硕士学位论文

非自回归机器翻译模型的训练目标优化研究

作者姓名：伍烜甫

指导教师：冯洋 研究员

中国科学院计算技术研究所

学位类别：工学硕士

学科专业：计算机应用技术

培养单位：中国科学院计算技术研究所

2023 年 6 月

A Study on Improving Training Objectives of Non-autoregressive Machine Translation

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Philosophy

in Computing Application Technology

by

Xuanfu WU

Supervisor: Professor Yang Feng

Institute of Computing Technology, Chinese Academy of Sciences

June, 2023

中国科学院大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。承诺除文中已经注明引用的内容外，本论文不包含任何其他个人或集体享有著作权的研究成果，未在以往任何学位申请中全部或部分提交。对本论文所涉及的研究工作做出贡献的其他个人或集体，均已在文中以明确方式标明或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院大学有关收集、保存和使用学位论文的规定，即中国科学院大学有权按照学术研究公开原则和保护知识产权的原则，保留并向国家指定或中国科学院指定机构送交学位论文的电子版和印刷版文件，且电子版与印刷版内容应完全相同，允许该论文被检索、查阅和借阅，公布本学位论文的全部或部分内容，可以采用扫描、影印、缩印等复制手段以及其他法律许可的方式保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

机器翻译是自然语言处理领域的一种重要技术，其任务是利用计算机将句子从一种自然语言转换为另一种自然语言。随着近年来深度学习技术的发展，神经机器翻译在性能获得了显著提升，成为了机器翻译采用的主流技术。但神经机器翻译也存在诸多缺陷，例如其采用了自回归的模式建模译文每个词的输出概率，即模型输出概率需要依赖于之前输出的译文，这使得模型只能逐个词地串行的得到输出译文，而无法充分利用神经网络模型并行计算的特性，导致模型解码阶段的用时大幅增加。非自回归神经机器翻译是一种能够实现快速解码的神经机器翻译模型。在这一模型中，解码器仅依赖于其位置和源端的输入句子来得到输出译文的各个词，且各个位置的输出分布相互独立，不需要依赖其他位置的上下文信息。这一特性使得模型在解码阶段能够并行输出参考译文的各个词，从而能够在解码速度上带来显著的提升。

虽然非自回归机器翻译模型通过并行解码的方式成功提升了解码速度，但其在输出译文的质量上与自回归模型存在着较大的差距，具体表现是解码的译文中有着大量的重复和漏译错译现象。由于对同一个源端句子来说，存在着多种可行的翻译译文，而且译文的各个词之间存在着相互联系；而非自回归模型采用了各个词输出分布相互独立的假设，使得模型在训练时无法结合其他位置的输出信息，导致最终难以得到正确的译文。本工作主要从设计训练目标的角度入手，循序渐进又各有侧重的设计了对非自回归模型的训练进行改进的策略，具体包括了以下的 3 个部分：

1. 结合上下文信息的训练目标设计

在非自回归机器翻译模型中，预测时缺乏其他位置输出的信息是阻碍非自回归模型得到准确译文的一大因素。如果能够在训练函数中引入输出译文的上下文信息，就能够缓解模型缺乏这一预测信息的问题，从而提升模型的性能。本工作利用了一个基于上下文信息预测当前位置输出分布的先见解码器，将先见解码器解码输出译文的分布与真实分布之间的距离作为损失函数的一部分，从而使得模型的训练目标结合其他位置解码的上下文信息。实验结果表明利用这一结合上下文信息的训练目标能够带来 2-8 个 BLEU 值的提升。

2. 多样本保序回归的损失函数设计

非自回归机器翻译模型在训练阶段通常会采用最大化参考译文概率的训练目标函数。但这种训练方式使得模型仅拟合到一种参考译文，而未能考虑到输出译文的多样性，导致输出译文的概率与其评价指标并不一致。针对这一问题，本工作基于输出译文概率大小应当与译文质量相一致的假设，在训练过程中引入了一种多样本保序回归损失函数。这一损失函数通过约束真实分布中采样的多组译文的概率与其评价指标的排序一致性，来确保模型的输出分布尽可能接近真实分布。实验结果显示，本方法在较强的非自回归机器翻译模型上能带来一定

的提升，同时能够得到质量更优的多样化译文。

3. **神经网络评价指标引导的训练设计**

非自回归机器翻译模型在各个位置输出分布相互独立的特性，使得输出分布难以拟合考虑上下文信息的真实分布。一些工作直接利用基于 *n*-gram 的评价指标引导模型进行训练，但这种评价指标与实际的译文质量存在一定偏差。神经网络评价指标具有建模语义更准确等优势，本工作参考了神经网络评价指标的实现方式，将类似的网络结构的输出用作激励信号，并引导非自回归机器翻译模型的训练；同时，本工作对其用于训练时存在的探索空间过大和对流利度不敏感的问题进行了改进。实验结果表明，在约束采样空间和引入流利度信息后，利用神经网络评价指标训练能够对非自回归翻译模型带来 4 个 BLEU 值左右的提升。

关键词：神经机器翻译，非自回归机器翻译，上下文协同解码，保序回归，神经网络评价指标

Abstract

Machine translation is an important technology in the field of natural language processing. Its task is to use computers to convert sentences expressed in one natural language into sentences expressed in another natural language. With the development of deep learning technology in recent years, neural machine translation has achieved significant performance improvement and has become the mainstream technology adopted by machine translation. However, neural machine translation also has many problems. For example, it uses the autoregressive mode to model the output probability of each word, which means that the model predicts the output probability of each word in the translation based on the previously output translation. As a result, the model can only obtain the output translation word by word in a serial manner, without fully utilizing the parallel computing feature of the neural network model, leading to a significant increase in the decoding time during the model's decoding phase. Non-autoregressive neural machine translation is a type of neural machine translation model that can achieve fast decoding. In this model, the output distributions at different positions of the decoder are mutually independent and only depend on their position and the input of the source side, rather than on the output of other words. This characteristic allows the model to output each word of the reference translation in parallel during the decoding phase, which can bring significant improvement in decoding speed.

Although non-autoregressive machine translation models have successfully improved decoding speed through parallel decoding, there is a significant gap in decoding translation quality compared to neural machine translation models that use autoregressive decoding. Specifically, the decoded translation of non-autoregressive machine translation models contains a large amount of repetition and mistranslation, due to the existence of multiple feasible translations for the same source sentence and the interdependence between the words in the translated sentence. However, non-autoregressive models adopt the assumption that the output distribution of each word is independent of each other, which leads to a lack of contextual information in the output distribution, making it difficult to learn the true distribution of the training data. Moreover, the training objective function of non-autoregressive machine translation models exacerbates this problem by treating each word as independent. This thesis mainly focuses on designing training objectives and proposes progressive and targeted strategies to improve the training of non-autoregressive models, including the following three aspects:

1. **Design of training objective that incorporates contextual information**

In non-autoregressive machine translation models, the lack of contextual information in the output during prediction is a major obstacle to obtaining accurate transla-

tions. Therefore, an improvement strategy is to introduce the contextual information of the output translation into the training objective function, so that the output distribution learned by the model can contain information from other positions' contexts. This work utilizes a seer decoder that predicts the output distribution of a position given contextual information and incorporates the distance between the decoded output distribution and the true distribution as part of the loss function. This allows the training objective of the model to incorporate contextual information from other positions during decoding. At the same time, this work also designs strategies for each module in the training model and conducts experiments, showing that incorporating this contextual information-based training objective can bring a 2-8 BLEU improvement.

2. **Design of multi-sample isotonic regression loss function**

For non-autoregressive machine translation models, the training objective function usually maximizes the probability of the reference translation output. However, due to the diversity of correct translations, the translation with the highest probability output by the non-autoregressive model is often not the best quality translation in reality. To address this issue, this work proposes a multi-sample isotonic regression loss function based on the assumption that the output probability of a translation should be consistent with its quality. This loss function ensures that the output distribution of the model is as close as possible to the real distribution by constraining the consistency between the probability of multiple sampled translations from the true distribution and their evaluation metric ranking. Moreover, this work improves the loss function by considering the difficulty of fitting multiple reference translations for non-autoregressive machine translation models. The corresponding experimental results show that this method can bring about 0.3 BLEU value improvement in DA-Transformer and obtain higher quality and diversified translations.

3. **Training guided by network evaluation metrics**

The independence of outputs at each position in non-autoregressive neural machine translation makes it difficult to fit the true distribution that contains output context information. Additionally, existing *n*-gram based evaluation metrics have some bias towards actual translation quality, which makes it difficult to train non-autoregressive models to output optimal translations. This work refers to the network structure of BERTScore and attempts to guide the training objective of non-autoregressive models with evaluation metrics modeled by similar neural network structures. At the same time, corresponding improvement strategies are designed for the problems of large exploration space and insensitive fluency of evaluation metrics in neural network evaluation metric training. Experimental results show that by constraining the sampling space and introducing fluency information, training with neural network evaluation metrics can bring about a 4-point BLEU score improvement for non-autoregressive translation models.

Key Words: neural machine translation, Non-autoregressive machine translation, bi-directional decoding, isotonic optimization, neural evaluation metrics

目 录

第 1 章 引言 ····· 1
1.1 研究背景与意义 ····· 1
1.2 研究目标 ····· 2
1.3 研究内容 ····· 3
1.3.1 结合上下文信息的训练目标设计 ····· 4
1.3.2 多样本保序回归的损失函数设计 ····· 5
1.3.3 神经网络评价指标引导的训练设计 ····· 5
1.4 章节组织 ····· 6
第 2 章 非自回归神经机器翻译背景介绍 ····· 7
2.1 机器翻译 ····· 7
2.2 神经机器翻译 ····· 8
2.2.1 RNNSearch 模型 ····· 8
2.2.2 Transformer 模型 ····· 9
2.3 非自回归机器翻译 ····· 12
2.4 非自回归机器翻译的相关研究 ····· 13
2.4.1 知识蒸馏 ····· 14
2.4.2 迭代式模型建模 ····· 15
2.4.3 引入隐状态 ····· 16
2.4.4 上下文友好的建模方式 ····· 17
2.4.5 损失函数 ····· 19
2.5 本章小结 ····· 21
第 3 章 结合上下文信息的训练目标设计 ····· 23
3.1 引言 ····· 23
3.2 相关工作 ····· 24
3.3 背景 ····· 24
3.3.1 建模上下文信息的先见解码器 ····· 24
3.4 上下文信息的建模 ····· 25
3.4.1 模型结构 ····· 26
3.4.2 训练 ····· 26
3.5 实验结果与分析 ····· 29
3.5.1 实验设置 ····· 30

3.5.2 主要实验结果 ····· 31
3.5.3 实验分析 ····· 32
3.6 本章小结 ····· 35
第 4 章 多样本保序回归的损失函数设计 ····· 37
4.1 引言 ····· 37
4.2 相关工作 ····· 38
4.3 背景 ····· 38
4.3.1 保序回归 ····· 38
4.3.2 基于有向图的非自回归机器翻译 ····· 39
4.4 非自回归损失函数的保序回归损失 ····· 40
4.4.1 极大似然损失学习真实分布的局限 ····· 40
4.4.2 保序回归损失函数 ····· 41
4.4.3 保序回归损失函数在非自回归模型的应用 ····· 41
4.4.4 训练流程 ····· 43
4.5 实验结果与分析 ····· 44
4.5.1 实验设置 ····· 44
4.5.2 主要实验结果 ····· 47
4.5.3 分析实验 ····· 49
4.6 本章小结 ····· 52
第 5 章 神经网络评价指标引导的训练设计 ····· 53
5.1 引言 ····· 53
5.2 相关工作 ····· 54
5.3 背景 ····· 54
5.3.1 基于神经网络的机器翻译自动评价指标 ····· 54
5.4 基于评价指标的训练策略 ····· 54
5.4.1 模型设计思路 ····· 55
5.4.2 主要改进 ····· 56
5.4.3 训练目标 ····· 58
5.4.4 训练流程 ····· 59
5.5 实验结果与分析 ····· 59
5.5.1 实验设置 ····· 59
5.5.2 主要的实验结果 ····· 60
5.5.3 分析实验 ····· 61
5.6 本章小结 ····· 63

第 6 章 总结和展望 ····· 65
6.1 总结 ····· 65
6.2 展望 ····· 66
参考文献 ····· 69
附录 ····· 77
致谢 ····· 79
作者简历及攻读学位期间发表的学术论文与其他相关学术成果 · 81

图目录

图 1-1 本论文中各个研究的相互联系 ····· 4
图 2-1 RNNSearch[1] 的模型结构 ····· 9
图 2-2 Transformer 的模型结构 ····· 10
图 2-3 Transformer 模型中注意力的实现机制 ····· 11
图 2-4 非自回归翻译模型[2] 的实现 ····· 12
图 2-5 非自回归模型训练数据序列级知识蒸馏的过程 ····· 14
图 2-6 迭代式非自回归机器翻译模型示例 ····· 15
图 2-7 不同任务上的 CTC 损失函数 ····· 18
图 2-8 传统的非自回归模型与 DA- Transformer 模型 ····· 19
图 3-1 先见解码器的结构 ····· 25
图 3-2 利用上下文信息训练的非自回归模型 ····· 27
图 3-3 不同模型训练下先见解码器的不同掩码 ····· 27
图 3-4 WMT14 En→De 上 BLEU 值随超参 β 的变化 ····· 34
图 4-1 DA-Transformer 的解码器结构 ····· 39
图 4-2 保序回归损失函数训练时随采样数量变化 BLEU 值的变化 ····· 49
图 4-3 保序回归损失函数训练的模型在多样性解码任务上的结果 ····· 52
图 5-1 计算翻译质量的模型结构和分数计算流程 ····· 55

表目录

表 1-1 有错误的非自回归翻译示例 ····· 2
表 3-1 结合上下文信息损失函数训练实验的数据集信息 ····· 30
表 3-2 结合上下文信息训练目标的实验的主要结果 ····· 32
表 3-3 上下文信息目标函数训练在 WMT14 En→De 消融实验的结果 ···· 33
表 3-4 模型解码句子的词级别准确性和语言模型困惑度的对比 ····· 34
表 3-5 上下文信息目标函数训练在 WMT14 De→En 上的结果示例 ····· 35
表 4-1 保序回归损失函数训练实验中数据集的信息 ····· 45
表 4-2 DA-Transformer 引入保序回归损失函数后的性能对比 ····· 47
表 4-3 引入保序回归损失函数的 DA-Transformer 与其他基线方法的对比 48
表 4-4 保序回归损失函数训练时随排序指标变化 BLEU 值的变化 ····· 50
表 4-5 保序回归损失函数训练时随单峰和多峰权重变化 BLEU 值的变化 50
表 4-6 保序回归损失函数训练时采用不同采样策略的结果 ····· 51
表 5-1 几种利用评价指标训练模型的策略的结果 ····· 61
表 5-2 约束探索空间和引入语言模型的消融实验 ····· 62
表 5-3 利用评价指标训练的模型在多个评价指标上的结果 ····· 62

第 1 章　引言

1.1　研究背景与意义

机器翻译是指在不改变语义的条件下，利用计算机将用某种自然语言（即源语言）表示的句子转换到用另一种自然语言（即目标语言）表示句子的技术。作为计算语言学的分支，机器翻译融合了语言学、数学和计算科学的知识，其实现过程依赖语言知识的充分理解、数学模型的合理构建、以及计算技术的搞笑实现。随着信息化和全球化的进展，使用不同语言的人群沟通交流的机会和需求也日益增多，对翻译的需求也日益增多；而计算机算法和硬件的发展，也使得机器翻译在如今能够在各种场景下得到应用，从而极大的方便了人们的日常生活。

机器翻译技术最早兴起于上世纪 40 年代末，其也经历了多个阶段的演进。具体而言，机器翻译技术经历了基于规则的机器翻译，统计机器翻译和神经机器翻译三个阶段。基于规则的机器翻译是 20 世纪 50 年代到 90 年代的主流方法，其做法主要是通过语言学专家总结的翻译规则，以及少量人工编纂的词典，从而将源语言的词转换为目标语言的词，同时使得目标语言句子具有目标语言的词法、语法等方面的特点。但随着技术的发展，基于规则的机器翻译面临知识老化维护成本高、专家与程序员沟通成本高、规则之间冲突增加、以及无法充分利用大量翻译实例的问题。

取代经验主义方法的基于经验主义的机器翻译方法、统计机器翻译和神经网络机器翻译都属于经验主义方法。统计机器翻译是基于噪声信道模型建立起来的，其目标是使得在给定输入下参考译文输出概率最大化。统计机器翻译由于引入了数学模型，使得其可以利用翻译实例来学习知识，从而在翻译效果上有了较大的提升；但其也面临需要存储大量离散的语言模型数据、模型适用性受到一系列人为设定的数学假设限制等问题。随着深度学习技术的发展和硬件性能提升，自 2010 年起，基于神经网络模型的机器翻译开始取得突破性进展，逐渐取代了统计机器翻译成为了翻译领域的技术主流。

主流的神经机器翻译模型可以视做一个使用编码器——解码器结构[3] 的条件语言模型：在这一架构下，源端的句子通过一个编码器编码为包含了上下文信息的向量表示，而模型再根据这一向量表示，生成相应的译文输出。此时模型在训练时的目标函数是在训练集上的极大似然损失函数，即在给定源端输入和当前位置之前的参考译文的条件下，最大化预测出最大化生成参考译文的概率，而在解码时则是根据每个词输出的概率来逐个词串行的得到输出的句子。

就网络结构而言，较早期的神经机器翻译模型[3,4] 在解码器端采用的都是循环神经网络的结构，这种结构的特点就是其每个位置的输入都必须依赖于前一个状态的输出，这使得其一方面在训练和解码时的速度比较缓慢，另一方面也带来了长距离遗忘等问题。2017 年提出的 Transformer 模型[5] 不仅将编码器和解码器都替换为了前馈神经网络，还对注意力机制进行了改进，从而解决了循环神经

网络中训练低效和长距离遗忘等问题，大幅提升了生成模型的性能，因而后来成为了包括机器翻译任务在内的多种自然语言任务的基线方法，但此时的神经机器翻译模型仍然采用的是串行的解码方式。

由于 Transformer 结构带来了并行能力的显著提升，人们相应的联想到是否可以通过并行生成序列的方式来取代之前的自回归生成序列的模型框架，从而在显著提升模型的生成速度的同时，缓解自回归生成模型中因训练和解码不一致带来的曝光偏差[6] 等问题。最终，Gu 等[2] 提出了非自回归神经机器翻译模型，该模型在沿用 Transformer 模型结构的基础之上，先利用编码器的输出对生成的句子长度进行预测，再利用解码器对各个位置的输出词进行相互独立的解码，从而实现并行的解码输出。而从最终的实验的结果来看，该模型实现了 10 倍以上的解码速度的提升。

虽然非自回归模型带来了显著的解码速度提升，但其在翻译的准确度和流利度上却有一定的下滑，其表现在译文上就是出现了大量的漏译和错译现象。究其原因，是由于自然语言在表示句子含义时需要充分的结合上下文的信息，而非自回归模型各个位置输出独立的特性虽然使得模型能够并行得到所有位置的输出，但也使得模型在预测时无法得到其他位置的上下文相关信息，得模型利用极大似然估计去拟合真实分布的难度显著变大。而对于自然语言来说，同一个意思往往可以有多种等价的表示形式，相应的，对于同一个源端的句子也就存在着多种可行的译文；而词级别的交叉熵损失函数则惩罚各个位置与参考译文不一致的词，使得模型难以给各个位置的输出一个准确的损失函数。这一系列的问题使得非自回归机器翻译任务普遍面对着多峰性问题[2]。针对非自回归机器翻译中存在的多峰性问题，一系列的工作从训练数据、模型结构、训练方式、损失函数等角度来改善非自回归翻译的性能，并取得了一定的效果。而本论文主要是从非自回归任务难度加大，且缺乏合适的训练目标入手，探索几种更准确合理的非自回归机器翻译的训练目标，从而实现改善非自回归机器翻译任务的准确性。

1.2 研究目标

正如前文所述，虽然非自回归翻译实现了解码速度的提升，也显著的消除了曝光偏差，但与此同时在翻译性能带来了显著的下降，主要的表现是重复词的增多和一些关键词的缺失，表1-1中展示了标准非自回归模型中出现的两个错误的例子：

表 1-1 有错误的非自回归翻译示例

Table 1-1 Examples of non-autoregressive machine translation with errors

源端句子	今天下午我吃了披萨。	我们当然会有点生气。
参考译文	this afternoon I ate pizza. I ate pizza this afternoon.	of course, we got a little pissed up. and of course, we got a little angry.
模型输出	I ate I ate pizza.	of course, course got got got something.

