

[2] OpenAI. GPT-4 technical report. arXiv:2303.08774, 2024.

[3] Dubey A, Jauhri A, Pandey A, *et al*. The llama 3 herd of models. arXiv:2407.21783, 2024.

[4] Lee J, Stevens N, Han SC, Song M. A survey of large language models in finance (FinLLMs). arXiv:2402.02315, 2024.

[5] Cai TL, Li YH, Geng ZY, Peng HW, Lee JD, Chen DM, Dao T. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. arXiv:2401.10774, 2024.

[6] Gu JT, Bradbury J, Xiong CM, Li VOK, Socher R. Non-autoregressive neural machine translation. arXiv:1711.02281, 2018.

[7] Guo P, Xiao YS, Li JT, Ji YX, Zhang M. Isotropy-enhanced conditional masked language models. In: Proc. of the 2023 Conf. on Empirical Methods in Natural Language Processing. Singapore: ACL, 2023. 8278–8289. [doi: 10.18653/v1/2023.findings-emnlp.555]

[8] Xiao YS, Wu LJ, Guo JL, Li JT, Zhang M, Qin T, Liu TY. A survey on non-autoregressive generation for neural machine translation and beyond. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2023, 45(10): 11407–11427. [doi: 10.1109/TPAMI.2023.3277122]

[9] Ghazvininejad M, Levy O, Liu YH, Zettlemoyer L. Mask-predict: Parallel decoding of conditional masked language models. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: ACL, 2019. 6112–6121. [doi: 10.18653/V1/D19-1633]

[10] Gu JT, Wang CH, Zhao JB. Levenshtein Transformer. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 11181–11191.

[11] Xiao YS, Xu RY, Wu LJ, Li JT, Qin T, Liu TY, Zhang M. AMOM: Adaptive masking over masking for conditional masked language model. In: Proc. of the 37th AAAI Conf. on Artificial Intelligence. Washington: AAAI Press, 2023. 13789–13797. [doi: 10.1609/AAAI.V37I11.26615]

[12] Duan SF, Zhao H, Zhang DD. Syntax-aware data augmentation for neural machine translation. IEEE/ACM Trans. on Audio, Speech, and Language Processing, 2023, 31: 2988–2999. [doi: 10.1109/TASLP.2023.3301214]

[13] Mondal SK, Zhang HX, Kabir HMD, Ni K, Dai HN. Machine translation and its evaluation: A study. Artificial Intelligence Review, 2023, 56(9): 10137–10226. [doi: 10.1007/S10462-023-10423-5]

[14] Wan F, Li P. A neural machine translation method based on split graph convolutional self-attention encoding. PeerJ Computer Science, 2024, 10: e1886. [doi: 10.7717/PEERJ-CS.1886]

[15] Guo JL, Xu LL, Chen EH. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In: Proc. of the 58th Annual Meeting of the ACL. 2020. 376–385. [doi: 10.18653/V1/2020.ACL-MAIN.36]

[16] Zhang Y, Li ZH, Zhang M. Efficient second-order TreeCRF for neural dependency parsing. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. 2020. 3295–3305. [doi: 10.18653/V1/2020.ACL-MAIN.302]

[17] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907, 2017.

[18] Zhang Y, Zhang B, Li ZH, Bao ZY, Li C, Zhang M. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. arXiv:2210.12484, 2022.

[19] Ma XZ, Zhou CT, Li X, Neubig G, Hovy EH. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: ACL, 2019. 4282–4292. [doi: 10.18653/V1/D19-1437]

[20] Gu JT, Kong X. Fully non-autoregressive neural machine translation: Tricks of the trade. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). 2021. 120–133. [doi: 10.18653/V1/2021.FINDINGS-ACL.11]

[21] Guo P, Xiao YS, Li JT, Zhang M. RenewNAT: Renewing potential translation for non-autoregressive Transformer. In: Proc. of the 37th AAAI Conf. on Artificial Intelligence. Washington: AAAI Press, 2023. 12854–12862. [doi: 10.1609/AAAI.V37I11.26511]

[22] Wei BZ, Wang MX, Zhou H, Lin JY, Sun X. Imitation learning for non-autoregressive neural machine translation. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 1304–1312. [doi: 10.18653/V1/P19-1125]

[23] Libovický J, Helcl J. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: ACL, 2018. 3016–3021. [doi: 10.18653/V1/D18-1336]