表1-1中所列出的两个例子都有一个特点，就是每一个句子都能够找到不同的翻译，且这些翻译存在着一定的结构错位。以第一个例子为例，在第一个参考译文中，“this afternoon” 是在 “I ate pizza” 之前的，而在第二个参考译文中，“I ate pizza” 则是在 “this afternoon” 之前。由于非自回归模型在训练和解码时又仅仅依赖于各个词的解码位置和源端的输出，使得非自回归模型在预测下一个词的时候缺乏足够的上下文信息，也就导致了一些高频的表达形式重复在模型中输出的情况，如第一个句子中的 “I ate” 和第二个句子中的 “course” 和 “got”。这种输出一方面会使得输出句子的流利度显著下降，另外一方面也使得一些表示句子含义的实词（如第一个例子中的 “披萨”，第二个例子中的 “我们” 和 “生气”）无法正常输出，这样使得输出的译文在准确度上也显著下降。

上述问题是非自回归翻译中一种普遍存在的现象，Gu 等[2] 曾称之为多峰性（multi-modality ）问题。而这一问题是造成非自回归机器翻译模型在输出译文质量上弱于自回归模型的主要原因。而对这一问题，除了缺乏上下文造成的非自回归翻译任务本身的难度上升，以及非自回归模型训练时缺乏上下文信息以外，还与标准的非自回归模型所采用的词级别的交叉熵损失函数密切相关。非自回归模型的各个位置的交叉熵损失函数相互独立，这一方面与自然语言中上下文紧密联系的事实相矛盾，也给模型训练分布拟合真实分布带来了难度。具体来说：以一个简单的训练集 D_{train} = {(Thank you, Vielen Dank), (Thank you, Danke schön)} 为例，理论上模型应当拟合的分布是：

$$p(y|x = \text{Thank you}; \theta) = \begin{cases} \frac{1}{2} & y = \text{Vielen Dank} \\ \frac{1}{2} & y = \text{Danke schön} \end{cases} . \tag{1-1}$$

但采用交叉熵损失函数时，模型却拟合了如下的输出分布：

$$p(y|x = \text{Thank you}; \theta) = \begin{cases} \frac{1}{4} & y = \text{Vielen Dank} \\ \frac{1}{4} & y = \text{Vielen schön} \\ \frac{1}{4} & y = \text{Danke Dank} \\ \frac{1}{4} & y = \text{Danke schön} \end{cases} . \tag{1-2}$$

显而易见，这种各个词独立的分布显然是不符合实际的情况。这个例子表现了词级别极大似然损失函数与非自回归模型特性之间存在的矛盾。而本文中几个工作的的主要研究目标则正是从模型训练的目标入手，通过合理设计训练过程来缓解非自回归机器翻译中的多峰性问题，提升模型输出译文质量。下面将会对这些工作的研究内容做一个简要介绍。

1.3　研究内容

针对现有的目标函数训练下的非自回归模型存在的问题，本工作设计了不同的损失函数来加以改进。首先，考虑到词级别损失没有上下文信息的问题，本

工作引入了基于上下文信息预测的解码器来辅助非自回归机器翻译；其次，针对非自回归模型的输出分布拟合于参考译文的偏差，本工作引入了输入样本的保序回归损失函数来计算相应的损失；最后，鉴于非自回归模型本身在真实分布上的建模能力，本工作中探索了直接利用 BERTScore 来优化输出译文，引导非自回归模型进行训练。以上工作整体的关系如下图所示：

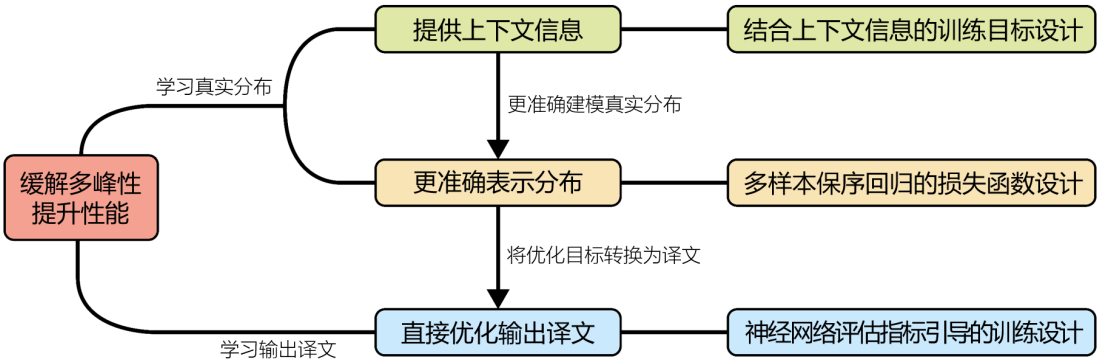


图 1-1 本论文中各个研究的相互联系

Figure 1-1 Overview of research works and their relationships

下面将对这些工作做一个详细的介绍。

1.3.1 结合上下文信息的训练目标设计

如前文所述，标准的非自回归机器翻译中所采用的交叉熵损失函数，使得模型各个位置的输出分布的损失函数仅仅依赖于模型的源端和目标端的词的位置，而没有考虑到其他位置的输出的信息，导致模型放大了存在错位和异位、但与参考译文在片段上接近的译文的损失，导致模型学到的输出分布与真实分布出现了偏差。如果能够使模型的训练目标能够考虑到其他位置的上下文信息，那么模型也就更容易学到一个符合真实数据分布的输出。基于这一思路，本工作中设计了一种结合双向信息的网络结构以及对应的训练策略，进而可以利用这一结构的输出来引导非自回归机器翻译模型的训练，从而缓解多峰性问题。

本工作在原有的非自回归模型的基础上引入了一个结合双向信息的网络结构，即先见解码器，并利用该网络模块的输出来校准非自回归翻译模型的输出分布。先见解码器以源端句子和目标端句子的其他词为条件，来建模目标端当前词的输出分布。在一个预训练好的非自回归解码器模型的基础上，首先，本工作用一定轮数的预训练先见解码器来冷启动先见解码器；然后，本工作使用非自回归解码器输出作为先见解码器输入来获得每个输出令牌的分布，并使用非自回归解码器和先见解码器之间的 Jensen-Shannon (JS) 散度作为训练目标。由于先见解码器的输出分布包含了整个非自回归模型输出句的信息，使得非自回归模型学会了考虑整个输出句子的信息来得到各个位置的输出，这样也就对非自回归机器翻译中的多峰性问题起到了缓解的作用。

1.3.2　多样本保序回归的损失函数设计

非自回归机器翻译的生成方式是并行的预测目标端句子的各个词，这种生成方式带来了两个显著的好处：一是模型各个位置的输出能够方便的通过并行方式生成，因而能够避免自回归解码过程中的多轮迭代，提升了解码的速度，二是非自回归模型的训练和解码的过程基本一致，从而避免了自回归模型中时常出现的曝光偏差现象。但这种生成方式也有一些显著的缺点，即模型的预测仅依赖于源端的输出，使得模型正确预测输出词的难度显著增加。而现有的非自回归训练方法常常是采用极大似然的方式去拟合确定的参考译文，由于训练信号的稀疏和模型能力的限制，这种学习方式会放大模型在一些质量较高但与原始译文有偏差的输出译文的损失，从而导致模型训练效果较差。特别的，极大似然估计仅仅要求训练集中参考译文的输出概率为 1，但译文多样性使得这种分布往往并不准确，这会进一步的影响非自回归模型的输出译文质量。

因此，本工作中利用非自回归模型预测过程并行，且训练过程与解码过程一致的优势，直接得到模型的多组输出译文，并利用这些译文的输出质量来引导模型进行训练，以缓解训练信号稀疏的问题。从最大熵原理的角度看，在给定信息的情况下，描述事实的最为合理的概率分布应当是熵最大的分布；而从翻译等自然语言生成的场景来看，给定源端的信息下存在着多种可行的输出是一种常见的现象，而这与极大似然估计的假设（即输出分布是熵为 0 的独热分布）是相悖的。虽说语言空间的复杂性使得语言的真实的分布无法被建模出来，但这一分布应当大体满足“译文质量越高，概率越大；译文质量相当，概率相当”的假设。因此，本工作采用了一种基于排序的损失函数，并通过这一式子，使得模型在最大化参考译文之外，能够去学习加大质量更高的译文的概率，从而使得非自回归模型学到更准确的输出分布。

1.3.3　神经网络评价指标引导的训练设计

前面两个研究分别通过提供上下文信息和更充分利用数据的方式，使非自回归模型的输出分布更接近真实分布。但非自回归模型本身的独立性特性与翻译任务中译文各个词的相关性的矛盾本身并没有被彻底解决，而这也使得模型分布始终无法拟合真实分布，而这一性能上界也通过数学方式得到了证明。解决这一问题可采用的有两种方式：一种是通过引入隐状态等建模方式来降低模型分布与真实分布的最小距离上界，而另一种方式则是直接对模型的输出译文进行优化。而将评价指标作为优化目标是一种常见的优化方法，最小风险训练和策略梯度算法等方法均可以应用于对这一指标的训练。现有的许多方法多是采用的基于 *n*-gram 的统计指标来引导模型的训练，但这种评价指标与真实的句子质量存在着差距。本工作则利用网络模型作为打分模型，并探索利用打分模型的分数来优化非自回归翻译模型。

参考 BERTScore 的模型结构，本工作中引入了一个对目标端句子的编码器。具体来说，本工作将非自回归模型的输出和参考译文的输出通过这一解码器之

后得到两两之间的距离，进而计算输出译文各个词的准确度和困惑度，进而得到相应的分值。而针对其在训练中的性能问题，本工作中针对这一指标在非自回归模型训练时存在的两个主要矛盾：探索与开发之间的矛盾、以及准确和流利之间的矛盾设计了一系列的改进的训练策略。

1.4 章节组织

本文的章节组织结构如下：

第一章介绍了非自回归机器翻译相关的研究背景和其存在的问题，并根据这些问题提出了本论文的研究目标以及相关的研究内容，并给出了论文的章节组织安排。

第二章介绍了神经机器翻译和非自回归神经机器翻译的具体背景知识，包括机器翻译的相关背景知识，神经机器翻译的基本模型架构和实现方式，非自回归翻译模型的基本实现，以及针对非自回归翻译模型中的问题所进行的一系列改进。

第三章介绍了结合上下文信息的训练目标的方法，重点介绍了针对非自回归模型训练时缺乏上下文信息的问题，如何通过引入一个结合上下文信息的网络结构来引导非自回归模型的训练。

第四章介绍了多样本保序回归的损失函数的方法，主要内容是针对极大似然估计与译文多样性之间存在的矛盾，结合非自回归模型的特性，通过从模型生成的译文中采集样本并计算保序损失函数，从而使得模型学到一个更精确的输出分布。

第五章则是介绍了神经网络评价指标引导训练的方法。本章重点是对非自回归模型分布拟合真实分布始终存在着一定的上限的问题，在参考 BERTScore 模型结构的情况下，通过神经网络来为评估模型的分值进而引导模型输出质量更好的译文。

第六章则是对全文的工作做出了总结，并对后续的研究方向进行展望。

第 2 章　非自回归神经机器翻译背景介绍

本文的研究内容是针对非自回归机器翻译中存在的训练目标的问题提出一系列对训练目标的改进。相应的，本章的内容是对非自回归机器翻译及其相关的神经网络基础架构进行介绍。本章首先将会对机器翻译有关的背景进行介绍；之后将会对神经网络机器翻译的几种常见的网络结构及其背后的原理进行介绍；接下来将介绍非自回归机器翻译的实现及其特点；最后将会介绍包括改进训练目标在内的对非自回归机器翻译进行改进的策略。

2.1　机器翻译

虽然机器翻译这一概念的历史最早可以追溯到 17 世纪，但其真正得以实现则与计算机的发明和数学领域的进步有着密切的关系。第一台计算机诞生后的 1947 年，美国数学家 Warren Weaver 发表了《翻译备忘录》[7]，提出了利用密码学的思路，将翻译过程视为一个解密过程，来实现利用机器进行翻译的工作方式，这也被认为是机器翻译诞生的一个标志。早期机器翻译的方法多是基于规则的一些翻译方法，具体来说包括了直接将源端句子一一映射到翻译端的方法，利用词法和句法分析等技术对句子进行复杂转换的方法，以及利用中间语言等方法。但由于 1966 年发布的 ALPAC 报告[8] 对机器翻译的前景持悲观看法，导致在相当长的一段时间里的机器翻译技术发展较为缓慢。

而到了 20 世纪 90 年代，随着冷战的结束和全球化的演进，对翻译的需求也逐渐增多。而与此同时，一些数据驱动的机器翻译方法也开始兴起：首先是基于实例的机器翻译系统，其通过从平行语料库中寻找到与源端句子相似的句子并将对应的目标端句子进行修改来进行翻译；而在这之后，一系列的统计机器翻译方法开始兴起。统计机器翻译系统包含了预测目标语言的语言模型，预测目标端向源端翻译的模型以及包含快速搜索算法的解码器。其中在语言模型方面通常是采用的是 n 元语法模型，而在翻译模型上，人们则提出了一系列的方案：其中较为经典的有 1993 年提出的 IBM 5 个翻译模型[9]，其通过概率表达式建模了诸如词对齐、词序、句子长度等语言信息。之后，Gale 等[10] 又提出了统计句子对齐模型，而再后来，又诞生了基于短语的翻译模型[11]。

随着深度学习技术的发展和硬件性能提升，自 2010 年起，基于神经网络模型的机器翻译[1,3–5] 开始取得突破性进展，逐渐取代了统计机器翻译成为了翻译领域的技术主流。如今神经机器翻译模型不仅在理论上取得了成功，且已经在多种场景下得到了应用，有力的促进了使用不同语言的人群的沟通理解。

2.2 神经机器翻译

现有的神经机器翻译采取的是编码器—解码器架构，即将源端句子先通过某种转换转变为一种表示形式，之后将前述的“表示”转变为输出的序列。而不同模型的区别则是在于编码器和解码器的不同设计。以经典的自回归机器翻译模型为例，设模型的输入为 $\mathbf{x} = (x_1, ..., x_m)$，而神经网络模型的编码器和解码器分别表示为 f_{Enc} 和 f_{Dec}，则编码器的输出可表示为：

$$h = f_{Enc}(\mathbf{x}). \tag{2-1}$$

与之对应的，解码器输出第 i 个词的概率表示为

$$P(y_i|y_{<i}, \mathbf{x}) = \mathrm{softmax}(f_{Dec}(h, y_{<i})). \tag{2-2}$$

其中，$\mathrm{softmax}(a_i) = e^{a_i}/\Sigma_j e^{a_j}$。

以上便是神经机器翻译模型的基本网络架构。而为了实现翻译的功能，上述模型可以在由平行语句对构成的训练集上采用极大似然损失函数加以训练，而在解码阶段则可以集束搜索等方式预测出质量最好的译文。

正如前文所述，在具体的结构设计上，神经机器翻译模型前期主要是基于循环神经网络结构，而后来的模型则主要采用的是前馈神经网络，而两者的代表则是基于循环神经网络的 RNNSearch 模型和基于前馈神经网络的 Transformer 模型。这两个模型都采用了注意力机制，即将输出的权重集中在最为重要的源端词上，以缓解对源端信息的遗忘问题。下面将会对这两个模型进行介绍。

2.2.1 RNNSearch 模型

RNNSearch[1] 是 2014 年提出的一种神经机器翻译模型，其采用了通用的编码器——解码器架构，源端的编码器采用了一个基于循环神经网络的双向循环神经网络[12]，而目标端则采用的是单向的循环神经网络。

而从具体流程来讲，源端输入的句子 $\mathbf{x}$ 在经过编码器之后得到了各个词包含了上下文信息的表示，而解码器端则通过与源端输出的上下文表示和历史的隐状态信息来给出相应位置各个词的输出的概率，整个模型的流程如图2-1所示：

RNNSearch 模型相较于之前的基于循环神经网络的机器翻译模型取得了显著的性能提升，其核心在于解码时采用了对源端信息计算对齐的注意力机制，RNNSearch 中的注意力机制具体包括了以下几个步骤：

(1) 将前一步的隐状态 s_{i-1} 和源端各个位置的隐状态 h_j 拼接，经过线性网络和 softmax 函数后，得到各个词的向量对上下文向量贡献权重 a_{ij}。

(2) 根据前面计算得到的权重 a_{ij} 和源端的隐状态得到上下文向量 c_i。

通过以上的注意力机制，RNNSearch 模型得到了句子在当前位置所需的上下文信息。而在接下来的解码器端，模型将 c_i 和前一步的隐状态 s_{i-1}，以及输出

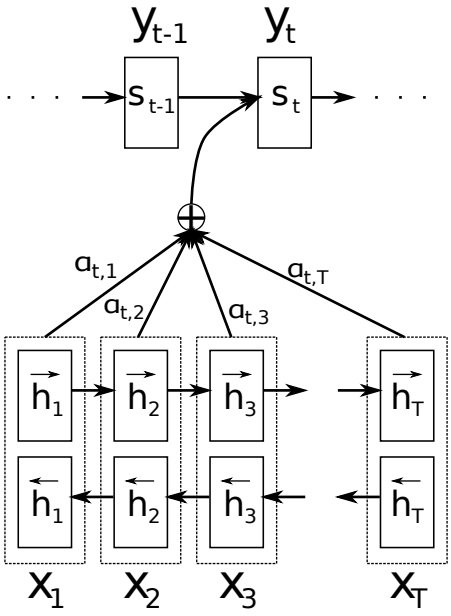


图 2-1 RNNSearch[1] 的模型结构

Figure 2-1 Model Structure of RNNSearch[1]

词 y_{i-1} 作为输入，通过循环神经网络得到此步的隐状态 s_i。最后，此步隐状态 s_i 和上下文向量 c_i 以及输出词 y_{i-1} 一起，得到输出词 y_i 的概率。

在上述流程当中，源端输入句子的信息不再存在一个固定向量之中，而是由源端的每个词通过双向门控循环单元学习到每个词和整个句子的信息后，存储在了一个属于词自身的向量中；而注意力机制也使得隐状态向量被解放出来。而从实验的效果来看，引入注意力机制能够改善长距离依赖问题，提升翻译效果。但由于其采用了循环神经网络，导致不同时刻的隐状态之间仍然存在着顺序依赖的关系，使得模型无法进行序列层面的并行训练，且解码效率较低。

2.2.2　Transformer 模型

Transformer[5] 是 Vaswani 等人于 2017 年提出的模型。与之前的模型类似，该模型也采用了编码器—解码器的结构，而其相较于前面所采用的模型的最大不同，在于其完全抛弃了前面所采用的循环神经网络，仅通过线性神经网络和注意力机制来完成模型的运算过程。Transformer 模型的整体结构如图2-2所示，源语言的句子通过词向量和位置向量的编码之后输入到编码器中，从而得到了相应的源端句子的上下文表示；而解码器端则基于解码器的输出（即参考译文当前位置之前的词）和编码器得到的上下文表示来对目标端相应位置的输出分布进行预测。

就更具体的编码器和解码器结构来说，编码器和解码器都采用了多层相同网络结构的堆叠，但其具体的网络结构又有所不同：编码器包含一个自注意力机器层和前馈神经网络层。而解码器则包含了三个模块：第一个是学习输出的句子信息的自注意力层，第二个是学习已经输出的句子与输入句子之间的相关性的编码器-解码器注意力层，第三个是前馈神经网络层。自注意力机制主要是为了

使得每个位置的词能够包含相应位置的上下文信息，编码器-解码器注意力层是为了使得目标端的各个位置能够对齐到源端相应位置的信息，而前馈神经网络层则能学到更丰富的表示。

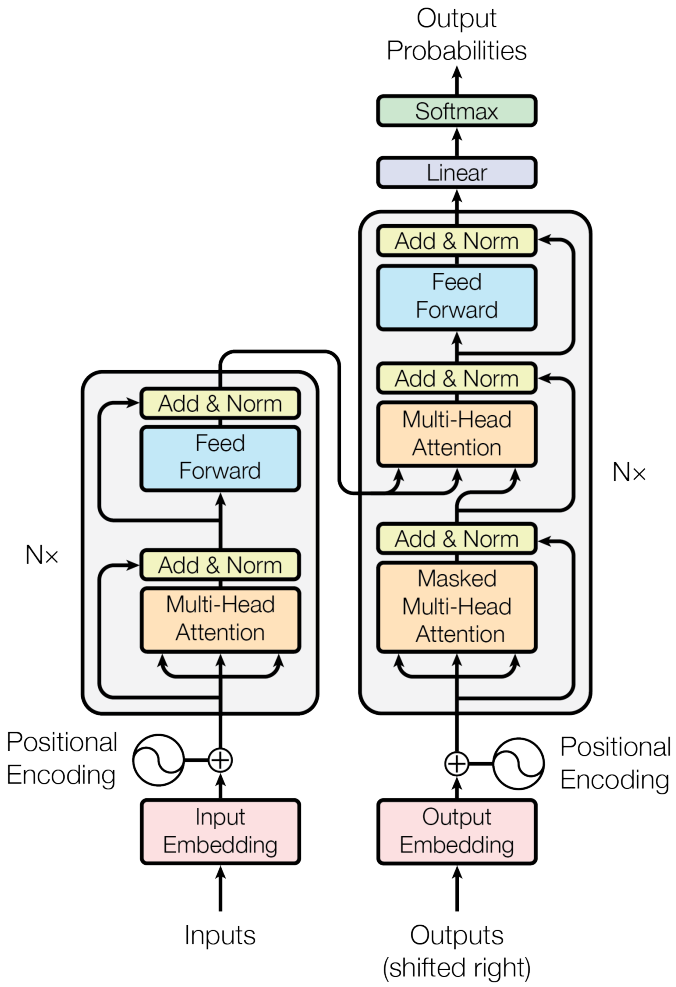


图 2-2 Transformer 的模型结构

Figure 2-2 Model Structure of Transformer Model

而就注意力机制而言，该模型也有许多贡献：一是引入了自注意力（Self-attention）机制，以提升模型提取句子自身信息的能力，二是首创了多头注意力（Multi-head attention）机制，使模型学习更充分。注意力模块的具体实现如图所示：

注意力模块有 Query （Q），Key （K）和 Value （V）三个输入。从图2-3中可以看到，在自注意力模块中，上述输入是相同的。而在编码器-解码器注意力模块，Q 为解码器前一层的输出，而 K 和 V 则为编码器的输出。需要说明的是，对于解码器的自注意力机制而言，为了避免解码时各个位置看到其之后的信息，模型引入了一个上三角的掩码矩阵 $\boldsymbol{M}$ ，该矩阵满足：

$$M_{ij} = \begin{cases} -\infty & i > j \\ 1 & i \leq j \end{cases} . \tag{2-3}$$

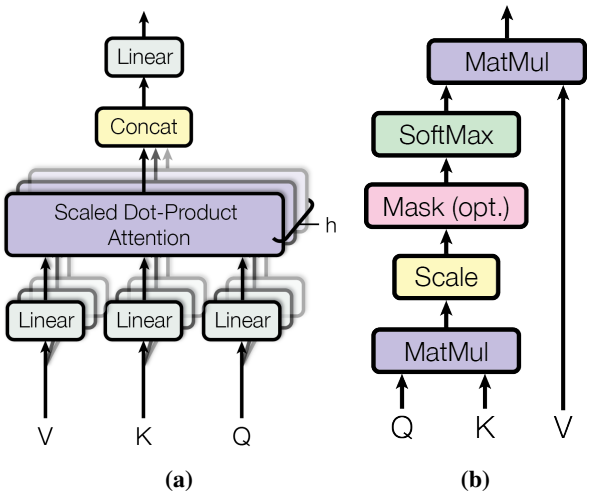


图 2-3 Transformer 模型中注意力的实现机制

Figure 2-3 Attention Mechanism of Transformer

而在多头注意力机制中，上述三个输入在经过线性网络层后被分割成了 H 个向量，并分别通过点乘来计算权重和输出向量，最后再将各个模块的输出拼接起来，在经过一个线性网络后得到输出。其具体表达式可由如下式子表示：

$$\text{Attention}_i(Q,K,V) = \text{softmax}(M \odot \frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_h}})(VW_i^V), \tag{2-4}$$

$$\text{head}_i = \text{Attention}_i(Q,K,V), \tag{2-5}$$

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_H)W^O. \tag{2-6}$$

需要说明的是，上式中 $\odot$ 表示逐个位置相乘。

除了上述的注意力机制模块外，在模型中还引入了线性前馈网络层，通过将注意力层中的输出映射到更大的空间来学习语言信息。其中包含了两层线性网络和一个激活函数，具体的表达式如下式所示：

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2. \tag{2-7}$$

除了编码器和解码器以外，在其他的位置 Transformer 也进行过一些特别的设计。由于 Transformer 模型的注意力机制对输入的绝对位置不敏感，因此为了确保模型的输入包含上下文的顺序信息，因此在编码器和解码器的词向量上需要加上相应的位置编码信息。具体的位置编码包括了利用余弦函数建模的位置编码和利用神经网络建模的位置编码等方式，通过这些方式 Transformer 模型获取了句子的上下文位置信息。

以上便是对整个 Transformer 的介绍，可以看到，上述模型结构较为简洁，训练时不需依赖之前各个位置的输出词，降低了计算的复杂度；且模型全部使用线性网络，能够并行训练，从而大幅提高了训练效率。而从翻译的结果来看，可以看到在模型包括机器翻译在内的多个 NLP 任务上实现了效果的提升。

2.3 非自回归机器翻译

正如前面所提到的，Transformer 模型利用前馈神经网络来建模整个模型结构，这一方面缓解了传统的神经机器翻译模型存在的长距离遗忘的问题，也使得翻译模型能够实现并行的训练。但与此同时，Transformer 模型仍然是采用了自回归生成的方法进行翻译，这意味着目标标记是按顺序逐词生成的。相应的，这一模型在解码生成目标句子时仍然会非常耗时。

为了缓解上述的问题，Gu 等[2] 提出了机器翻译的非自回归生成方法，即非自回归机器翻译（Non-autoregressive translation ， NAT ）。图2-4展示了该非自回归模型的结构。

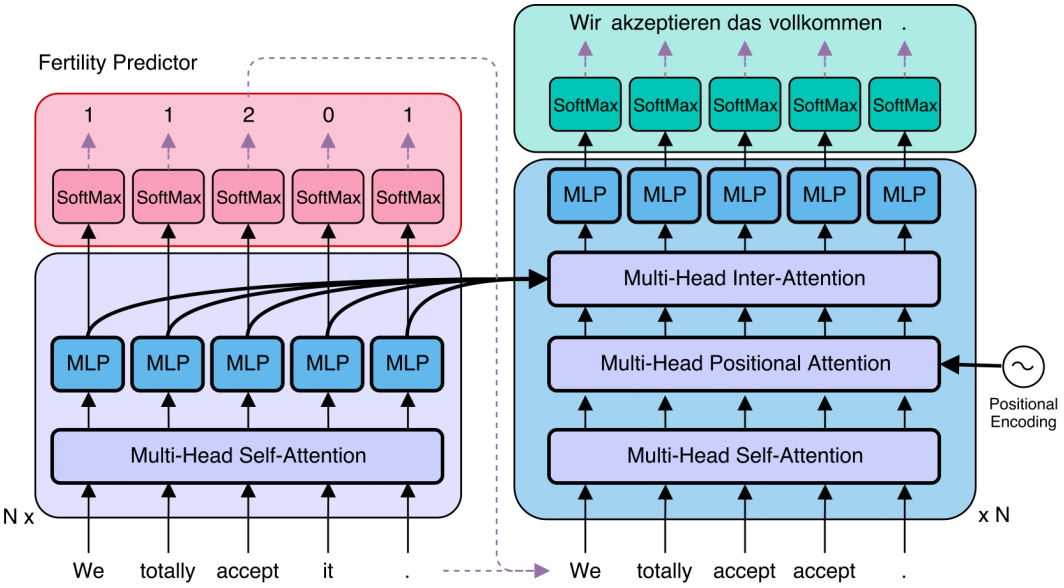


图 2-4 非自回归翻译模型[2] 的实现

Figure 2-4 Non-autoregressive Translation Model[2]

从图2-4中，可以看到非自回归翻译模型在结构上大体与 Transformer 模型相近，其结构上的区别在于以下一些地方：首先，模型的编码器后接入了一个预测源端各个词复制份数的预测器，模型通过这一模块相应的预测出解码器最终输出的词的个数，并根据其结果将源端的词向量复制到解码器的输入端；其次，由于非自回归模型的解码器是并行输出的，因此模型解码时的自注意力机制也不再需要相应的掩码矩阵。

在以上结构的基础下，在 Gu 等[2] 提出的非自回归机器翻译模型中，对源端的语言句子 $\mathbf{x}$，模型的输出概率被建模为：

$$P(\mathbf{y}|\mathbf{x};\theta) = P(T_{\mathbf{y}}|\mathbf{x};\theta)\prod_{i=1}^{T_{\mathbf{y}}} P(y_i|\mathbf{x};\theta). \tag{2-8}$$

其中，$T_{\mathbf{y}}$ 是模型输出句子的长度，而 y_i 是解码器输出的第 i 个词。

而就训练而言，作为基础的非自回归模型，Gu 等[2] 采用了词级别的交叉熵作为损失函数，设对训练集 $D = (\mathbf{x}, \mathbf{y})$，其中样本 $\mathbf{x} = (x_1, ..., x_m)$，而 $\mathbf{y} = (y_1, ..., y_n)$，则有：

$$L_{CE} = -\log P(T = T_{\mathbf{y}}|\mathbf{x}; \theta) - \sum_{i=1}^{T} \log P(y_i = \mathbf{y}_i|\mathbf{x}; \theta). \tag{2-9}$$

而在解码时，模型通过选取概率最大的词，从而并行的得到了解码译文中各个位置的词的输出。而 Gu 等[2] 又在此基础上进一步提出，可以通过在词的长度预测环节采集多个样本的方式，通过给出多个样本的解码并选取其中概率最大的句子的方式，来使得模型最终得到的译文是接近真实分布中概率最大的译文。

以上介绍了非自回归机器翻译的模型结构和生成方式，可以看到这种生成方式利用了 Transformer 模型的并行特性，将各个词的输出视为独立过程，因而可以并行的输出目标端的各个词，相应的大幅提升了解码的速度。但与此同时，由于非自回归模型对各个位置的输出的相互独立性做了较强的假设，导致其与语言生成任务上下文联系紧密的实际情况不符，导致解码的时候得到的输出分布往往与真实的分布之间存在着较大的偏差，带来了多峰性的问题。

而在 Gu 等[2] 的工作中，作者为了缓解这一问题，采取了句子级知识蒸馏的方法，具体来说就是模型的训练目标从原始的参考译文变为经过一个由训练集数据训练的自回归翻译模型解码后的译文。这种做法使得非自回归模型所要学习的目标分布变得更简单和更尖锐，从而使得非自回归模型更容易学习到。这种做法给非自回归模型带来了显著的提升，但其与自回归模型仍然存在着较大的差距。

2.4　非自回归机器翻译的相关研究

正如前文所提到的，非自回归机器翻译充分利用了 Transformer 模型的并行特性，对解码速度带来了 10 倍以上的提升，充分展现了非自回归模型对模型解码速度的显著提升；但与此同时，非自回归机器翻译模型也面临着严重的多峰性问题，这一问题使得非自回归模型输出的译文有着严重的漏译和重复现象。具体来说，自然语言在表示句子含义时需要充分的结合上下文的信息，而非自回归模型各个位置输出独立的特性虽然使得模型能够并行得到所有位置的输出，但也使得模型在预测时无法得到其他位置的上下文相关信息。与此同时，自然语言的表示的多样性使得训练集中存在着同一个句子的多种风格不同的表示，这使得非自回归模型各个位置的输出不得不去拟合多种不同的分布，使得模型最终的输出分布往往是几种分布的叠加，而不是某一种译文的表示，从而导致模型没有办法输出连贯的译文。

针对非自回归机器翻译的多峰性问题，一系列的研究通过进行知识蒸馏、引入迭代式的解码流程、引入隐状态、采用更易于建模上下文的输出结构以及设计

更合理的损失函数等角度提出了相应的改进策略，本节将会对这几种类型的改进进行具体介绍：

2.4.1 知识蒸馏

正如前文所提到的，同一个源端句子确实存在着多种参考译文，但非自回归模型在本身缺乏上下文信息的情况下表示这种多峰的分布是存在困难的。相应的，既然模型只需要学出某一种参考译文的表示，那么是否能够让模型学习一个其能学到的、对上下文信息依赖较少的单峰的目标呢？而这正是序列级知识蒸馏[13]改进非自回归机器翻译的原理。

在非自回归机器翻译中，序列级知识蒸馏通常是指通过自回归模型蒸馏等方式，使得非自回归模型能够拟合一个大体准确但更为简单的目标端句子，其方法的整体流程如下图所示：

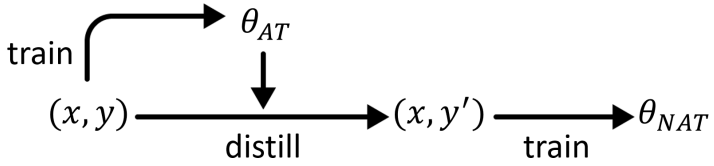


图 2-5 非自回归模型训练数据序列级知识蒸馏的过程

Figure 2-5 Process of sequence-level knowledge distillation for training data of non-autoregressive translation

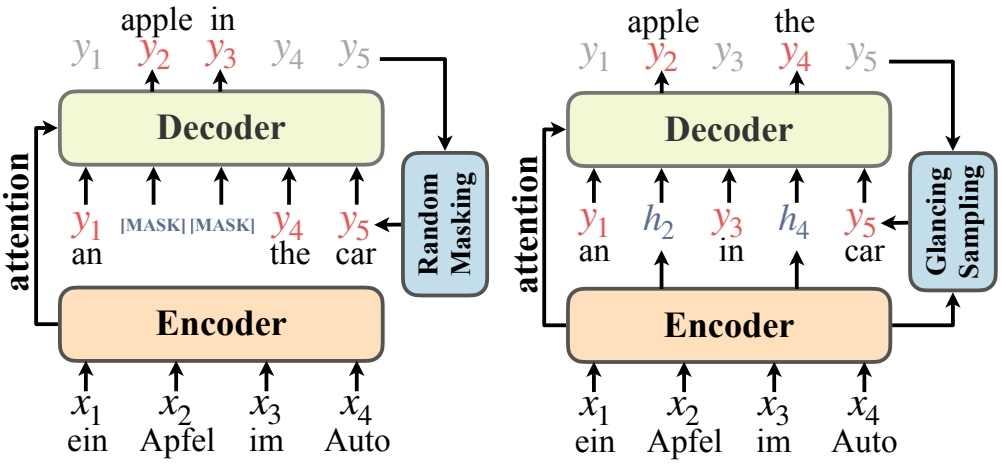
具体来说，序列级知识蒸馏的过程包括了三个步骤：第一步是利用训练集的数据得到一个自回归的翻译模型，第二步是利用训练好自回归的翻译模型将原始训练集的源端进行解码得到蒸馏后的目标端译文，最后一步则是利用蒸馏得到的目标端译文训练非自回归模型。由于自回归解码得到的目标端译文相较于原始的目标端句子，在表示上更为简单，多峰现象也更少，因此非自回归模型能够更容易的准确的拟合这种译文，相应的能够对非自回归模型的性能带来提升。

序列级非自回归模型一方面使得非自回归模型能够拟合到某一种准确的译文，但由于其学到的是一种比真实分布更为简单的分布，从而给非自回归模型的性能带来了一定的上界。相应的，针对序列级知识蒸馏这一方法，出现了一系列的分析和改进的工作：Zhou 等[14]通过数据复杂性和忠实度来分析蒸馏数据的特点并发现，虽然能通过降低复杂性和增加忠实度来简化训练数据，但更大更好的教师模型并不总是会产生更好的学生模型。而从数据层面，Ding 等[15]通过对评估指标分析发现，当 NAT 模型在蒸馏数据集上进行训练时，低频词的准确性会严重下降。Sun 等[16]通过采用 EM 算法，利用自回归模型蒸馏得到的输出和非自回归模型蒸馏得到的输出来迭代训练两个模型，从而实现了非自回归任务上输出译文质量的提升。Shao 等[17]则通过改进数据蒸馏的方式，得到了多组高质量且多样的蒸馏参考译文，并提出了相应的训练策略，让模型自适应的拟合其最容易学到的译文。

2.4.2　迭代式模型建模

导致非自回归翻译模型出现多峰性的问题除了译文句子本身存在多峰性之外，非自回归模型缺乏对上下文信息的建模能力也是一大原因。在自回归机器翻译中，模型是在已经给出的当前位置之前译文的基础上进行预测，这使得各个位置输出分布的多峰性大为降低；而在自回归分布中则不存在这样的信息，这使得模型的预测难度也有所加大。而迭代式模型则是通过模型在模型的预测阶段给出一部分上下文信息，从而使得模型的预测难度降低。

迭代式的非自回归模型最早是由 Ghazvininejad 等[18] 提出的。这一工作主要针对的是自回归模型串行解码存在的问题，设计了一种能够用有限的轮数得到解码译文的模型，即 CMLM 模型。这种模型结构的特点是模型解码器端会提供输出句子的一部分上下文：在训练阶段，对一个目标端参考译文长度为 n 的句子，模型先会从 1 到 n 中随机确定一个数字 k 作为待预测的词的数目，之后在解码器端的输入处随机选出 k 个位置，将参考译文置换为掩码，然后再训练模型在以上述被掩码替换的参考译文为输入的情况下来预测被掩码覆盖的词。而在解码阶段，模型共进行 T 轮解码，初始时模型将所有位置的输入均设为掩码，之后每轮解码出预测概率最大的 n/T 个词，并将解码出来的词作为下一轮的输入来解码剩余的词，最终，模型通过多轮的解码得到了最终的输出。由于解码的轮数为一固定值，因此在设定合适的轮数的情况下，其解码所需的速度相对于随解码时间随句子长度线性增长的自回归翻译模型仍能得到提升。总而言之，Ghazvininejad 等[18] 以牺牲解码速度提升为代价，实现了引入模型解码所需的上下文信息，从而提升了翻译模型的输出译文质量。



(a) CMLM 模型　　(b) GLAT 模型

图 2-6 迭代式非自回归机器翻译模型示例

Figure 2-6 Examples of iterative non-autoregressive machine translation model

而基于这一迭代式的模型，后续产生了一系列的对迭代式模型本身的改进[19,20]，而一些模型也采用了一些类似迭代式的输入建模方式来预测模型的输出。Qian 等[21] 提出的 GLAT 模型是其中的一个代表性的作品，其中的瞥见机

制也采用了类似的迭代式的模型思路，具体来说，在训练时模型需要经过两轮解码：第一轮解码时模型每个位置预测时没有上下文信息，相应的可以得到了解码的输出和参考译文不匹配的词数 L ；而在第二轮解码时，模型再随机选出 $\lambda \cdot L$ 个参考译文中的词与原来的输入端进行替换，并预测剩余的词；而瞥见系数 λ 的取值则随着训练过程的进行而逐渐降低。通过这样的课程学习的训练过程，模型逐渐的获得了结合上下文信息的并行的预测多个词的能力，从而显著的提升了非自回归模型的输出译文质量。