[24] Wang YR, Tian F, He D, Qin T, Zhai CX, Liu TY. Non-autoregressive machine translation with auxiliary regularization. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI Press, 2019. 5377–5384. [doi: 10.1609/AAAI.V33I01.33015377]

[25] Guo JL, Tan X, He D, Qin T, Xu LL, Liu TY. Non-autoregressive neural machine translation with enhanced decoder input. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI Press, 2019. 3723–3730. [doi: 10.1609/AAAI.V33I01.33013723]

[26] Li F. Research of neural machine translation model and algorithm based on non-autoregressive decoding [MS. Thesis]. Nanning: Guangxi University, 2024 (in Chinese with English abstract). [doi: 10.27034/d.cnki.ggxiu.2024.000806]

[27] Li K. Research of neuron machine translation based on non-autoregressive method [MS. Thesis]. Chengdu: University of Electronic Science and Technology of China, 2013 (in Chinese with English abstract). [doi: 10.27005/d.cnki.gdzku.2023.003186]

[28] Cui YF. Research on non-autoregressive machine translation based on curriculum learning and enhancing network [MS. Thesis]. Changsha: Central South University, 2023 (in Chinese with English abstract). [doi: 10.27661/d.cnki.gzhnu.2023.001110]

[29] Qian LH, Zhou H, Bao Y, Wang MX, Qiu L, Zhang WN, Yu Y, Li L. Glancing Transformer for non-autoregressive neural machine translation. In: Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int'l Joint Conf. on Natural Language Processing. 2021. 1993–2003. [doi: 10.18653/V1/2021.ACL-LONG.155]

[30] Lee J, Mansimov E, Cho K. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: ACL, 2018. 1173–1182. [doi: 10.18653/V1/D18-1149]

[31] Jiang H, Cai GY, Tang XL. Improving iteration-based non-autoregressive language model with time step awareness. In: Proc of the 29th IEEE Int'l Conf. on Parallel and Distributed Systems ICPADS. Ocean Flower Island: IEEE, 2023. 2798–2801. [doi: 10.1109/ICPADS60453.2023.00385]

[32] Zhao GY, Wang J, Gao SX, Yu ZT. Non-autoregressive neural machine translation with contextual feature integration. Journal of Shaanxi University of Technology (Natural Science Edition), 2024, 40(3): 44–51, 83 (in Chinese with English abstract). [doi: 10.3969/j.issn.1673-2944.2024.03.006]

[33] Ziv A, Gat I, Le Lan G, Remez T, Kreuk F, Copet J, Défossez A, Synnaeve G, Adi Y. Masked audio generation using a single non-autoregressive Transformer. arXiv:2401.04577, 2024.

[34] You WJ, Guo P, Li JT, Chen KH, Zhang M. Efficient domain adaptation for non-autoregressive machine translation. In: Findings of the Association for Computational Linguistics: ACL 2024. Bangkok: ACL, 2024. 13657–13670. [doi: 10.18653/V1/2024.FINDINGS-ACL.810]

[35] Li JH, Xiong DY, Tu ZP, Zhu MH, Zhang M, Zhou GD. Modeling source syntax for neural machine translation. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Vol.1: Long Papers). Vancouver: ACL, 2017. 688–697. [doi: 10.18653/V1/P17-1064]

[36] Bastings J, Titov I, Aziz W, Marcheggiani D, Sima'an K. Graph convolutional encoders for syntax-aware neural machine translation. In:

Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing. Copenhagen: ACL, 2017. 1957–1967. [doi: 10.18653/V1/D17-1209]

[37] Chen HD, Huang SJ, Chiang D, Chen JJ. Improved neural machine translation with a syntax-aware encoder and decoder. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017. 1936–1945. [doi: 10.18653/V1/P17-1177]

[38] Ding L, Wang LY, Liu SY. Recurrent graph encoder for syntax-aware neural machine translation. Int'l Journal of Machine Learning and Cybernetics, 2023, 14(4): 1053–1062. [doi: 10.1007/s13042-022-01682-9]

[39] Chen KH, Wang R, Utiyama M, Sumita E, Zhao TJ. Syntax-directed attention for neural machine translation. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. New Orleans: AAAI Press, 2018. 4792–4799. [doi: 10.1609/AAAI.V32I1.11910]