2.4.3 引入隐状态

在前面的迭代式的模型建模方式中，研究者们针对非自回归翻译模型在预测过程中缺乏上下文信息的问题，提出通过迭代式生成的方式，使得模型在训练和（或）解码阶段能够看到一部分的上下文信息，从而降低模型解码分布的多峰性，提升模型在预测时利用上下文信息的能力，从而改善模型的输出译文质量。而除了通过提供部分的目标端的参考译文信息以外，隐状态也是一种建模上下文信息的有效方式。具体来说，设要预测一个长度为 T 的句子，若将隐状态用字母 $\mathbf{z}$ 表示，则此时非自回归模型的输出分布为：

$$P(\mathbf{y}|\mathbf{x};\theta) = P(\mathbf{z}|\mathbf{x};\theta)\prod_{i=1}^{T} P(y_i|\mathbf{x},\mathbf{z};\theta). \tag{2-10}$$

从上述输出分布中可以看出，此时非自回归模型需要先根据模型输入的源端译文 $\mathbf{x}$ 来给出相应的隐状态 $\mathbf{z}$ 的概率，再在接下来根据 $\mathbf{x}$ 和 $\mathbf{z}$ 来计算出所需要解码的每一个词的概率。而由于解码时解码器端会得到一个固定的隐状态 $\mathbf{z}$，因此非自回归模型在解码输出各个词 $\mathbf{y}$ 时就获取了一定的有关待解码译文条件信息，从而使得模型输出分布的多峰性有所降低。

以上是通过隐状态改进非自回归机器翻译的基本原理，而就具体实施而言，不同的工作建模的方式又有所不同，具体的做法包括以下几种：

- 基于变分自编码器的隐状态建模

变分自编码器[22] 是一种通过神经网络学习到的隐状态的模型分布去近似真实空间的后验分布的做法，其针对真实空间中的后验分布难以学习的问题提出了一个近似真实分布的变分下界损失来进行训练。而在非自回归机器翻译中也有采用相关原理的研究：在 Ma 等[23] 中，作者采用了生成流模型[24] 的做法，通过一系列的可逆的线性变换将用标准高斯分布表示的向量变换为隐状态向量，而在训练时则使得模型在能够输出正确的译文的同时，模型输出的隐状态分布与后验分布之间的距离也能够尽可能的小。而在 Shu 等[25] 中，作者则是通过一个独立的 Transformer 编码器—解码器结构来建模隐状态的后验分布，也取得了不错的效果。

- 基于向量量化的隐状态建模

向量量化是一种将连续空间的向量映射到有限的矢量空间中的做法。以自

然语言中的词为例，设单词 y 映射到连续空间的表示为一个 n 维向量 $\text{repr}(y)$，若有一个包含了 K 个 n 维的隐状态向量的集合 $\mathbf{Q}$，则此时的隐状态就可以表示为：

$$z = \arg\min_{\mathbf{q}\in\mathbf{Q}} ||\,\text{repr}(y) - \mathbf{q}||_2. \tag{2-11}$$

而参考这一原理，一些工作将隐状态定义在取值有限的 $\mathbf{Q}$ 上，将计算隐状态的过程定义为一个有限类别的分类任务，并基于这一隐状态对后续的模型结构进行预测。在 CNAT 模型[26] 中，作者令模型将聚类结果作为非自回归模型解码器端的输入，使模型在有各个类别的先验信息的情况下进行预测；与此同时，作者利用神经网络模块建模了一个条件随机场以在给出源端句子的情况下预测参考译文所属词的类别，从而使得模型在没有后验的参考译文各个词类别的情况下也能得到正确的解码输出。而在 Latent-GLAT[27] 中，作者将预测隐状态的模块替换为了 Transformer 编码器结构，并采用了瞥见机制，从而进一步提升了模型的输出译文质量。

- 利用显式的语言特征建模隐状态

除了利用神经网络自身学习到的隐状态来建模输出分布之外，一些工作也采用了一些翻译中显式的特征来建模隐状态。具体来说，在 Ran 等[28] 中，作者利用 fast-align 工具[29] 抽取了模型各个位置的调序信息，并让模型基于源端的隐状态去拟合这一调序，从而在这一调序的基础之下进行后续的预测。而在 Song 等[30] 中，作者将则是利用 fast-align 和 GIZA++[31] 等统计学工具建模了目标端句子与源端的对齐，并将原有的对齐过程进行了分解以确保解码器的输入端融合了各个不同源端词的特征。

总而言之，利用隐状态建模的非自回归翻译模型，其核心都在于通过隐状态的结构提供了一些相关于参考译文的后验信息，从而降低了任务难度并缓解了多峰性，提升了模型的性能。

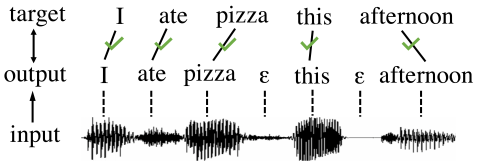
2.4.4　上下文友好的建模方式

正如前面所介绍的，一系列的研究通过引入序列级知识蒸馏、采用迭代式输出方式以及引入隐状态的方式来缓解模型建模多峰分布困难以及缺乏上下文信息等问题。但这些研究并没有对非自回归翻译模型的模型结构上并没有做出太大的变化。而标准非自回归模型[2] 的结构中也存在着一些显而易见的问题，比如模型在解码时需要提前预测参考译文的长度、缺乏对上下文关联信息的建模等，这些因素都显著的影响了非自回归翻译的输出译文质量。

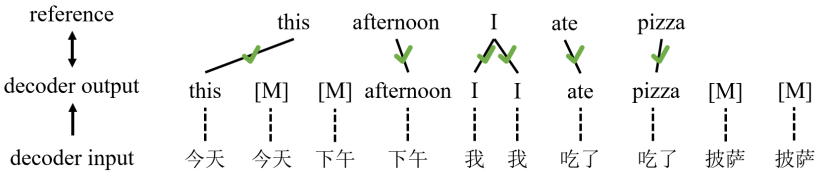
针对模型结构存在的这一问题，一些工作则采用了对上下文信息更为友好的网络建模方式。其中，CTC-NAT[32] 和 DA-Transformer[33] 是其中的典型代表，这两者都采用了对模型输出降采样的方式，并取得了显著的效果。下面将会对这两个工作进行介绍：

CTC-NAT[32] 是一种利用 CTC 损失函数[34] 进行训练的非自回归模型，这一损失本来是针对语音识别等任务中模型各个位置输入与输出的标签不存在严格

的一一对应的关系但存在相对的顺序关系而设计的，以下图中的语音识别任务为例，CTC 模型允许模型的输出存在一定的空白词 ϵ，只要模型最终解码得到的输出与参考的目标相匹配即可。



(a) 语言识别任务上的 CTC 损失函数



(b) 非自回归翻译任务上的 CTC 损失函数

图 2-7 不同任务上的 CTC 损失函数

Figure 2-7 CTC loss function in different tasks

而正如前面提到的，非自回归模型需要并行的得到各个词的输出，而在没有上下文信息的情况下，无论是预测解码句子的长度还是预测各个位置解码出来的词都会比较困难，因此也可以设计一些对解码的绝对位置进行松弛的做法。具体来说，该工作词表中引入掩码词 [M]，之后再令解码端的输入变为各个词逐词复制 n 倍，此时模型解码得到的句子的长度也就由待输出的译文变为了解码器长度的 N 倍，此时只要求模型在去除重复词和 [M] 的情况下得到正确的译文即可。这种建模方式不再要求模型能够强制在每个位置上输出相应的词，而是保持相对顺序即可，从而避免了长度预测和强制拟合参考译文等问题。

在前述 CTC 训练模型的基本原理之下，下面将会介绍其训练和解码的相应流程。具体来说，在训练阶段，该工作将所有能够通过降采样过程 $\mathcal{B}$ 得到参考译文的输出的可行输出的概率的和作为训练目标，其损失函数如下式所示：

$$L_\theta = -\log \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{y})} P(\boldsymbol{\pi}|\mathbf{x}). \tag{2-12}$$

从上式中可以看出，由于 $P(\boldsymbol{\pi}|\mathbf{x})$ 是解码出的各个词的乘积，因此模型在当前位置的损失函数的梯度就与其他位置的输出产生了联系，这样就使得模型能够学到一定的上下文信息。

由于能够通过降采样得到输出译文的路径过多，因此在 Graves 等[34] 提出算法时，也给出了利用动态规划的方式计算损失函数和相对于各个位置概率输出的梯度。而在解码时，模型只需要取每个位置上概率最大的词，并去除其中的重复词和掩码即可得到输出的译文。

从上述过程中可以看出，CTC-NAT 实现了模型的输出不再依赖于模型解码的绝对位置，而是各个词的相对顺序，其过程也隐式的学到了一些上下文信息。那么能否将各个位置的隐状态的相关性显式的表示出来呢？利用有向图机制来改进非自回归翻译的模型[33]（即 DA-Transformer ）给出了一种可行的途径。作为近期非自回归领域的一个重要工作，该研究设计了一种类似于 CTC 损失函数的损失函数，即确保模型的输出经过一定的下采样后的输出与目标端的译文一致，即：

$$L_\theta = -\log P(\mathbf{y}|\mathbf{x}) = -\log \sum_{A} P(\mathbf{y}|A,\mathbf{x})P(A|\mathbf{x}). \tag{2-13}$$

上式中，A 表示的是 Transformer 解码器的输出。

而相比起基于 CTC 损失函数中固定的降采样策略，Huang 等[33] 则对模型结构做了修改，在解码器输出后引入了一种基于有向图的计算方法，即利用一个有向的转移矩阵决定从第一个词到最后一个词的转移矩阵，如图2-8所示，采用不同的转移位置就可以得到不同的输出译文：

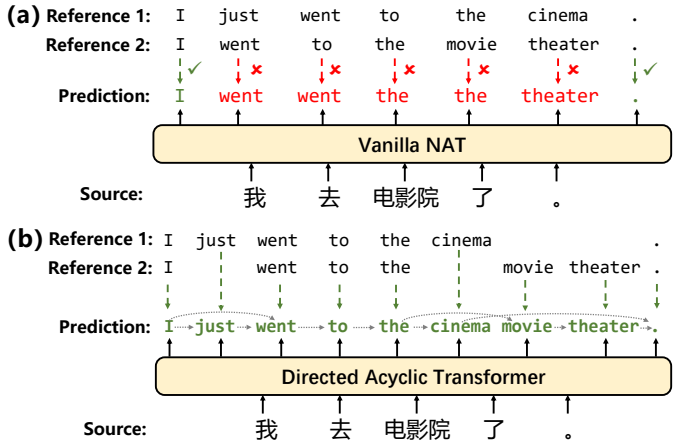


图 2-8 传统的非自回归模型与 DA- Transformer 模型

Figure 2-8 Vanilla NAT and DA- Transformer

根据上式中的采样策略，可以在训练时通过动态规划的方式计算出前式的损失函数，而在解码阶段则可以通过多种轻量级的搜索策略（如贪心解码、先行解码、维特比解码[35] 等）来实现降采样的过程。

以上工作中介绍了两种更容易建模上下文结构的模型，这两种模型都通过在输出后接入一个降采样模块的方式，从而避免了长度预测和预测分布对绝对位置的依赖，从而提升了模型建模上下文的能力，显著的改善了模型性能。

2.4.5　损失函数

前述的几种改进方法分别通过知识蒸馏、采用迭代式解码架构、引入隐状态和改进上下文模型结构等方式来对非自回归模型的训练目标进行改进。但这些

目标大体上仍然采用的是以参考译文为目标的极大似然的损失函数，而本章将会介绍几种对非自回归模型的损失函数进行改进的设计。

- 自适应对译文重排的训练目标

词级别损失函数中存在的一个核心问题是对位置的要求过于严格，其要求预测出来的词所在的位置与参考译文必须一致，即使对于一些和参考译文比较接近（比如编辑距离比较小的句子）也是如此，这样的过大的损失显然和实际的情况存在着偏差。于是，对损失函数的位置进行适当的松弛成为了缓解多峰性问题的一种途径。

在 AXE 模型[36] 中，作者对原有的交叉熵函数进行了一定的松弛，即只要求输出的译文尽可能的和参考译文保持单调对齐，而不是要求位置绝对的一一匹配，在具体实现时，作者通过动态规划算法来寻找在给定模型输出分布的条件下，能够达到最小的损失的单调对齐。而在 OAXE 模型[15] 中，作者则进行了进一步的松弛，不再要求模型的输出与参考译文保持单调的对齐关系，而是利用匈牙利算法[37]，直接寻找能够使得模型输出分布的交叉熵损失最小的参考译文的重排，同时，为了避免模型学到无效的对齐，作者在利用重排的译文进行训练之前，先利用参考译文对模型进行预训练。在多个任务上的实验证明，两种方法都能够实现一定的提升。

- 结合能量函数的非自回归翻译训练目标

基于能量函数的建模方式[38] 是将一个关于模型的输入和输出的函数作为训练目标的建模方法。而这种建模方法在非自回归机器翻译当中也有应用，在 Tu 等[39] 中，作者引入了一个自回归的翻译模型并固定模型的参数，然后将非自回归模型的输出输入到自回归的翻译模型中。在后续的训练中，模型将输出分布与自回归模型的互信息作为损失函数，通过减小损失函数来提升非自回归模型的输出译文质量。作者实验了多种将非自回归模型的输出输入到自回归模型的方法，并在一些小数据集任务上进行了一些实验，验证了模型的输出译文质量提升。

- 句子级的损失函数

由于非自回归能够直接解码出译文，而模型在训练阶段所采用的词级别的极大似然损失函数已经表现出与真实分布之间存在着严重偏差，因此，一种显而易见的思路是通过更高层级的指标进行训练。在 Shao 等[40,41] 中，作者利用增强学习中的策略梯度算法[42,43]，将评价指标的值（如 BLEU[44]）作为奖励来引导模型训练。而在 Shao 等[45] 中，作者针对直接采用句子级损失目标时将整个句子的奖赏均匀分配在各个词上所带来方差的问题，推导出了一种基于 *n*-gram 的匹配的损失函数，从而将模型输出分布的各个 *n*-gram 的概率期望与实际 *n*-gram 之间的距离作为损失函数以改进输出的准确性。相关的方法在 CTC[46] 和 DA-Transformer[47] 上均有实现，也取得了输出译文质量的提升。

需要特别说明的是，作为对非自回归模型的训练目标的改进研究，本论文中的工作一定程度上受到了这些改进损失函数的相关工作的启发。但在解决问题

的目标、训练目标以及训练策略的设计等方面，本工作与现有工作有着一定的差别。比如，第一个工作同时利用了前向和后向的神经网络知识，这与能量模型只利用单向的语言知识存在部分；第二个工作没有采用绝对的评价指标而是利用的对相对的评价指标的排序；而第三个工作则是直接针对现有工作中所采用的基于 *n*-gram 指标的改进进行的研究。以上的内容将在论文的后续章节中予以展开介绍。

2.5　本章小结

本章主要介绍了机器翻译的发展，神经网络机器翻译的几种常见的网络结构及其背后的原理；并介绍了非自回归机器翻译的实现及其特点，以及一系列从不同角度非自回归机器翻译进行改进的策略。在接下来的部分将会对本工作中对非自回归模型的训练目标所进行的改进做详细介绍。

第 3 章　结合上下文信息的训练目标设计

3.1　引言

正如前文中所提到的，作为机器翻译领域中一个新发展的方向，非自回归翻译通过同时并行解码的方式，压缩了其推理过程中花费的时间。然而，尽管在推理速度上取得了进步，但非自回归模型仍存在输出译文质量下降的问题，其具体的表现是出现了错译和漏译较多，语句不连贯且缺失关键信息，这些现象限制了其在工业界中的应用。

针对这一现象的一系列理论分析表明，非自回归模型在表示多峰性上的困难，是阻碍其输出译文质量提升的主要原因。首先，在自然语言中，同一句话往往会有多种意思相近但形式不同的表达方式，相应的，这也使得一句话往往有多种可行的译文。然而，非自回归模型在建模输出分布时，其却假设各个位置的输出分布是独立的，这就带来了模型分布和实际分布之间的巨大偏差。除此以外，通常对非自回归模型[2] 进行训练时，其采用的损失函数确实词级别的交叉熵损失函数，即训练时的各个位置损失函数仅仅依赖于源端译文和输出的位置。这样会使得一些位置有错位但上下文连贯的一些译文的损失函数显著偏高，导致多峰性问题进一步加重。

针对非自回归机器翻译中这一突出存在的问题，前人已经提出了一系列的针对性措施[14,15,17,18,21,23,25,26,30]。但是，这些研究或是降低了语料中的信息量，或是带来了更长的训练或解码时间，又或者是面临着在不使用外部特征建模工具（如 GIZA++[31]）时参考特征的能力有限的问题。因此，直接通过设计新的训练目标[14,15,36] 来缓解非自回归模型的多峰性问题成为了解决这一问题的可行途径。

本工作则提出了一种新的训练目标和训练策略来引导非自回归模型进行训练。具体来说，本工作在非自回归模型的基础上，又引入了一个可以提供上下文信息的解码器，即先见解码器。先见解码器是 Feng 等[48] 提出的一种解码器网络结构，这一解码器的功能是提供给出上下文信息情况下的当前位置的隐状态，从而为自回归机器翻译模型提供其预测时缺乏的下文信息知识。而本工作则将其应用在非自回归翻译场景中，即通过一个已经训练过的先见解码器来训练非自回归模型，结合非自回归模型本身的知识，来引导非自回归模型学到正确的上下文信息。在对先见解码器进行冷启动的基础上；本工作使用非自回归解码器得到的输出作为先见解码器输入来获得每个输出位置的分布，并引入非自回归解码器和先见解码器之间的 Jensen–Shannon (JS) 散度来作为训练目标。通过在训练时引入结合上下文的信息，本工作实现了在一定程度上纠正词级别极大似然损失的偏差的目标。在 WMT14 En↔De 和 WMT16 En↔Ro 数据集上进行的实验表明，本方法的输出译文质量相比标准非自回归模型[2] 提高了约 2-8 个 BLEU 值。

3.2 相关工作

缺乏输出译文的信息是影响非自回归机器翻译输出译文质量的一个重要问题，一系列的工作从不同的角度对其加以建模。迭代式的训练和解码方式[18,21]通过在训练和（或）解码时提供一些上下文的信息来降低偏差，而另一些工作[23,25,26,30]则利用隐状态来捕获整个输出译文的特征。

而除了上述工作以外，一些研究也从非自回归机器翻译模型中存在上下文信息缺失的角度，也有很多对这一问题的改进研究，如 Shao 等[45]中通过引入基于 n-gram 匹配的训练目标来使得各个词的损失函数结合相邻单词的特征，而 AXE 模型[36]和 OAXE 模型[15]则是利用非自回归模型在前期存在一定知识基础的情况下，通过引入自监督策略为非自回归模型找到最佳参考词顺序，从而缓解多峰性问题。此外，一些方法[18–21]通过在解码器输入端给出一部分参考译文的信息的方式，使得解码器端在训练时能够利用各个各个位置的上下文知识。这些方法通过结合上下文信息，并利用非自回归模型自身知识改进了非自回归模型的输出译文质量，但其也存在着上下文信息过于依赖参考译文的问题，而本工作将对这一问题加以改进。

3.3 背景

本工作是在非自回归模型的基础上，利用先见解码器实现的对上下文信息的建模。因此在详细介绍本工作之前，有必要对先见解码器的一些背景知识做介绍。

3.3.1 建模上下文信息的先见解码器

正如前文所提到的，先见解码器（Seer Decoder，Feng 等[48]）是作者为了改进自回归模型的输出分布提出的模型结构。在该工作中，作者针对自回归翻译在训练和预测时只有上文知识而缺乏下文知识的问题，额外引入了如图3-1所示的上下文解码器，即先见解码器。

先见解码器包含了三个模块，即前向子解码器，后向子解码器和混合层。其中，$N-1$ 层的前向子解码器和后向子解码器共享参数，其主要区别在于两者的自注意力机制掩码模块所遮盖的范围：前者是通过掩码置零当前词和之后的词中的信息，而后者则是是通过掩码置零当前词和之前的词中的信息。当模型的输入分别通过了前向解码器层与后向解码器层之后，输出的信息分别经过一个自注意力模块后再通过一个线性网络进行混合，从而得到了各个位置包含该位置之前和之后信息的隐状态向量。这个向量再经过后续的模块处理之后从而得到了先见解码器的输出。

从上述的结构中可以看出，先见解码器具有能够融合上文和下文特征的特点，且在融合的过程中相应位置能够与前文和（或）后文的信息进行充分的交互。除此以外，这一结构中前向和后向的模块独立进行，信息的融合过程只有一

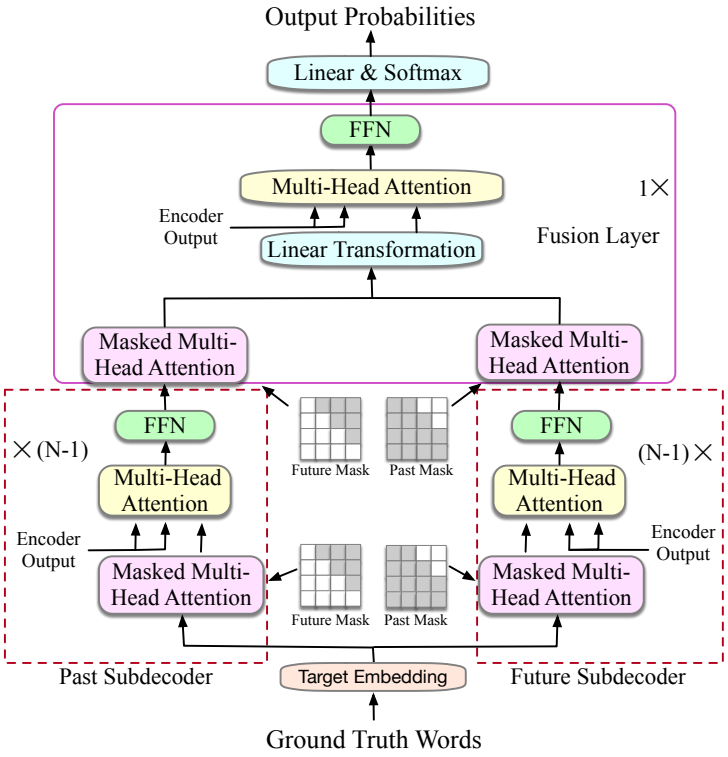


图 3-1 先见解码器的结构

Figure 3-1 Structure of seer decoder

层网络结构且放在最后，这样也使得模型避免了上下文信息的泄露导致的训练过程中的遗忘现象。而针对上述结构的特点，训练模块的损失函数如下式所示：

$$L_{seer} = -\sum_{i=1}^{N} \log P(y_i|\mathbf{x}, \mathbf{y}\backslash y_i; \theta_{seer}). \tag{3-1}$$

从上式中可以看出，通过强制先见解码器基于除去当前位置以外的词来预测之前的信息，先见解码器获得了基于当前位置所需的信息去利用上文和下文知识的能力，从而大幅提升了模型捕获上下文信息的性能，提升了机器翻译模型的输出译文质量。

3.4　上下文信息的建模

在 Transformer 模型中，对于一个句子的建模可以通过编码器结构的网络模型（如 BERT[49]）或者解码器结构的网络模型（如 GPT[50]）结构来实现。这两者学习的信息是有所不同的：对编码器模型而言，其学习的是整个句子中各个位置的特征；而对于解码器模块来说，其学习到的是一个语言模型，即在给出当前位置之前的词的情况下当前词的概率分布。而模型输出信息的不同也使得模型的结构出现了一定的变化，其中最主要区别在于，解码器结构的自注意力机制中存在着一个掩码的模块，即通过一个上三角的掩码矩阵置零掉当前词与其之后

的词的信息，从而确保模型每个位置预测的信息都是只依赖于当前词之前的词。而除了结构以外，这两种模型的训练方式也存在着差异。解码器可以自然而然的采用极大似然估计的训练方式，通过训练模型在给出之前的词的条件下的下一个词来进行预测，而对编码器而言，为了避免输入端信息泄露对模型性能的影响，常常会采用在输入端引入噪声（如对部分词掩码）等方式，并引导模型预测未知的信息来训练模型。

而对于非自回归模型而言，由于非自回归模型在解码器端的输入为固定的位置嵌入或对源端词向量的简单复制，其并不包含输出译文的上下文信息。而前文也提到，在非自回归模型的训练时，训练目标中往往缺乏上下文的信息，因此，在损失函数中引出译文上下文信息显得很重要。而接下来将会介绍本工作中引入上下文信息的神经网络结构。

3.4.1 模型结构

基于本工作所要实现的目标——即利用上下文信息缓解非自回归模型中词级别损失的多峰性问题，本文在训练时引入了先见解码器，由于其在建模上下文信息训练时，需要确保模型能够将上文和下文的信息都包含在模型的输出当中，而不是将简单将当前位置的词泄露给模型，因而适合用于非自回归机器翻译的场景下。而结合前述的先见解码器的强大能力，本工作中设计了如图3-2中的结构。

从图3-2中可以看出，在这一结构中，除了左侧的非自回归编码器和解码器外，还引入了如图右侧所示的先见解码器。新引入的先见解码器与非自回归解码器共享同一编码器，非自回归模型的预测输出再输入到先见解码器之后会得到一个包含上下文信息的输出分布，而在训练时，除了利用词级别的极大似然损失函数以外，也会去拉近先见解码器输出分布与非自回归模型输出分布的距离，从而使得模型的输出能够包含更多的上下文信息。此外，由于本方法是将非自回归模型的输出作为先见解码器的输出，这利用了非自回归模型本身学到的信息，也使得非自回归模型学习上下文的难度变得更低。

需要说明的是，由于非自回归翻译任务与自回归翻译任务在目标上存在着一定的差异，即在第 i 个位置，自回归模型的输出是要预测第 $i+1$ 个词，而非自回归模型则是预测第 i 个词，因此需要在先见解码器的前向子解码器和后向子解码器中采用了有所不同的掩码矩阵，以确保模型始终掩盖当前位置词的信息。具体的结构如图3-3所示。

3.4.2 训练

从前面的介绍中可以看出，虽然都需要共享同一个编码器的信息，但先见解码器和非自回归解码器这两者的任务目标完全不同，前者是为了在没有上下文的情况下给出翻译的整个句子，后者则是在给出输入句子的情况结合编码器信息预测在给定上下文的情况下各个位置的输出分布，也因此，两者的训练目标有着较大的差异。与此同时，由于两者存在着共享的网络模块，导致整个模型的训

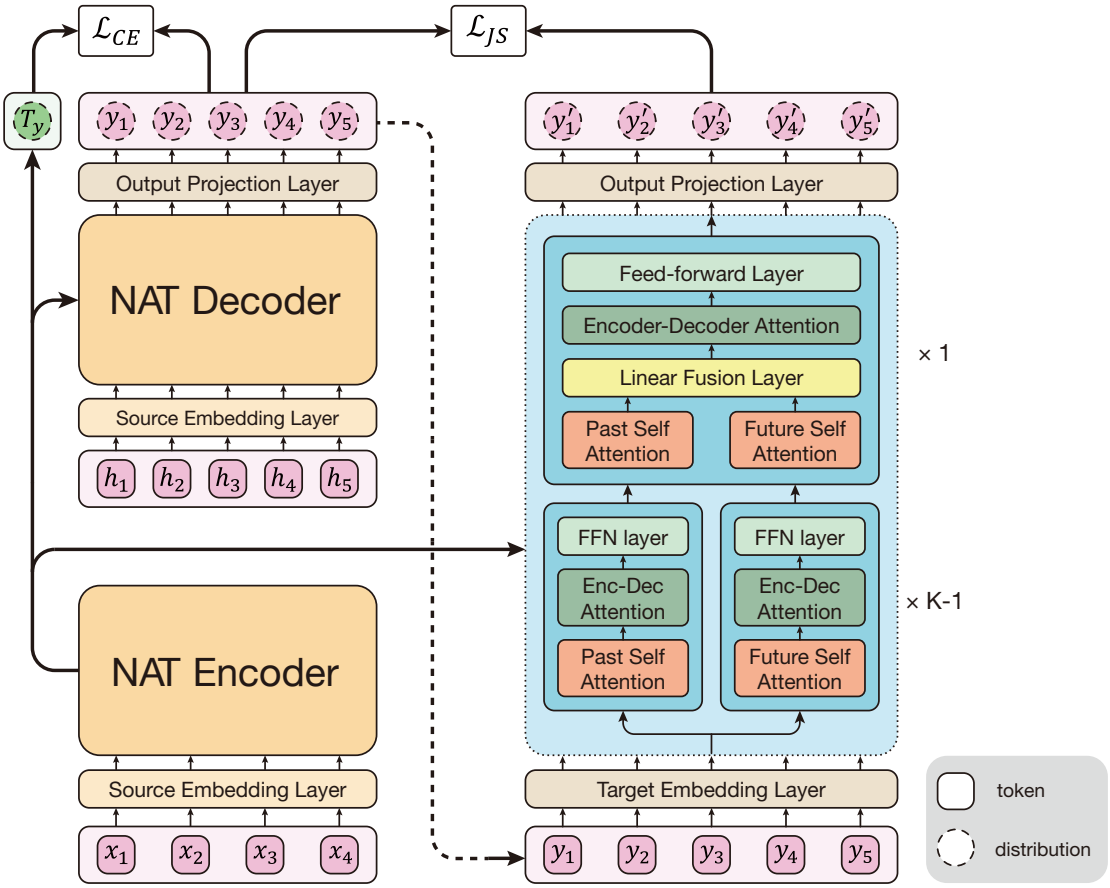


图 3-2 利用上下文信息训练的非自回归模型

Figure 3-2 Non-autoregressive translation model trained with context information

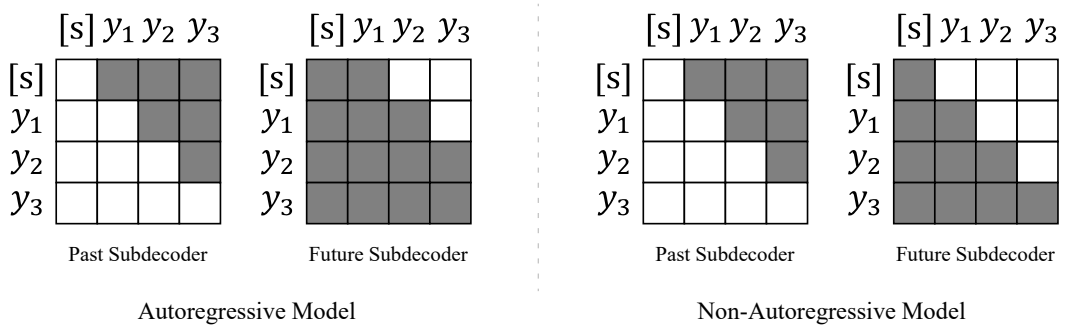


图 3-3 不同模型训练下先见解码器的不同掩码

Figure 3-3 Different masks in seer decoder with different models

练算法需要进行特别的设计，以确保模型能够学到相应的知识。下面将会对方法的训练目标和训练算法做出介绍。

3.4.2.1 对先见解码器的训练

正如前文所述，训练先见解码器的目标是为了让模型能够基于输入的上文和下文准确的预测出当前位置的词。相应的，采用的训练目标即是令模型在给定上文和下文的情况下来预测当前位置的词。设给定训练集 $D = \{(\mathbf{x}, \mathbf{y}^*)\}$，则有：

$$L_{seer} = -\sum_{i=0}^{T_{\mathbf{y}^*}} \log P(y_i^*|\mathbf{x}, \mathbf{y}\backslash y_i; \theta_{seer}). \tag{3-2}$$

上式中 $T_{\mathbf{y}^*}$ 为目标端句子 $\mathbf{y}^*$ 的长度。需要说明的是，相比起 Feng 等[48] 的训练目标函数，(3-2) 的区别主要是在训练的初始角标的选取上，前者是从第 1 个词开始，而后者则是从第 0 个词开始。

3.4.2.2 对非自回归解码器的微调

正如之前的一些研究所展示的那样，对于一个被预训练过的非自回归模型，其本身已经具有的根据源端输入生成相应输出的能力，只是由于模型训练时仅仅依赖于单个词的信息，使得模型没有办法综合上下文的输出给出各个位置的损失函数，从而带来了一定的偏差。而先见解码器的恰好能够在词级别极大似然损失缺乏上下文信息的情况下，根据其输入提供相应位置的上下文信息。因此，在利用先见解码器对非自回归解码器进行微调的时候，本方法选择采用非自回归模型解码器输出的词作为先见解码器的输入，并通过先见解码器得到的相应的输出分布来引导模型训练。具体来说，模型训练时的损失除了包含公式 (2-10) 以外，还包含了先见解码器输出分布和非自回归模型输出分布直接的 Jensen-Shannon 散度，具体的损失函数表达式如下式所示：

$$L_{NAT} = \beta L_{CE} + (1 - \beta) L_{JS}. \tag{3-3}$$

而由于非自回归模型各个词的输出分布独立，Jensen-Shannon 散度也可以进一步表示为：

$$L_{JS} = \sum_{i=0}^{T_{\mathbf{y}^*}} \mathcal{D}_{JS}\left(P(y_i|\mathbf{x}; \theta_{NAT}) \,||\, P(y_i|\mathbf{x}, \mathbf{y}/y_i; \theta_{seer})\right). \tag{3-4}$$

3.4.2.3 训练算法

本部分将会介绍本工作中利用前述的损失函数训练模型时的训练算法。正如前文所提到的，由于本工作中模块和损失函数众多，因此在训练过程中需要对训练算法进行专门设计。在整个模型的训练过程中，需要处理以下的三个问

题：首先是要确保非自回归模型本身具有一定的先验知识来引导模型的后续训练；其次是要确保非自回归模型在训练过程中能够给出正确的上下文知识；最后是在模型存在着多个训练目标的情况下，确保相应的模块都能学到相关的知识。

据此，在训练阶段，模型被拆解成了两个部分：一部分是原有的非自回归模型模块，即 θ_{NAT}，另外一部分是新增加的用于解码上下文信息的模块，即 θ_{seer}，需要说明的是，非自回归解码器和先见解码器共用的目标端词表和输出映射模块都属于 θ_{NAT} 的一部分。结合以上的定义，训练算法的设计如下所示：

算法 1 训练算法

Input: 训练集 $D = (X, Y)$, 训练过的非自回归解码器 θ_{NAT}，待训练的模块 θ_{seer}，训练步数 N，冷启动轮数 M

$i = 0$
将数据集 D 分割出相应的 batch 并计算出数目 K
while $i < N$ **do**
    for (**x**, **y**) in D **do**
        利用（3-2）来计算 L_{SEER}
        if $i > M \times K$ **then**
            利用（3-3）来计算 L_{NAT}
        end if
        利用 L_{seer} 来更新 θ_{seer}
        if $i > M \times K$ **then**
            利用 L_{NAT} 来更新 θ_{nat}
        end if
    end for
    $i = i + 1$
end while

从上述算法中可以看出，首先，针对方法中需要上下文知识的问题，算法中引入了一个预训练过的非自回归解码器进行训练；其次，为了确保先见解码器的性能，模型在利用先见解码器对非自回归解码器进行训练之前先对先见解码器进行了一定轮数的预训练；最后，为了确保各个模块能够学到清晰的训练目标，本方法中非自回归解码器的损失和先见解码器的损失分别被反传给了不同的模块。以上述算法为基础，本工作进行了后续的实验和一系列的分析工作。

3.5　实验结果与分析

为了验证上述方法对非自回归翻译模型输出译文质量的提升作用，本工作中进行了一系列的实验并进行了相应的分析工作。在本部分中，首先将会对模型所采用的实验设置（包括数据集、知识蒸馏、基线模型、评价指标和其他的实现细节等）进行介绍，之后将会展示非自回归模型的训练结果，接下来将会进行一

表 3-1 结合上下文信息损失函数训练实验的数据集信息

Table 3-1 Dataset statistics in training experiment combined with context information loss

数据集	训练集	验证集		测试集	
	样本数	名称	样本数	名称	样本数
WMT14 En-De	4500960	newstest2013	3000	newstest2014	3003
WMT16 En-Ro	608319	newsdev2016	1999	newstest2016	1999

系列的分析实验，包括对各个模块的消融实验，模型的输出分布的特征，以及对输出句子的统计分析和案例分析等。

3.5.1 实验设置

本节将会介绍本工作中有关实验的实验设置，包括训练数据、知识蒸馏情况、基线模型、评价指标等。

3.5.1.1 训练数据

就训练而言，本工作分别采用了如下表所示的两个翻译任务，即 WMT14 英语到德语双向（WMT14 En↔De ）和 WMT16 英语到罗马尼亚语双向（WMT16 En↔Ro ）。对于这些翻译任务对应的数据集，本工作采用了 32000 步的 BPE 模型[51] 来对语料进行预处理，同时对上述几个任务，本工作也共享了源端和目标端的词表。

3.5.1.2 知识蒸馏

在知识蒸馏方面，本工作对上述数据集均采用了通过知识蒸馏的数据进行训练。具体来说，对上述几个任务而言，本工作都采用了 Kim 等[13] 中的做法，即将非自回归训练时的目标端换成了一个通过自回归的模型解码后得到的译文。在本工作中，对各个任务都采用了 Transformer-Base[5] 模型训练得到的自回归翻译模型进行训练，并采用了大小为 5 的集束解码得到相应的目标端。

3.5.1.3 基线模型

考虑到非自回归模型的结构相似性，本工作相应的与两类非自回归模型进行对比：

- 基于标准 NAT[2] 的模型：其包括了 Vanilla NAT[2]、Bags-of-ngram 模型[14] 和 GLAT 模型[21]。
- 基于 mask-predict[18] 的模型：其包括了 CMLM 模型[18]、基于 CMLM 微调的 AXE 模型[36] 和基于 CMLM 的 OAXE 模型[52]。

3.5.1.4　评价指标

本工作中对翻译质量的评价主要是采用的是翻译中常用的 BLEU 指标。在本实验中，对 WMT14 英语到德语双向任务，评估时采用的是脚本 multi-bleu.perl[44] 来对模型指标进行测量；而在 WMT16 英语到罗马尼亚语双向中，则采用的是 sacreBLEU[53] 对译文质量进行测量。

3.5.1.5　其他的实现细节

在模型参数方面，在本实验中，为了平衡 NAT 解码器和先见解码器的能力，本工作中训练了一个具有 11 个编码器层、2 个非自回归解码器层和 2 个先见解码器层的模型。对于所有的训练任务，内部模块的参数遵循 Transformer-Base 模型的结构，词表的维数为 512 维，中间层隐状态维数为 2048 维，而注意力机制的头数为 8。源端和目标端共享词嵌入模块，且两个解码器的解码器输出投影层与目标嵌入层共享相同的参数。此外，本工作在所有数据集中将非自回归编码器和解码器的 dropout 概率设置为 0.3，将先见解码器中的解码器设置为 0.2。

而在训练阶段，本工作都是在一个已经采用标准 NAT 模型[2] 的基础上进行预训练来完成。在训练的批处理上，本工作每次对 WMT14 En↔De 使用 128000 个词，对 WMT16 En↔Ro 使用 64000 个词来微调模型。学习率在前 10000 步中为 $2e-4$，并随着平方根倒数时间表逐渐衰减。而优化器方面，本实验采用的是 Adam 优化器[54]，其中 $\beta=(0.9,0.98)$ 。而就超参数而言，本工作将对先见解码器训练中微调时控制权重的超参数 β 分别设置为 0.5（WMT14 En→De ）和 0.6（其他任务）；而对先见解码器冷启动的训练轮数 M 则分别为 2（WMT14 En↔De ）和 8（WMT16 En↔Ro ）。而在评估质量时，本工作对 5 个最佳检查点进行平均，并相应的设置 5 个长度候选。