[40] Duan SF, Zhao H, Zhou JR, Wang R. Syntax-aware Transformer encoder for neural machine translation. In: Proc. of the 2019 Int'l Conf. on Asian Language Processing (IALP). Shanghai: IEEE, 2019. 396–401. [doi: 10.1109/IALP48816.2019.9037672]

[41] Liu Y, Hou YX. Syntax-aware complex-valued neural machine translation. In: Proc. of the 32nd Int'l Conf. on Artificial Neural Networks. Heraklion: Springer, 2023. 474–485. [doi: 10.1007/978-3-031-44192-9_38]

[42] Zhang MS, Li ZH, Fu GH, Zhang M. Syntax-enhanced neural machine translation with syntax-aware word representations. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol.1 (Long and Short Papers). Minneapolis: ACL, 2019. 1151–1161. [doi: 10.18653/V1/N19-1118]

[43] Wu SZ, Zhang DD, Zhang ZR, Yang N, Li M, Zhou M. Dependency-to-dependency neural machine translation. IEEE/ACM Trans. on Audio, Speech, and Language Processing, 2018, 26(11): 2132–2141. [doi: 10.1109/TASLP.2018.2855968]

[44] Wu SZ, Zhou M, Zhang DD. Improved neural machine translation with source syntax. In: Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. Melbourne: IJCAI.org, 2017. 4179–4185. [doi: 10.24963/IJCAI.2017/584]

[45] Akoury N, Krishna K, Iyyer M. Syntactically supervised Transformers for faster neural machine translation. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 1269–1281. [doi: 10.18653/V1/P19-1122]

[46] Liu Y, Wan Y, Zhang JG, Zhao WT, Yu PS. Enriching non-autoregressive Transformer with syntactic and semantic structures for neural machine translation. In: Proc. of the 16th Conf. of the European Chapter of the Association for Computational Linguistics: Main Volume. ACL, 2021. 1235–1244. [doi: 10.18653/V1/2021.EACL-MAIN.105]

[47] Li JJ. Research on non-autoregressive neural machine translation method based on dependency relations [MS. Thesis]. Kunming: Kunming University of Science and Technology, 2023 (in Chinese with English abstract). [doi: 10.27200/d.cnki.gkmlu.2023.002734]

[48] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE Computer Society, 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

[49] Ba JL, Kiros JR, Hinton GE. Layer normalization. arXiv:1607.06450, 2016.

[50] Zhang C, Li QC, Song DW. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: ACL, 2019. 4568–4578. [doi: 10.18653/V1/D19-1464]

[51] Zhang B, Zhang Y, Wang R, Li ZH, Zhang M. Syntax-aware opinion role labeling with dependency graph convolutional networks. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. 2020. 3249–3258. [doi: 10.18653/V1/2020.ACL-MAIN.297]

[52] Hamming RW. Error detecting and error correcting codes. The Bell System Technical Journal, 1950, 29(2): 147–160. [doi: 10.1002/j.1538-7305.1950.tb00463.x]

[53] Kim Y, Rush AM. Sequence-level knowledge distillation. In: Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing. Austin: ACL, 2016. 1317–1327. [doi: 10.18653/V1/D16-1139]

[54] Ott M, Edunov S, Baevski A, Fan A, Gross S, Ng N, Grangier D, Auli M. FAIRSEQ: A fast, extensible toolkit for sequence modeling. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics (Demonstrations). Minneapolis: ACL, 2019. 48–53. [doi: 10.18653/V1/N19-4009]

[55] Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: A method for automatic evaluation of machine translation. In: Proc. of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia: ACL, 2002. 311–318. [doi: 10.3115/1073083.1073135]

[56] Kasai J, Pappas N, Peng H, Cross J, Smith NA. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. arXiv:2006.10369, 2021.

[57] Gu JT, Kong X. Fully non-autoregressive neural machine translation: Tricks of the trade. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. ACL, 2021. 120–133. [doi: 10.18653/V1/2021.FINDINGS-ACL.11]

[58] Helcl J, Haddow B, Birch A. Non-autoregressive machine translation: It's not as fast as it seems. In: Proc. of the 2022 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle: ACL, 2022. 1780–1790.

[doi: 10.18653/V1/2022.NAACL-MAIN.129]

[59] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2017.