以上的模型和训练过程的代码实验都是在基于 Pytorch 框架[55] 下的 fairseq[56] 工具包上完成。

3.5.2　主要实验结果

下表展示了主要的实验结果：

表 3-2 结合上下文信息训练目标的实验的主要结果

Table 3-2 Experiment results in experiments of training with context information

模型		WMT14		WMT16	
		En→De	De→En	En→Ro	Ro→En
Vanilla NAT	NAT-FT[2]	17.69	21.47	27.29	29.06
	Bags-of-ngram[14]	20.90	24.61	28.31	29.29
	GLAT[21]	**25.21**	**29.84**	**31.19**	32.04
CMLM	CMLM[18]	18.05	21.83	27.32	28.20
	CMLM+AXE[36]	23.53	27.90	30.75	31.54
	CMLM+OAXE[52]	**26.10**	**30.20**	**32.40**	**33.30**
Vanilla NAT(ours)		19.28	22.11	29.02	29.67
Vanilla NAT w/ seer		**25.36**	**29.34**	**31.97**	**31.71**

从上表中可以看出，首先，相比起进行微调训练之前的基线模型，即 Vanilla NAT (ours)，利用上下文信息辅助训练的模型实现了超过了两个点以上的提升；其中，在数据集规模较大的 WMT14 英语到德语上，本方法的模型分别实现了 5.94 和 7.23 个点的提升，这充分说明了本方法在较大的数据集上的显著提升。

除此以外，本工作还利用上下文信息辅助训练的模型做了一系列的微调实验，发现方法在输出译文质量上能够显著的超过 Vanilla NAT 和 CML M 这两个基线模型，同时也能够达到比其他的改进训练的方法更强或者相当的水平，这也充分说明了在解码端提供上下文信息在改进模型输出译文质量上的有效性。

3.5.3 实验分析

本节将会对本工作中所进行的实验及其结果进行分析。

3.5.3.1 消融实验

由于本实验中所涉及的模块众多，为了能够进一步的理解上下文信息辅助训练的方法中各个模块的重要性，本工作中进行了一系列的消融实验。就模块而言，本工作中分析了双向协同解码与仅使用前向解码和仅使用后向解码之间的差别；而就先见解码器引导的上下文信息的角度，本工作中分析了采用参考译文作为先见解码器输入和采用模型输出作为先见解码器的输入对模型输出译文质量的影响。本工作中采用了标准 Transformer 结构（即 6 层编码器和 6 层解码器），选取最优的检查点来得到相应的参考译文。表3-3中展示了实验结果：

表 3-3 上下文信息目标函数训练在 WMT14 En→De 消融实验的结果

Table 3-3 Results of ablation studies in WMT14 En→De

先见解码器		先见解码器输入		BLEU
前向	后向	NAT 模型输出	参考译文	
√	√	√		**23.60**
√		√		21.31
	√	√		21.74
√	√		√	22.01

表3-3为在 WMT14 En→De 任务上所进行的消融实验的主要结果表，表中对比了在控制其他模块不变的情况下，调整先见解码器或更改先见解码器的输入所带来的不同结果。

首先，对先见解码器而言，本实验通过控制注意力机制的掩码从而得到了只依赖上文和只依赖下文的解码器，并将其与双向的解码器进行对比；而从实际的结果来看，如果分别采用仅依赖上文的解码器和仅依赖下文的解码器进行微调，模型的输出译文质量虽然也能够得到提升，但其相比起原始的上下文解码器来说还是会有 1.5 到 2 个 BLEU 值的差距，这充分说明了引入双向的信息对于改善非自回归翻译译文的连贯程度，从而提升非自回归模型输出译文质量的作用。

其次，为了验证模型训练时结合自身的上下文信息能够自适应的找到一个合适的输出，本实验还对比了先见解码器的输入是参考译文时解码的译文质量。从实验结果来看，采用参考译文来训练模型只能对模型输出带来相当有限的提升，与直接用模型解码的输出的上下文信息比起来更是有 1.59 个 BLEU 值的差距，这也说明利用模型自身学到的先验知识，在结合一定的上下文信息来引导模型训练，是能够对模型的输出译文质量带来一定的提升的。

3.5.3.2　部分超参对训练结果的影响

接下来的实验主要是分析训练过程中的超参对结果的影响，在本实验中主要分析了控制 L_{seer} 和 L_{NAT} 权重的超参 β 对结果的影响。具体来说，本实验中控制了其他参数不变，采用了多个不同的超参数 β 的取值，β 越大，交叉熵的权重越大，反之，则先见解码器引导的损失越大。本实验在 WMT14 En→De 上进行。其最终得到的结果如下图所示：

首先，可以看到，对于先见解码器而言，随着超参数 β 的增加，模型的 BLEU 值呈现出先上升再下降的趋势。对 WMT14 En-De 而言，$\beta = 0.6$ 时模型输出的 BLEU 值达到了最高值，之后随着 $\beta = 0.6$ 的增加而迅速衰减。这说明缺少足够的上下文信息会使得模型的输出译文质量显著下降，这也证实了本方法对模型输出译文质量的提升。

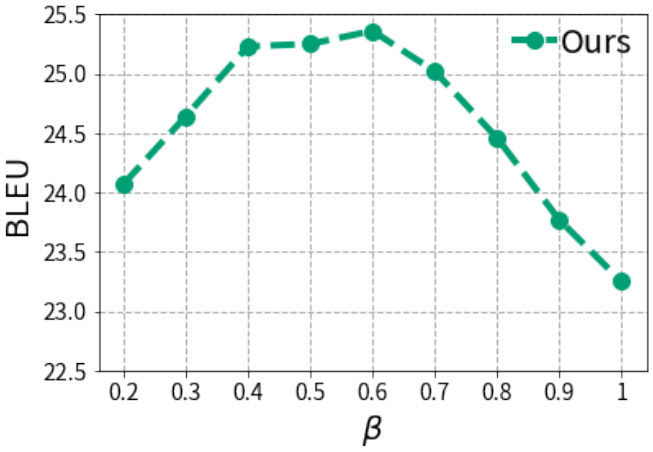


图 3-4 WMT14 En→De 上 BLEU 值随超参 β 的变化

Figure 3-4 The change of BLEU value with hyperparameter β on WMT14 En→De task

3.5.3.3 输出句子质量的变化

正如之前所提到的，在非自回归翻译中普遍存在着多峰性的问题，其结果是翻译的句子出现大量高频词的重复，导致输出的句子缺乏关键信息且不流利。而为了验证方法对输出译文质量带来了一定的提升，本工作分别针对模型输出译文的准确度和流利度进行了分析实验。就准确度而言，本实验中计算了输出译文与参考译文的词级别准确性；而就困惑度而言，本实验利用原始训练集的目标端数据和 KenLM 统计工具[57] 练了一个 5-gram 语言模型，通过计算输出译文的困惑度来计算句子的流利度。本实验将标准 NAT 模型与结合上下文的方法进行了对比，在 WMT14 En→De 进行了实验，结果如下表所示：

表 3-4 模型解码句子的词级别准确性和语言模型困惑度的对比

Table 3-4 Comparisons in Token-Level Accuracy and Perplexity of Decoded Sentences

	词级别准确率	困惑度
Vanilla NAT	47.87%	3.212
+ Seer	**53.50%**	**2.835**

从上表中可以看到，无论是从准确度还是流利度的角度，本方法都比之前的方法都有一定的改善，这说明本方法一方面通过在训练时提供上下文信息，使得模型的输出更加连贯，且更多的相关于参考译文的信息会出现在输出译文中。

3.5.3.4 样例分析

前面的分析实验中展示了本方法对翻译句子输出译文质量的提升，为了展示模型对翻译输出译文质量改进的效果，本工作中选取了一些相关的样例进行展示。通过对比了在 WMT14 De→En 测试集上通过上下文信息训练后的模型解码的例子和基线模型中的 3 个例子，并与目标端句子进行了对比，具体如下所

示：

表 3-5 上下文信息目标函数训练在 WMT14 De→En 上的结果示例

Table 3-5 Results of some cases in WMT14 De→En

源端句子	Snow@@ den könnte auch in Russland aus@@ sagen
参考译文	Snow@@ den may make further statements in Russia
标准 NAT 输出	Snow@@ den could also say Russia in Russia
本工作模型输出	Snow@@ den could also say in Russia
源端句子	Denn Matthias Sch@@ rel@@ l hat noch einiges vor am Standort .
参考译文	For Matthias Sch@@ rel@@ l has some further plans at the location .
标准 NAT 输出	Matthias Matthias rel@@ l still has a a lot to do at the site .
本工作模型输出	Matthias Sch@@ rel@@ l still has a lot to do at the site .
源端句子	Zu uns kommen Kunden aus jeder sozialen Schi@@ cht .
参考译文	We see customers from all walks of life .
标准 NAT 输出	Customers come to customers from every social class .
本工作模型输出	Customers come from every social class .

从上述示例中可以看出，首先，非自回归的翻译中确实存在着较为普遍重复的现象，比如第一句中的”Russia”，第二句中的”Matthias” 和”a” 以及第三句中的”Customers” 。而本方法输出的句子则消除了这种重复的内容，在使得句子表述的意思更准确的同时，能够获得更流畅的输出译文。但于此同时，也要注意到，对于模型中翻译得不够准确的地方，比如第一句中的”statement”，方法还有一定的改进空间。这也是后续可以研究的方向。

3.6　本章小结

本工作主要是提出了一种通过利用上下文信息辅助非自回归模型训练的方法。具体来说，针对非自回归模型的极大似然损失函数中缺乏对模型输出的上下文的结合的问题，本工作引入了一个建模在给出上下文的条件下模型在相应位置的输出分布的先见解码器模型，在训练时，本方法将模型的输出词作为先见解码器的输入来计算各个位置基于相应上下文信息的分布，并使得模型在降低极大似然损失的同时能够降低模型输出分布和先见解码器的输出分布之间的 Jensen 散度。本工作针对这一方法设计了一系列的训练策略，并在多个数据集上进行了实验，而实验的结果表明，本方法在多个数据集上都实现了超过 2 个点的显著的 BLEU 值提升，且在准确度和流利度上都能够实现一定的提升，这说明通过在训练目标中利用模型先验知识引入上下文信息能够显著缓解非自回归翻译中的多峰性问题，改善模型输出译文质量。

第 4 章　多样本保序回归的损失函数设计

4.1　引言

正如前文所述，虽然非自回归机器翻译实现了快速的并行解码，但由于非自回归翻译采用了输出独立的假设，使得模型拟合的分布偏离真实分布。上一个工作利用了一个可以引入上下文信息的解码器来导入模型上下文的信息，从而实现了模型输出译文质量的提升。但引入这种结构后的模型仍然存在着模型内生的问题，一方面是在独立输出假设下模型对真实分布的建模能力存在一定限制，而另一方面则是在解码时只能对固定长度的模型输出进行解码，导致在实际解码时需要对长度进行预测。

针对这一类问题，一些研究者提出了基于降采样的非自回归模型，在 Huang 等[33] 中，作者在非自回归模型的输出之后接入了一个建模有向图的网络结构，这种结构在确保解码器的输出独立的同时，通过一个轻量级的有向图模块来建模上下文的信息，从而使得模型拟合真实分布的能力大幅提升，从而改善了非自回归翻译的输出译文质量。

而正如之前的工作中所提到，作为一种非常常见的损失函数，极大似然损失函数具有实现简单，且在理论上是真实分布的无偏估计的特点。但这种对真实分布的拟合能力其实是存在着一定的理论限制的：具体来说，其要求模型训练时有足够多的来自真实分布的样本。在自回归模型中，由于模型每一步是在给出句子的一部分的情况下预测模型输出的下一个词，使得相应的真实分布的熵大为降低，训练所需要用来拟合的样本也有所减少；但是对非自回归模型而言，由于预测时完全没有上下文，因此输出分布的复杂度大为增加，相应的，模型在训练时所需要的数据量也显著提升。但这也就使得在采用极大似然估计对模型进行训练的时候，有限的训练集会带来较大的偏差，即使得模型的输出分布权重集中在参考译文上。而众所周知，由于一个意思往往同时会有多种不同的表达方式，因此同一句话常常会对应多种不同的翻译。因此，这种利用有限的训练集进行极大似然训练得到的模型会显著影响非自回归翻译的输出译文质量。

既然采用前述的极大似然损失函数会使得模型在拟合真实分布时存在偏差，那么就需要设计新的目标来加以拟合。由于文本空间的复杂性，直接将译文的分布显式地表示出来显得比较困难，但也可以大体的做出一些假设。比如，一个比较符合生活常识的假设是：句子质量越好，相应的概率越大。而本工作将会基于这样的假设来设计模型的训练目标，从而改进模型的输出译文质量。

本文中将会利用非自回归可以直接建模完整的输出分布的特性来设计方法。具体来说，相比起自回归模型学习的是给定源端和部分标准参考译文下的分布，非自回归模型的得到的是只依赖于源端的分布。因此可以直接从非自回归模型分布中进行采样，从而得到数个样本和他们的对应概率。而与此同时，根据前面所提到的“句子质量越好、概率越大”的假设，本工作中采用了一种保序回归损

失函数，即约束各个译文的输出概率与其质量相一致。保序回归函数曾在摘要等生成式任务中使用[58,59]，但这些研究都基于预训练过的自回归模型，且都依赖于提前得到的固定的多组参考输出。而本工作则是基于解决非自回归模型拟合真实分布时的困难，充分利用非自回归模型的特性进行实时采样。本工作在多个数据集上进行了相应的实验，发现结果都带来了一定的提升。

4.2 相关工作

作为一种常用于统计推断的统计学技术，保序回归的结构也在生成式的语言模型训练中得到过应用。Liu 等[58] 在摘要任务中引入过保序回归损失函数。作者先利用已经预训练好的模型，通过多样化解码[60] 的方式得到多组输出译文；之后在微调阶段在原有交叉熵损失函数的基础上，引入了一个保序的损失函数，确保前面通过多样化解码的输出译文的概率同质量呈正相关。作者通过上述方法，在生成式摘要任务上实现了性能的显著提升。后来 Zhao 等[59] 将这一技术与大模型相结合，用在了多种输出任务上，也取得了一定的提升。但受自回归模型特性等因素的影响，这些计算保序回归的时候都采用了固定的输出，这一定程度上限制了这一方法的性能。

而近年来，非自回归机器翻译模型技术取得了显著的发展，以 CTC-NAT[32] 和 DA-Transformer[33] 为代表的一类模型，通过输出降采样的方式来改善模型输出译文质量。其中，CTC-NAT 模型通过将去除并行输出结果中的掩码词 [M] 来得到最终的输出译文，而 DA-Transformer 则根据由输出隐状态计算得到的有向图概率转移矩阵来得到降采样后的输出。两种方法在保留非自回归模型并行解码特性的同时取得了显著的性能提升。这些模型的非自回归特性为利用更灵活的采样的方式计算保序回归损失函数创造了条件。而其本身采用的交叉熵损失函数在与真实分布之间也有着较大的方差，这为我们结合保序回归损失函数改进输出译文质量创造了条件。

4.3 背景

本节将介绍有关多样本保序回归的损失函数设计的一些背景知识。

4.3.1 保序回归

保序回归（Isotonic Regression）也可被称为单调回归（Monotonic Regression）。根据 Nuesch[61] 中的定义，给定集合 $D = (x_1, y_1), ..., (x_n, y_n)$，其中，$x \in X$（$X$ 是一个有序的集合），$y \in \mathbb{R}$。设有一个待学习的，从集合 X 映射到实数空间的函数 $f(x)$，则定义保序回归的目标函数为：

$$\begin{aligned} &\min_f \sum_i w_i (y_i - f(x_i))^2, \\ &\text{s.t.} f(x_i) \leq f(x_j), \forall x_i < x_j. \end{aligned} \tag{4-1}$$

在传统的统计推断中，相较于传统的、参数数目有限的线性回归方法，保序回归在单调的场景下具有能够更好的建模局部的特征的优势，因此得到了广泛的应用。而这一回归问题也可以利用 PAVA 法[62]，该方法将保序回归问题利用拉格朗日乘子法转换一个凸优化问题加以解决。而其面对的单调排序场景，为将其在机器翻译任务中应用创造了条件。

4.3.2　基于有向图的非自回归机器翻译

基于有向图的非自回归机器翻译模型又被称为 DA-Transformer[33]，该方法是一种通过结合有向图转移矩阵对非自回归模型输出进行降采样，从而得到输出各个位置输出的方法。

从模型结构上讲，DA-Transformer 沿用了一般的非自回归模型采用的编码器—解码器框架，其改进主要在解码器的结构上，如图4-1所示。

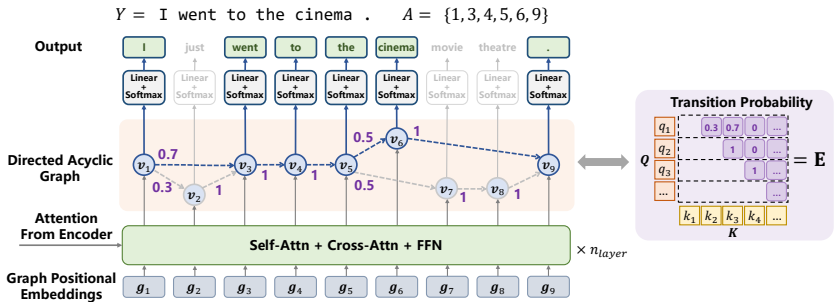


图 4-1 DA-Transformer 的解码器结构

Figure 4-1 Decoder structure in DA-Transformer

具体来说，模型的解码器输入一个源端长度数倍的位置嵌入向量，输入到多层的解码器后得到了各个位置的输出隐状态，隐状态的功能主要有两个，一个则是用于计算上三角的状态转移矩阵，以用于模型根据矩阵选择相应的隐状态并输出各个词，即：

$$P\left(A|\mathbf{x};\theta\right)=\prod_{i=1}^{M-1}P(a_{i+1}|a_i,\mathbf{x};\theta)=\prod_{i=1}^{M-1}E_{a_i,a_{i+1}}. \tag{4-2}$$

其中有向图矩阵的生成可以用如下过程来表示：

$$E_i=\mathrm{softmax}(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_h}}), \tag{4-3}$$

$$E=\mathrm{Concat}(E_1,...,E_H)AW^O. \tag{4-4}$$

而隐状态的另一个功能是用于对相应位置的输出词的预测，即：

$$P(\mathbf{y}|A,\mathbf{x};\theta)=\prod_{i=1}^{M}P(\mathbf{y}_i|a_i,\mathbf{x})=\prod_{i=1}^{M}\mathrm{softmax}(\mathbf{W}_{\mathrm{P}}v_{a_i}). \tag{4-5}$$

最终模型根据各个位置输出词的概率和转移路径的概率得到最终输出句子的概率分布，即：

$$P(\mathbf{y}|\mathbf{x}) = \prod_{A} P(\mathbf{y}|A,\mathbf{x})P(A|\mathbf{x}). \tag{4-6}$$

根据前述的建模方式，DA-Transformer 可以在训练时通过动态规划的方式计算出所有能得到参考译文的路径的概率，并计算相应的交叉熵损失函数。而在解码阶段则可以通过贪心解码、先行解码、维特比解码[35] 等轻量级的解码策略来实现降采样的过程。同时，其为了令模型能够有更好的输出译文质量，其采用了 Qian 等[21] 中的瞥见策略来改进模型的训练，实验的结果表明，在维持解码速度提升的情况下，这一方法在不采用知识蒸馏的原始数据上超过了所有的基线方法，其与同等参数规模的自回归模型在输出译文质量上的差距也降到了 0.5 个 BLEU 值以内。

4.4 非自回归损失函数的保序回归损失

在本节中将会具体的对本方法中训练非自回归模型所采用的方法进行介绍。首先是介绍本方法的基本概念及其用于改进非自回归翻译的理论基础；接下来将对所采用几种的保序回归损失函数进行介绍；之后将会展示本方法的训练流程。

4.4.1 极大似然损失学习真实分布的局限

在前面的工作中已经看到，直接采用词级别独立的极大似然估计会使得模型训练的分布偏离真实分布，带来显著的多峰性问题。而根据 Huang 等[63] 的论证，在假设各个词概率相互独立的情况下，模型拟合真实分布的能力存在着一定的上限，而以 DA-Transformer[33] 为代表的一类模型通过将模型并行得到的输出进行相关性降采样的方式，大幅提升了非自回归模型拟合真实分布的能力，从而提升了非自回归模型的翻译质量。而这种性能的提升也使得利用非自回归模型更好的建模真实分布成为了可能。