[60] Graves A, Fernández S, Gomez F, Schmidhuber J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proc. of the 23rd Int'l Conf. on Machine Learning. Pittsburgh: ACM, 2006. 369–376. [doi: 10.1145/1143844.1143891]

[61] Shu R, Lee J, Nakayama H, Cho K. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. New York: AAAI Press, 2020. 8846–8853. [doi: 10.1609/AAAI.V34I05.6413]

[62] Kasai J, Cross J, Ghazvininejad M, Gu JT. Parallel machine translation with disentangled context Transformer. arXiv:2001.05136, 2020.

[63] Hao YC, He SL, Jiao WX, Tu ZP, Lyu M, Wang X. Multi-task learning with shared encoder for non-autoregressive machine translation. In: Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACL, 2021. 3989–3996. [doi: 10.18653/V1/2021.NAACL-MAIN.313]

[64] Geng XW, Feng XC, Qin B. Learning to rewrite for non-autoregressive neural machine translation. In: Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021. 3297–3308. [doi: 10.18653/V1/2021.EMNLP-MAIN.265]

[65] Huang XS, Pérez F, Volkovs M. Improving non-autoregressive translation models without distillation. In: Proc. of the 10th Int'l Conf. on Learning Representations (ICLR 2022). 2022.

[66] Chen XR, Duan SF, Liu GS. Improving non-autoregressive machine translation with error exposure and consistency regularization. In: Proc. of the 13th National CCF Conf. on Natural Language Processing and Chinese Computing. Hangzhou: Springer, 2025. 240–252. [doi: 10.1007/978-981-97-9437-9_19]

[67] Li ZH, Lin Z, He D, Tian F, Qin T, Wang LW, Liu TY. Hint-based training for non-autoregressive machine translation. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: ACL, 2019. 5708–5713. [doi: 10.18653/V1/D19-1573]

[68] Sun ZQ, Li ZH, Wang HQ, He D, Lin Z, Deng ZH. Fast structured decoding for sequence models. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 3016–3026.

[69] Ran Q, Lin YK, Li P, Zhou J. Guiding non-autoregressive neural machine translation decoding with reordering information. In: Proc. of the 35th AAAI Conf. on Artificial Intelligence. Virtually: AAAI Press, 2021. 13727–13735. [doi: 10.1609/AAAI.V35I15.17618]

[70] Song J, Kim S, Yoon S. AligNART: Non-autoregressive neural machine translation by jointly learning to estimate alignment and translate. In: Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021. 1–14. [doi: 10.18653/V1/2021.EMNLP-MAIN.1]

[71] Du CX, Tu ZP, Jiang J. Order-agnostic cross entropy for non-autoregressive machine translation. arXiv:2106.05093, 2021.

[72] Zhan JA, Chen Q, Chen BX, Wang W, Bai Y, Gao Y. DePA: Improving non-autoregressive translation with dependency-aware decoder. In: Proc. of the 20th Int'l Conf. on Spoken Language Translation (IWSLT 2023). Toronto: ACL, 2023. 478–490. [doi: 10.18653/v1/2023.iwslt-1.47]

[73] Gui ST, Shao CZ, Ma ZR, Zhang XS, Chen YJ, Feng Y. Non-autoregressive machine translation with probabilistic context-free grammar. In: Proc. of the 37th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 5598–5615.

[74] Huang F, Zhou H, Liu Y, Li H, Huang ML. Directed acyclic Transformer for non-autoregressive machine translation. In: Proc. of the 39th Int'l Conf. on Machine Learning. Baltimore: PMLR, 2022. 9410–9428.

[75] Saharia C, Chan W, Saxena S, Norouzi M. Non-autoregressive machine translation with latent alignments. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP). 2020. 1098–1108. [doi: 10.18653/V1/2020.EMNLP-MAIN.83]

[76] Ran Q, Lin YK, Li P, Zhou J. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. arXiv:2006.05165, 2020.

[77] Shannon CE. A mathematical theory of communication. The Bell System Technical Journal, 1948, 27(3): 379–423. [doi: 10.1002/J.1538-7305.1948.TB01338.X]

[78] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional Transformers for language understanding. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: ACL. 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

[79] Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 1966, 10(8): 707–710.

[80] Damerau FJ. A technique for computer detection and correction of spelling errors. Communications of the ACM, 1964, 7(3): 171–176. [doi: 10.1145/363958.363994]

[81] Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Communications of the ACM, 1975, 18(11): 613–620. [doi: 10.1145/361219.361220]

[82] Warstadt A, Singh A, Bowman SR. CoLA: The corpus of linguistic acceptability (with added annotations). 2019. http://hdl.handle.net/2451/6044/

[83] Warstadt A, Singh A, Bowman SR. Neural network acceptability judgments. Trans. of the Association for Computational Linguistics, 2019, 7: 625–641. [doi: 10.1162/tacl_a_00290]

[84] Naber D. A Rule-Based Style and Grammar Checker. Munich: GRIN Verlag, 2003.

[85] Fleiss JL. Measuring nominal scale agreement among many raters. Psychological Bulletin, 1971, 76(5): 378–382. [doi: 10.1037/h0031619]

[86] Dettori JR, Norvell DC. Kappa and beyond: Is there agreement? Global Spine Journal, 2020, 10(4): 499–501. [doi: 10.1177/2192568220911648]

附中文参考文献:

[26] 李锋. 基于非自回归方式解码的神经机器翻译模型及算法研究 [硕士学位论文]. 南宁: 广西大学, 2024. [doi: 10.27034/d.cnki.ggxiu.2024.000806]

[27] 李克. 基于非自回归方法的神经机器翻译模型及算法研究 [硕士学位论文]. 成都: 电子科技大学, 2023. [doi: 10.27005/d.cnki.gdzku.2023.003186]

[28] 崔玉峰. 基于课程学习和增强网络的非自回归机器翻译研究 [硕士学位论文]. 长沙: 中南大学, 2023. [doi: 10.27661/d.cnki.gzhnu.2023.001110]

[32] 赵光耀, 王剑, 高盛祥, 余正涛. 融入上下文特征提取的非自回归神经机器翻译. 陕西理工大学学报 (自然科学版), 2024, 40(3): 44–51, 83. [doi: 10.3969/j.issn.1673-2944.2024.03.006]

[47] 李佳佳. 基于依存关系的非自回归神经机器翻译方法研究 [硕士学位论文]. 昆明: 昆明理工大学, 2023. [doi: 10.27200/d.cnki.gkmlu.2023.002734]

附录 A. 歧义标注

A.1 明确“歧义”定义与标注目标

在翻译质量评估中, 歧义性是影响可读性和可理解性的关键因素之一. 为了构建有效的歧义性标注方案, 首先定义并区分不同类型的歧义.

词义歧义 (lexical ambiguity): 译文中某个词可能对应多种含义, 读者难以判断其确切含义.

结构歧义 (syntactic ambiguity): 译文中某些短语或句法结构可能存在不同的解析方式, 导致理解上的差异.

指代歧义 (referential ambiguity): 译文中的代词或指代成分 (如 he/she/they/it) 不清楚指向哪个实体.

语用歧义 (pragmatic ambiguity): 译文可能由于语气、态度或隐含信息的多重解读而产生歧义.

由于 IWSLT14 DE→EN 语料库为句子级别的翻译数据, 不包含足够的上下文信息. 因此, 语用歧义 (如语气、隐含含义等) 难以判断. 在本研究中, 我们仅关注词义歧义、结构歧义和指代歧义, 排除对语用歧义的标注.

A.2 人工标注歧义方法

标准的翻译歧义标注通常要求标注者同时查看源文本, 以判断目标语言的译文是否产生或保留了比源文本更多或更少的歧义. 然而, 由于本研究的标注人员不具备德语能力, 因此人工标注任务不依赖源文本, 仅针对目标语言文本进行歧义性分析, 这可能导致某些在源语与目标语对照时才能确认的歧义没有被识别到. 本研究从原始的 IWSLT14 DE→EN 测试集中随机抽取 200 条数据, 交由 3 名标注人员独立标注, 以评估标注一致性. 评估的歧义类型包括词义歧义、结构歧义和指代歧义. 为量化翻译文本中的歧义性, 要求标注人员以打分的方式进行标注. 具体而言, 标注人员要求在阅读翻译句子后, 评估其是否存在词义、结构或指代方面的歧义. 标注者需根据 1–3 分的评分标准来衡量词义、结构或指代歧义的程度, 只要句子中出现任何一种歧义, 就给出一个综合评分.

1 分: 完全无歧义, 词义、结构或指代清晰.

2 分: 中等程度的歧义, 词义、结构或指代存在歧义可能导致不同解读.