长久以来，为了建模真实空间的分布，在自然语言处理的各种任务当中，极大似然损失函数都是最为常见的一种损失函数，因为在样本量足够的情况下，极大似然估计训练得到的分布就是对真实分布的一个无偏估计。但对于句子到句子的翻译而言，由于一个句子往往普遍的存在着数个可行的翻译，而训练集中往往无法覆盖所有的翻译，这就会使得模型倾向于训练得到一个在参考译文上的独热分布，而这种模型分布显然不符合实际情况。相应的就需要拟合一个更实际的分布以提升模型的泛化性能。虽然由于训练所用的知识有限，导致并没有办法确定实际的分布[1]，但仍然可以从日常生活中得出一些显而易见的结论，如“译

[1]数学上可以证明，一个在固定维数的实数空间的随机变量若要满足熵最大，其分布必然是一个高斯分布。虽然尚无法证明可以将一个句子的所有译文建模在一个有限维的实数空间之上，但这一结论仍然能为

文质量越高，译文的概率应当越大”。而这些结论为利用保序回归这一数学工具训练模型创造了条件。

4.4.2　保序回归损失函数

接下来介绍本工作中采用的保序回归损失函数，其表达式如下式所示：

$$L_{score} = \sum_{C}(\max(0, \beta_{ij} - (\log p(\mathbf{y}^i) - \log p(\mathbf{y}^j)))). \tag{4-7}$$

其中，对于任意的 i, j ，只要 $i > j$ ，都有 $\text{score}(\mathbf{y}^i) > \text{score}(\mathbf{y}^j)$ 。

在上式中，score 代表的是采样的句子的质量，其可以通过各种方式（包括 BLEU[44] ，chrF[64] 等指标、模型学习或人工评估）来得到，β_{ij} 是一个约束各个句子对之间概率距离的项，而保序集合 $C = \{(\mathbf{y}^i, \mathbf{y}^j)\}$ 则为参与保序回归损失函数计算的各个损失函数，具体的选取将会在后面加以介绍。

从上式中可以看到，保序回归方法其实是对输出的概率引入了一种较为宽松的约束，具体来说就是对于质量高于 $\mathbf{y}^j$ 的 $\mathbf{y}^i$ （即满足 $\text{score}(\mathbf{y}^i) > \text{score}(\mathbf{y}^j)$ ），如果模型分别输出的概率 $P(\mathbf{y}^j) > P(\mathbf{y}^i)$ ，则对质量较差的句子的概率予以惩罚，而对质量较好的句子予以鼓励。相应的，也可以将这一损失函数写作如下的形式：

$$\begin{aligned} L_{score} &= -\sum_{i=1}^{n} R_i \log p(y^i), \\ R_i &= \sum_{\substack{\forall (i,k) \in C \\ score(\mathbf{y}^i) > score(\mathbf{y}^k)}} \mathbb{1}(\log p(\mathbf{y}^i) < \log p(\mathbf{y}^k) + \beta_{ij}) \\ &\quad - \sum_{\substack{\forall (k,i) \in C \\ score(\mathbf{y}^k) > score(\mathbf{y}^i)}} \mathbb{1}(\log p(\mathbf{y}^i) > \log p(\mathbf{y}^k) - \beta_{ij}). \end{aligned} \tag{4-8}$$

从式 (4-8) 中可以看出，保序回归损失函数可以写为某种与增强学习的策略梯度算法相似的一种形式。但在这里需要说明的是，根据策略梯度的数学证明，策略梯度算法中需要确保各个样本 $\mathbf{y}^i$ 是状态空间中依概率采样得到的，才能使得式 (4-8) 能够表示对期望惩罚的无偏估计，而保序回归中构建保序样本的方式相较而言就要丰富很多，并不一定是直接采样得到。

以上介绍的是保序回归损失函数的实现将会结合非自回归模型的特性，相应的设计保序保序集合 C 和相应的损失函数。

4.4.3　保序回归损失函数在非自回归模型的应用

在之前的一些工作（如 Zhao 等[59], Vijayakumar 等[60]）中，作者利用自回归模型的特性，利用集束搜索或者多样集束搜索等方法来得到参与保序回归计

本工作给出一些启示。

算的句子，并通过两两之间的比较来计算保序回归损失函数。由于自回归模型的串行解码特性，所以采用集束搜索解码得到保序集合的方式不能直接进行，而是只能从预训练的模型中得到固定的解码样本。这样使得模型在训练过程中存在着一定的偏差。而另一方面，由于自回归模型预测各个位置的输出分布时需要之前的各个位置的输出，因此对每个样本都要重复输入到解码器中，这会带来显著的训练时间和显存开销增加。相较而言，非自回归模型具有解码过程并行进行、可实时计算所有位置的分布、训练过程与解码过程一致等特点，这就使得采用一些实时的方式获取参与保序回归训练的样本成为了可能。但与此同时，非自回归模型的多峰性问题也使得本工作需要考虑如何更好的构建保序的集合，从而使模型尽量的学到一个单峰的分布。下面将对本工作的方法改进进行介绍。

4.4.3.1 参与保序运算的句子的采样

如前文所述，非自回归翻译模型的一些特性为实现在线采样创造了一定的条件。因此，本工作采用了直接从输出分布中采样的方式来构建保序回归的样本。具体来说，通过分别对 DA-Transformer 中各个词的发射概率 $P(y_i|\mathbf{x};\theta)$ 和转移概率 $P(a_{i+1}|a_i,\mathbf{x};\theta)$ 进行采样，根据采样的结果从第一个词进行解码得到最终的译文。设上述位置中发射得到的词为 $\mathbf{y}=(y_1,...,y_T)$，其对应的转移到的位置为 $\mathbf{a}=(a_1,...a_{T-1})$，相应的其概率表达式可以记为：

$$\begin{aligned} p(\mathbf{y},\mathbf{a}|x;\theta) &= p(y_1,...,a_i,y_{i+1},a_{i+1},...,y_T|x;\theta) \\ &= p(y_1,...,a_i,y_{i+1},a_{i+1},...,y_T|x,a_0;\theta) \\ &= \prod_{i=1}^{T} p(y_i,a_i|a_{<i},y_{<i},x;\theta) \\ &= \prod_{i=1}^{T} p(y_i|a_{i-1},x;\theta)\cdot p(a_i|a_{i-1},x;\theta). \end{aligned} \tag{4-9}$$

从上式中可以看出，前述的采样的方式实际上是得到了一个对 $p(\mathbf{y},\mathbf{a}|\mathbf{x};\theta)$ 的无偏采样，而由于在任务中只需要使用采集到的样本中的目标句子 $\mathbf{y}$，所以这里也可以视作是对 $p(\mathbf{y}|\mathbf{x};\theta)$ 采样得到的样本。在训练过程中，上述采样过程是并行进行的，而根据转移矩阵获取采样结果的过程则是串行进行，因此这样的方式显著的减少了计算所需的时间，为实现实时的采样创造了条件。在本工作中，通过实时进行上述采样过程，从而实现了对模型的更充分的训练。

4.4.3.2 非自回归模型的保序集合的构建

正如本工作中所提到的，一般意义上的保序回归损失函数是通过将所有采集到的样本通过两两比较的方式来计算相应的损失函数，此时的保序集合为：

$$C_{multi}=\{(\mathbf{y}^i,\mathbf{y}^j)|1<=i<j<=n\}. \tag{4-10}$$

此时，对于样本 $\mathbf{y}^i$ ，保序集合中与其相关的子集包括了 $\{(\mathbf{y}^1,\mathbf{y}^i),...,(\mathbf{y}^{i-1},\mathbf{y}^i),(\mathbf{y}^{i+1},\mathbf{y}^i),...,(\mathbf{y}^n,\mathbf{y}^i)\}$ 。而从式 (4-8) 中，可以看到，在训练过程中，可能存在着同时有多个 $\mathbf{y}^i$ ，满足对应的 $R_i>0$ 。而对于非自回归翻译模型而言，由于其本身表示多峰分布较为困难，因此学习这样的损失会对模型的输出译文质量带来一定的影响。针对这样的问题，本工作中引入了一个单峰的保序集 C_{one} ，即

$$C_{one}=\{(\mathbf{y}^{max},\mathbf{y}^i)|\forall i\neq max,\text{score}_{max}>=\text{score}_i\}. \tag{4-11}$$

上式中，$\mathbf{y}^{max}$ 为评价指标最高的保序样本。而当采用上述保序集合去计算保序回归损失时，式 (4-8) 中的 R_i 可以简写为：

$$R_i=\begin{cases}\sum_{k\neq i}\mathbb{1}(\log p(\mathbf{y}^i)<\log p(\mathbf{y}^k)+\beta_{ij}) & i=max\\ -\mathbb{1}(\log p(\mathbf{y}^{max})<\log p(\mathbf{y}^i)+\beta_{ij}) & i\neq max\end{cases}. \tag{4-12}$$

从上式中可以看出，采样上述单峰保序集合时，模型只有一个样本（即评价指标最高的样本）的 R_i 是大于零的，这样就能够引导模型去学习一个单峰的训练目标，当然，这种训练目标中由于学习的保序关系更少，其建模一个准确分布的能力也会相对弱化一些。因此，在本工作的实现中采用了课程学习的方法，即随着训练步数的增加，模型的损失逐步从多峰的保序集过渡到单峰的保序集，如果设多峰保序和单峰保序的损失分别为 L_{multi} 和 L_{one} ，那么实际训练时的保序回归损失函数 L_{rank} 可以表示为：

$$L_{rank}=\lambda L_{multi}+(1-\lambda)L_{one}. \tag{4-13}$$

$$\lambda=\lambda_{max}-\frac{\max(\min(x,\text{step}_{\max}),\text{step}_{\min})-\text{step}_{\min}}{\text{step}_{\min}-\text{step}_{\min}}\cdot(\lambda_{max}-\lambda_{min}). \tag{4-14}$$

4.4.4　训练流程

以上介绍了多样本保序回归方法的背景，以及其结合和针对非自回归翻译场景的一系列改进。下面将会介绍训练的流程，在本工作中，训练集句子 $\mathbf{x}$ 先通过模型 θ 得到输出分布 $p(\mathbf{y}|\mathbf{x};\theta)$ ，之后再从 $p(y|x;\theta)$ 中采样得到多组输出 $\mathbf{y}^1,\mathbf{y}^2,\ldots,\mathbf{y}^n$ ，接下来，再相应的利用各种评价指标计算各个译文的指标分数 $\text{Score}(\mathbf{y}^i,\mathbf{y})$ 并排序，最后相应的计算损失函数并更新模型 θ 。

而在上述的训练过程中，也有一些需要展开介绍的地方。首先，本工作中针对模型损失函数的正向计算和反传梯度的过程进行了改进。在 Huang 等[33] 中，作者以发射概率 P 、转移概率 E 和参考译文 $\mathbf{y}$ 作为输入构建了计算 DAG 损失函数的模块 $\text{DAG}(P,E,\mathbf{y})$ ，其前向和后向过程如下所示：

- 前向：根据 P 、E 和 $\mathbf{y}$ 来计算前向和后向的状态转移概率矩阵 $M_\alpha(\mathbf{y})$ 和 $M_\beta(\mathbf{y})$ ，相应的计算出损失函数 $\mathrm{DAG}(P,E,\mathbf{y})$ 并将 $M_\alpha(\mathbf{y})$ 和 $M_\beta(\mathbf{y})$ 存储在计算图中。
- 后向：根据反传的梯度 $\frac{\partial L}{\partial \mathrm{DAG}(P,E,\mathbf{y})}$ 计算出损失函数的梯度，即 $\frac{\partial L}{\partial P}$ 和 $\frac{\partial L}{\partial E}$ 。具体如下式所示：

$$\begin{aligned}\frac{\mathrm{DAG}(P,E,\mathbf{y})}{\partial P} &= f_P(P,E,M_\alpha(\mathbf{y}),M_\beta(\mathbf{y})),\\ \frac{\mathrm{DAG}(P,E,\mathbf{y})}{\partial E} &= f_E(P,E,M_(\mathbf{y}),M_\beta(\mathbf{y})),\end{aligned} \tag{4-15}$$

$$\begin{aligned}\frac{\partial L}{\partial P} &= \frac{\partial L}{\partial \mathrm{DAG}(P,E,\mathbf{y})}\cdot\frac{\partial \mathrm{DAG}(P,E,\mathbf{y})}{\partial P},\\ \frac{\partial L}{\partial E} &= \frac{\partial L}{\partial \mathrm{DAG}(P,E,\mathbf{y})}\cdot\frac{\partial \mathrm{DAG}(P,E,\mathbf{y})}{\partial E}.\end{aligned} \tag{4-16}$$

而在本工作中，如果根据公式（4-8）并结合函数 $\mathrm{DAG}(P,E,Y)$ 来完成损失函数的计算，则会导致对每一个采样到的译文都需要在计算图中存储一个中间变量 M_α 和 M_β ，这样会带来较大且不必要的显存开销。因此本工作重新结合了链式法则设计了模块 $\mathrm{MULTI_DAG}(Y,R,P,E)$ ，其将所有采集到的参考译文集合 Y ，参考译文对应的分值 R 、发射概率 P 、转移概率 E 作为输入，前向和后向的过程如下所示：

- 前向：对参考译文集合 Y 中的每一句 $\mathbf{y}$，根据 P、E 和 $\mathbf{y}$ 来计算前向和后向的状态转移概率矩阵 $M_\alpha(\mathbf{y})$ 和 $M_\beta(\mathbf{y})$，进而直接计算出 $f_P(P,E,M_\alpha(\mathbf{y}),M_(\mathbf{y}))$ 和 $f_E(P,E,M_\alpha(\mathbf{y}),M_\beta(\mathbf{y}))$。接下来利用分值 R，结合（4-8）计算出损失函数 $\mathrm{MULTI_DAG}(Y,R,P,E)$ 和梯度 $\frac{\partial \mathrm{MULTI_DAG}(Y,R,P,E)}{\partial P}$ 和 $\frac{\partial \mathrm{MULTI_DAG}(Y,R,P,E)}{\partial E}$，并将计算得到的梯度缓存于计算图。
- 后向：根据反传的梯度 $\frac{\partial L}{\partial \mathrm{MULTI_DAG}(Y,R,P,E)}$，并利用之前存储的梯度并结合链式法则计算出 $\frac{\partial L}{\partial P}$ 和 $\frac{\partial L}{\partial E}$。

通过 $\mathrm{MULTI_DAG}(Y,R,P,E)$，本工作在计算损失函数的过程中计算图只存储了 2 个变量，从而显著的减小了模型训练时所需的显存开销，加快了训练的速度。

4.5 实验结果与分析

本部分主要是对所进行的实验进行介绍，并结合实验结果进行一定的分析，以验证保序回归损失作为能够作为一种更优的损失函数。

4.5.1 实验设置

本节主要介绍本实验中采用的实验设置，包括训练数据、评价指标以及其他的实现细节。

4.5.1.1　训练数据

就训练而言，分别采用了 WMT14 英语到德语翻译任务（WMT14 En→De ）和 IWSLT14 德语到英语翻译任务（IWSLT14 De→En ）。对 WMT14 En→De 任务，为了能够与 Huang 等[33] 相对照，本工作对数据集进行了清洗，而这使得实际上本工作的训练集规模会略小于 WMT14 En→De 任务原本的训练集，两个任务的数据集具体的数据集规模如下表所示。而对于上述几个任务，本工作采用了 32000 步的 BPE 模型[51] 来对语料进行预处理，同时也共享了源端和目标端的词表。需要说明的是，由于 DA-Transformer 模型本身性能上的大幅提升，因此在本实验中直接采用了原始的数据集，而没有用自回归模型对训练集集进行知识蒸馏。

表 4-1 保序回归损失函数训练实验中数据集的信息

Table 4-1 Dataset statistics in monotonic regression loss training experiments

数据集	训练集	验证集	测试集
WMT14 En-De	3961179	3000	3003
IWSLT14 De-En	160239	6750	6750

4.5.1.2　评价指标

在本工作中，评价指标的用途主要有两个，一个是在训练时，通过计算采集到的样本的质量来计算各个样本概率的排序；而另一个则是在测试时，用于对测试集翻译的准确性来评估翻译的性能。结合了计算速度和翻译质量、本工作采用了如下 2 种基于统计的评价指标：

- BLEU[44]：这一指标是一种基于 n-gram 的评价指标。具体来说，其根据各个词级别 n-gram 在论文中的准确度来进行统计。在 BLEU 这一指标中，计算各个 n-gram 的准确度如下式所示：

$$P_{n-\text{gram}} = \frac{\sum(\min(\text{count}_{\text{output}}, \text{count}_{\text{reference}}))}{\sum \text{count}_{\text{output}}}. \tag{4-17}$$

在计算得到词级别 n-gram 的基础上，可以表示出如下式的 BLEU 计算公式：

$$\text{BLEU} = \text{BP} \cdot \exp(\sum_{i=1}^{N} w_n \cdot \log P_n). \tag{4-18}$$

需要注意的是，上式中 $\text{BP} = \min(\exp(1 - \frac{r}{c}), 1)$，其中 r 和 c 分别是参考译文和候选译文的长度，其主要是为了惩罚较短的句子。

- chrF[64]：顾名思义，这一指标计算的是字符级的 F-score，具体来说，其先计算了字符级别 n-gram 的准确率 chrP_n 和召回率 chrR_n（即候选译文中的 n-gram 在参考译文中出现的比重和参考译文的 n-gram 在候选译文中的比重）。之后再通过下式来计算字符级的 F-score：

$$\mathrm{chrF}_n = (1+\beta^2)\frac{\mathrm{chrP}_n \cdot \mathrm{chrR}_n}{\beta^2 \cdot \mathrm{chrP}_n + \mathrm{chrR}_n}. \tag{4-19}$$

再得到从 1 到 m 的 F-score 的基础上，再通过对上述 F-score 进行平均，从而可以得到最终得到的分值。

在本实验中，对于 BLEU 指标，本工作采用的超参数是 $N = 4$；而对于 chrF，本工作采用的超参数是 $M = 6$，$\beta = 3.0$。在排序阶段，本工作对上述指标利用 pybind11[65] 封装的多线程 C++ 模型来进行并行计算。而在测试时，本工作则通过 multi-bleu.perl 对输出译文的 BLEU 值进行计算。

4.5.1.3 其他的实现细节

在模型参数方面，对 WMT14 En→De 任务，本工作采用了类似于 Transformer-Base 模型的结构，即词表的维数为 512 维，中间层隐状态维数为 2048 维，注意力机制头数为 8，而对于 IWSLT14 De→En 任务，本工作采用的配置就会稍小一些，即词表维数为 256 维，中间层隐状态维数为 1024 维，注意力机制头数为 4。而就 dropout 概率而言，本工作分别对 WMT14 En→De 任务和 IWSLT14 De→En 任务分别取值 0.1 和 0.3。在本实验的模型中，源端和目标端共享词嵌入模块，且两个解码器的解码器输出投影层与目标嵌入层共享相同的参数。

就学习策略而言，对训练集较小的 IWSLT14 De→En 任务，本工作采用的是从头训练的策略，其每次处理的批的词数为 32000 个词；而数据规模更大的 WMT14 En→De 任务来说，本工作采用的是基于现有模式微调的策略，其每次采用的批的次数为 64000 词。就优化器和学习率设置来说，虽然两者均采用了 Adam 优化器[54] 且超参数均为 $(\beta_1 = 0.9, \beta_2 = 0.999)$，而且都采用了平方根逆衰减的学习率调整策略，但在具体实施上却有很多不同：对 IWSLT14 De→En 任务而言，其学习率为 0.0007，且会进行 10000 步的预热步骤；而对于 WMT14 En→De 任务，由于其学习率仅则为 0.0002，且只会进行 1000 步的预热操作。两者在训练时均采用了瞥见训练的模式，即根据错误的词的数量在解码器的输入端提供一定的信息，但对 IWSLT14 De→En 任务，本工作采用了从 0.5 到 0.1 衰减的瞥见系数，而对 WMT14 En→De 任务，本工作则只采用了 0.1 的瞥见系数。