3 分: 严重歧义, 词义、结构或指代存在歧义导致句子难以理解.

我们使用 Fleiss' Kappa[85]来评估 3 位标注员对 200 条翻译数据的一致性, 计算结果为 0.64. 这一数值表明标注人员之间具有较高的一致性[86], 说明标注质量较为可靠.

A.3 LLM 标注歧义方法

在本文中, 我们运用了大语言模型 GPT-4o-mini 来评估句子翻译的歧义性, 以缓解人工标注的主观偏差. 与人工标注的区别是, 我们要求 LLM 对源文本充分参考, 此外人工标注给定的是一个综合评分, 而 LLM 被要求在"词义、结构、指代"这 3 个维度分别进行评估. 我们对原始的 IWSLT14 DE→EN 测试集中的所有句子进行标注, 并向大语言模型提供了提示, 如图 A1 所示. 我们将模型测得的评分首先在句子层面将词义、结构、指代这 3 个维度的评分求平均, 得到每个句子的综合分数; 随后, 再对所有句子的综合分数进行平均, 得到整份测试集的最终平均评分.

LLM-based prompt template for ambiguity detection

```
You are a professional linguist and machine translation evaluator. Your task is to assess the ambiguity in a translated sentence compared to its original source text.
### Ambiguity Types
Analyze the following source sentence (original language) and its translated version (target language), then evaluate ambiguity based on three aspects:
1. Lexical Ambiguity: Does any word in the translation have multiple possible meanings, making it unclear which meaning is intended?
2. Syntactic Ambiguity: Can the sentence be parsed in multiple ways due to its grammatical structure, leading to different interpretations?
3. Referential Ambiguity: Is there any pronoun or reference in the translation where it is unclear which entity it refers to?
### Evaluation Criteria
For each category, assign a score from 1 to 3:
- 1 (No ambiguity): The meaning is completely clear.
- 2 (Moderate ambiguity): The sentence has multiple interpretations and requires additional context to resolve.
- 3 (Severe ambiguity): The sentence is highly ambiguous, making correct interpretation nearly impossible.
### Example Evaluation 1
Source Sentence (German):
"Er sah den Mann mit dem Fernglas."
Translated Sentence (English):
"He saw the man with the binoculars."
Expected Output:
1. Lexical Ambiguity: 1 (All words have a clear meaning.)
2. Syntactic Ambiguity: 2 (The phrase "with the binoculars" could describe either "he" or "the man.")
3. Referential Ambiguity: 1 (No pronoun ambiguity.)
### Example Evaluation 2
Source Sentence (German):
"Sie gab ihrer Freundin das Buch, als sie ankam."
Translated Sentence (English):
"She gave her friend the book when she arrived."
Expected Output:
1. Lexical Ambiguity: 1 (All words have clear meanings.)
2. Syntactic Ambiguity: 1 (The overall sentence structure is straightforward.)
3. Referential Ambiguity: 2 (The pronoun "she" could refer to the person giving the book or the friend who received it. Additional context is needed to clarify who
arrived.)
### Example Evaluation 3
Source Sentence (German):
"Er setzte sich auf die Bank, um sich auszuruhen."
Translated Sentence (English):
"He sat on the bank to rest."
Expected Output:
1. Lexical Ambiguity: 2 (The word "bank" can mean a bench, a riverbank, or even a financial institution. Without further context, multiple interpretations are possible.)
2. Syntactic Ambiguity: 1 (The overall sentence structure is straightforward.)
3. Referential Ambiguity: 1 (No pronoun ambiguity.)

### Task
Now, analyze the following translation:
#### Source Sentence (Original Language):
"[SOURCE SENTENCE]"
#### Translated Sentence (Target Language):
"[TRANSLATED SENTENCE]"
Provide your evaluation using the same format, assigning scores for each category and explaining your reasoning.
Let's think step by step:
```

图 A1 译文歧义性评估提示

作者简介

张建新, 博士生, 主要研究领域为自然语言处理, 大模型应用, 非自回归生成, 机器翻译.

郭沛, 硕士生, 主要研究领域为自然语言处理, 大模型应用, 非自回归生成, 机器翻译.

李俊涛, 博士, 副教授, CCF 专业会员, 主要研究领域为自然语言处理, 大语言模型.

张民, 博士, 博士生导师, CCF 高级会员, 主要研究领域为自然语言处理, 机器翻译, 人工智能.