而就其他超参数设置而言，如无特殊说明，就 DA-Transformer 的参数而言，本工作在两个任务上均设置上采样倍数为 8，且采用前向展望解码方式；而就保序相关超参而言，本工作对 IWSLT14 De→En 任务和 WMT14 En→De 任务设置超参 $\beta = 0.002$ 和 0.005，采样的个数为 16 个且均采用 chrF 评价指标进行排序。最后的实验结果均是选择验证集上最优的 5 个检查点进行平均得到。

以上的模型和训练过程的代码实验都是在基于 Pytorch 框架[55] 下的 fairseq[56] 工具包上完成。

4.5.2　主要实验结果

本节将会介绍多样本保序回归方法的主要实验结果，包括与 DA-Transformer 模型在几种解码方式上的对比，以及与其他模型的结果的对比。

4.5.2.1　与 DA-Transformer 的对比

由于本工作的方法是基于 DA-Transformer 这一框架，因此本部分的主要工作是将实验结果与原始的模型进行对比。本工作采用原始的数据集在前述的数据集上进行了实验，并采用了四种解码方式（即贪心解码（Greedy decoding）、前向展望解码（Lookahead decoding）、维特比解码（Viterbi decoding）[35] 和集束搜索解码（Beam search decoding）。本工作在实施的过程中均沿用了原作者的实现方式，最终的结果如下表所示：

表 4-2 DA-Transformer 引入保序回归损失函数后的性能对比

Table 4-2 Comparison with DA-Transformer in different decoding methods after utilizing monotonic loss

模型	解码方式	WMT14 En→De	IWSLT14 De→En	加速比
DA-Transformer	Greedy	26.08	33.24	×14.0
	Lookahead	26.57	33.56	×13.9
	Viterbi	26.89	33.65	×13.2
	Beam Search	27.02	33.90	×7.1
+ 保序回归损失	Greedy	26.34(+0.26)	33.59(+0.35)	×14.0
	Lookahead	26.77(+0.20)	33.89(+0.33)	×13.9
	Viterbi	27.02(+0.13)	33.95(+0.30)	×13.2
	Beam Search	27.39(+0.37)	34.24(+0.34)	×7.1

从上表的结果可以看出，相比起 DA-Transformer 来说，本工作的方法在各种解码方式上均能够取得 0.1 到 0.4 左右的 BLEU 值得提升。这一方面证明了采用保序回归损失函数能够提升模型输出译文质量；另一方面，现有的方法在模型输出译文质量的提升上还比较有限，说明本工作的方法还有着很大的改进空间。

4.5.2.2　与其他基线方法的对比

除了与 DA-Transformer 进行对比，本工作还利用原始数据训练的结果与其他几种方法进行了对比，由于这里需要与其他模型采用原始数据集进行对比，具体包括了以下几类的方法：

- 基于 Transformer[5] 的自回归的翻译模型。
- 经典的非自回归翻译模型，如标准 NAT 模型[2] 和 10 轮迭代式解码的 CMLM 模型[18]（即 CMLM_{10}）。
- 一些与经典的非自回归模型结构相近的改进模型，如 SynST 模型[66]、

GLAT 模型[21] 及其衍生的 Latent-GLAT 模型[27]、OAXE 模型[52]、以及 CNAT 模型[26]。

- 基于降采样机制的模型，如 DA-Transformer[33]。

下表中列举了主要的实验结果：

表 4-3 引入保序回归损失函数的 DA-Transformer 与其他基线方法的对比

Table 4-3 Comparison between DA-Transformer with monotonic loss and other baselines

模型	WMT14 En→De	IWSLT14 De→En	加速比
Transformer[5]	**27.4**	**34.4**	×1.0
Vanilla NAT[2]	11.79	17.77	**×15.3**
CMLM_{10}[18]	25.21	31.19	×3.8
SynST[66]	18.05	23.82	×4.9
GLAT[21]	19.42	29.07	**×15.3**
OAXE[52]	22.4	\	×14.2
CNAT[26]	21.3	29.81	×10.4
Latent-GLAT[27]	24.71	32.31	×11.3
DA-Transformer[33]	26.57	33.56	×13.9
+ 保序回归损失	**26.77**	**33.89**	×13.9

从下表中可以看出。首先，在原始的数据上，虽然基础的非自回归方法能够取得一系列的速度优势，但其在输出译文质量上是远不如自回归的模型；而即使是类似 CMLM_{10} 这样的通过多轮迭代的方式牺牲了一定输出译文质量的模型，其也与自回归的模型存在着 2 个点以上的 BLEU 值的差距。这也展示出在缺乏合理的上下文信息建模的情况下，利用知识蒸馏降低句子复杂度对非自回归模型获得较高质量的输出，实现模型输出译文质量提升的重要性。

虽然知识蒸馏本身具有改善非自回归翻译模型性能的作用，但由于此时自回归的模型训练的不是真实的参考译文，而是自回归模型输出的译文，因此在后续的一系列非自回归翻译的工作中，都研究了能否在不使用知识蒸馏的情况下接近自回归的输出译文质量。但正如表中所展示的那样，这些基于 Vanilla NAT 的改进虽然能够大体上保持模型速度提升的优势，但在最终的输出译文质量上仍然与自回归模型有着 2 个点的 BLEU 值的差距。直到 DA-Transformer 横空出世，才使得非自回归模型在翻译输出译文质量上能够达到与自回归模型相当的水平，此时模型与非自回归模型之间的输出译文质量差距缩小到了 1 个点以内。本工作的方法在这一方法的基础上又进一步的拉近了模型输出与自回归翻译模型的输出译文质量，这一方面说明了 DA-Transformer 的机制上仍然有很大的改进的前景，另外一方面也说明了利用自回归的生成模型达到非自回归模型的性能是可行的。

4.5.3　分析实验

本节将会对本工作中所进行的实验及其结果进行分析。

4.5.3.1　采样数量对结果的影响

首先，本工作分析了训练过程中的采样数量对结果的影响，实验在在 IWSLT14 De→En 任务上进行，固定其他的参数和过程不变，仅调整了训练过程中采样的数量，最终的结果如下图所示：

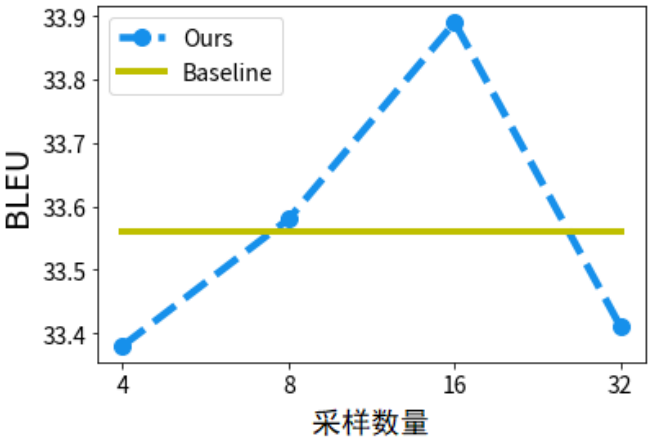


图 4-2 保序回归损失函数训练时随采样数量变化 BLEU 值的变化

Figure 4-2 The change of BLEU value with the number of samples during training with monotonic loss

从上图的情况来看，在采样数目比较小的时候，引入保序回归损失函数会导致解码的 BLEU 值比 DA-Transformer 略低，这是由于采样的样本数量较少，导致模型在采用保序回归损失训练时的方差较大，从而带来了一定的输出译文质量损失，而随着参数增加，当样本数从 4 增加到 16 时，模型的输出译文质量得到了一定的提升，这也说明随着样本数目的增加，模型训练的方差有所降低，这样也就带来了翻译输出译文质量的提升。而相比起更大的超参，当采样数量为 16 时模型能够取得最优的水平，这也说明本工作选取的训练超参是合适的。

4.5.3.2　采样指标对结果的影响

接下来，本工作比较了采样时采用不同的评价指标对结果的影响。如前文所述，这里采用了两种指标，即 BLEU 和 chrF，在其他参数不变的情况下进行了对比，结果如下表所示：

表 4-4 保序回归损失函数训练时随排序指标变化 BLEU 值的变化

Table 4-4 The change of BLEU value with the rank metrics in sampling process during training with monotonic loss

模型	排序指标	IWSLT14 De→En
DA-Transformer	\	33.56
+ 保序回归损失	BLEU	33.66
	chrF	**33.89**

从下表中可以看出：首先，可以看到模型在采用两种指标的情况下模型的输出译文质量都产生了一定的提升。其次，采用字符级别的 chrF 采样指标的训练结果要高于采用 BLEU 时的训练结果，这一方面也验证了 Xiang 等[67] 等的假设，即在英语和德语之间的翻译任务上 chrF 比 BLEU 与人工评测更加一致，另一方面也说明了字符级指标相对于词级别的指标更易于学习的特点。

4.5.3.3 混合训练对结果的影响

接下来，本工作将会展示采取混合训练时相比起仅仅采用单峰训练和多峰训练带来的输出译文质量的提升。参照公式（4-14），采用了如下的几种训练设置进行对比：

(1) $\text{step}_{min}=0$，$\text{step}_{\max}=10000$，$\lambda_{min}=0.2$，$\lambda_{max}=1.0$。

(2) $\text{step}_{min}=0$，$\text{step}_{\max}=10000$，$\lambda_{min}=0.1$，$\lambda_{max}=1.0$。

(3) $\text{step}_{min}=0$，$\text{step}_{\max}=10000$，$\lambda_{min}=0.0$，$\lambda_{max}=1.0$。

针对这一实验配置，本工作进行了一系列的实验，结果如下所示：

表 4-5 保序回归损失函数训练时随单峰和多峰权重变化 BLEU 值的变化

Table 4-5 Changes in BLEU values with unimodal and multimodal loss weights during training with monotonic loss

	DA-Transformer	λ=0	λ=1	随时间变化的 λ		
				(1)	(2)	(3)
IWSLT14 De→En	33.56	33.71	33.78	33.63	33.68	**33.89**

从上述结果中可以看出，首先，直接采取多峰的保序回归损失函数会带来模型输出译文质量的下降，而采取单峰的形式会带来一定的输出译文质量提升。而接下来则采用了带有时间策略的进行实验，结果发现以下两点，首先，随着训练结束时单峰比例的增加，模型的输出译文质量得到了提升；其次，在前期采用一定比例的多峰分布比起只采用单峰分布能够达到更高的输出译文质量。

根据上述的分析过程可以得到如下的一些结论：首先，直接学习多峰的损失会令模型难以学习，从而使得模型的输出译文质量降低；其次，虽然学习更容易的单峰损失能够带来输出译文质量的提升，但在前期若采用知识更丰富的多峰损失训练能够带来输出译文质量的提升。

4.5.3.4　在更尖锐分布上采样对结果的影响

Shen 等[68] 是利用评价指标训练翻译模型的一个经典工作，其中为了确保模型输出的句子最终能够拟合到参考译文上，作者在采样时强制采集了参考译文，从而使得模型在最终收敛时能个得到一个接近真实译文的输出。相应的，为了让模型采样到的译文能够更容易采集到参考译文，本工作采用了如下的策略：

(1) 强制模型在输出时采样参考译文。

(2) 约束模型每次采样时仅采样概率前 K 大的发射词和路径，在本实验中 $K = 512$。

本工作通过控制其他参数不变，将上述两种方法与本工作采用的方法在 IWSLT14 De→En 任务上进行了对比，结果如下：

表 4-6 保序回归损失函数训练时采用不同采样策略的结果

Table 4-6 Changes in BLEU values with difference sampling strategy during training with monotonic loss on the IWSLT14 De→En task

	DA-Transformer	+ 保序回归损失	采样策略	
			(1)	(2)
BLEU	33.56	**33.89**	33.63	33.46

从上表中的结果可以看出，采用上述的两种方式都试图从一个参考译文权重更大的分布中采样，结果在输出译文质量上都出现了 0.2 到 0.4 个 BLEU 值的差距。这说明对于本方法而言，相较于从一个更接近参考译文的分布中采样，从真实分布中采样得到的结果有着更高的输出译文质量。

4.5.3.5　多样性解码实验

正如前面所提到的，通过引入保序回归损失函数的方式能够使得模型得到的分布更为准确。而这种准确的分布并不仅仅反映在最优的解码方式得到的译文的准确度上，而且还能够反应在模型生成更为多样的译文的能力上。本实验参考了 Huang 等[33] 中的做法，通过不同温度的 top-p 采样（即核采样[69]）来在控制解码得到的译文的多样性。而在对多样性和准确性的评估上，则是参考了 Shen 等[70] 的做法，采用不同的随机数种子（从 1 到 5）采集了 5 组译文，将其与参考译文比较 BLEU 的同时计算各个译文两两之间的 Pairwise-BLEU。本工作在 IWSLT14 De→En 任务上进行了实验，结果如下图所示：

从上图中可以看出，首先，本方法在控制解码译文的多样性相同的情况下，本方法（绿色线条）的准确性要高于标准的 DA-Transformer，且两者之间的 BLEU 值相差达到了近一个点，这进一步突出的表明了通过本方法进行多样化的采样对模型输出译文质量带来的提升。

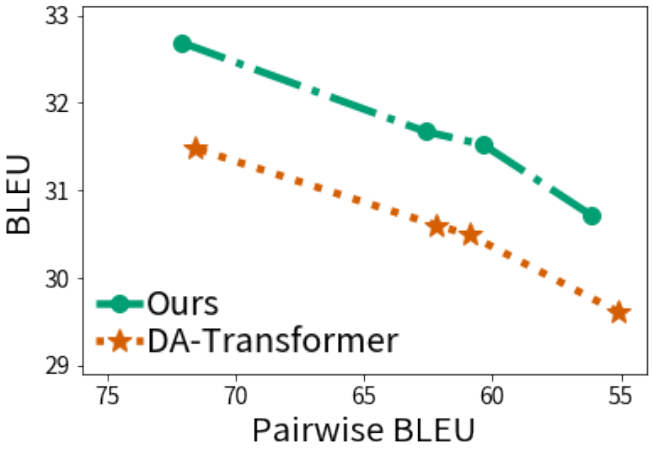


图 4-3 保序回归损失函数训练的模型在多样性解码任务上的结果

Figure 4-3 Results of model trained with monotonic regression loss in diverse decoding task

4.6 本章小结

本工作主要是提出了一种通过利用多样本保序回归改进非自回归翻译模型训练的方法。这一工作主要是针对极大似然损失强制非自回归模型拟合于参考译文的独热分布，从而与丰富多样的真实译文分布的偏差对损失函数进行了改进。本工作通过从模型的输出分布中采集样本，并引入了保序回归损失函数来确保样本的输出概率大小排序与样本的质量相一致。同时，针对非自回归模型学习多峰分布存在一定困难的特点，本工作设计了不同的保序集合来引导模型的后续训练。而在 WMT14 英语到德语和 IWSLT 德语到英语上进行了实验显示，多样本保序回归的方法能够实现一定的 BLEU 值的提升，这说明这一方法能够起到改善非自回归模型输出译文质量的作用。而针对本方法对非自回归模型的输出译文质量提升还较为有限的问题，后续可以通过设计更好的训练策略来减小训练方差的方式加以改善。

第 5 章　神经网络评价指标引导的训练设计

5.1　引言

前两个工作分别探索了通过引入建模上下文信息的解码器，以及通过对多个样本进行保序回归的方式来改善非自回归模型的输出译文质量，而实验的结果也表明，上述工作分别从不同的侧重方向改善了模型的输出译文质量。但这两个工作都面临着理论上的性能上限问题，具体来说，由于非自回归模型的解码器输出是各个位置相互独立的分布，所以在拟合真实分布的时候存在着一定的性能瓶颈，即模型输出分布与真实分布的距离存在一个大于零的下界，这也是导致非自回归模型多峰性出现的一个本质原因。上述观点已经在 Huang 等[63] 证明。除此以外，Huang 等[63] 中也证明了，虽然类似于 DA-Transformer 这样的通过下采样的方式获取输出的工作能够大幅降低这一下界，但其仍然没有办法完全消除这一性能瓶颈。而在上一个工作中也能够看到，多峰性对于模型输出译文质量的影响。这充分说明了无论用何种方式来直接拟合分布，非自回归模型解码器的输出分布无法做到对真实分布做完全的无偏估计。

鉴于直接拟合真实分布存在的一定困难，相应的，可以通过尝试采用不同类型的训练目标。利用评价指标进行训练是一种比较典型的做法，上一个工作中的保序损失函数就是一种利用评价指标来进行训练的方法，只是这种做法并没有直接利用指标的绝对数值，而是利用的多个样本指标的相对大小。而在翻译领域上，已经有一系列的工作[41,68] 直接采用评价指标进行训练。但这些在翻译领域的工作都有一个共同的特点，就是都采用了基于 *n*-gram 的评价指标来引导模型进行训练，而在 Xiang 等[67] 中已经表明，基于 *n*-gram 的评价指标在一致性上是不如基于神经网络的评价指标的。

在本文中将会采用一种类似于 BERTScore[71] 的评价指标来引导模型的训练过程。BERTScore 是一种利用预训练模型 BERT[49] 对文本生成类任务质量进行评价的一个指标，候选译文和参考译文先通过一个 BERT 模型得到相应句子的各个词的输出，再利用候选译文和参考译文各个词之间的余弦相似度来计算句子的质量。由于这一方法能够更准确的建模各个词之间的词意相似性，使得模型的输出译文质量得到了一定的提升。而在本工作中，在利用网络评估指标的基础上，针对直接训练过程中存在的问题：即探索空间过大导致任务复杂度过高，以及神经网络指标只提供准确性信息而不提供流利性信息的特点，采用了一系列的措施来确保编码器提供的上下文信息能够准确的引导模型的输出。本工作在多个模型上进行了实验，证明了在约束探索范围和提供流利度信息的情况下，利用神经网络评价指标能够带来一定的提升。

5.2 相关工作

翻译领域有着许多采用评价指标进行训练的工作，在 Shen 等[68] 的工作中，其通过采集多个样本并优化使其评价指标最大化的方式来改善模型的输出译文质量，最终在自回归翻译上实现了一定的提升。而在非自回归机器翻译中，Shao 等[41] 中已经包含了直接利用评价指标训练模型的思想，其采用了策略梯度算法，利用训练指标作为奖赏进行优化训练。但这些工作采用的指标对模型的输出译文质量带来了限制。

而随着深度学习的发展，近年来也出现了一系列的利用神经网络建模的自然语言评价指标[71–74]。其中，BERTScore[71] 和 Prism[72] 是直接利用模型结构进行计算，而 BLEURT[73] 和 COMET[74] 则都是在利用人工评价指标作为目标对模型进行预训练后得到模型。这为本工作结合神经网络评价指标改进模型训练创造了条件。

5.3 背景

由于本工作主要是基于神经网络的自动评价指标，因此在本节将会对机器翻译任务中常用的一系列的基于神经网络的自动评价指标进行介绍。

5.3.1 基于神经网络的机器翻译自动评价指标

正如前面提到的，机器翻译中存在着许多基于统计的评价指标（如基于 *n*-gram 的 BLEU[44] 、NIST[75] 、METEOR[76] 、chrF[64] 等，以及基于编辑距离的 TER[77] 、PER[78] 、WER[79] 等）。而近些年来随着神经网络技术的发展，一系列的基于神经网络的评价指标也应运而生，具有代表性的有 BERTScore[71] 、BLEURT[73] 、Prism[72] 和 COMET[74] 等。

具体来说，BERTScore 和 Prism 是直接利用模型结构进行训练的模型。BERTScore 主要基于 *n*-gram 的统计评价指标缺乏上下文信息和长距离依赖的问题，通过将候选译文和参考译文句子都输入到 BERT 模型[49] 中，在得到两个句子通过模型后各个词的输出隐状态之后，两两计算余弦相似性，再通过最大匹配的方式来计算准确率和召回率，进而得到评价候选译文的分值。Prism 则是利用了多语言翻译模型的强大性能，通过计算将候选译文和参考译文输入到多语言翻译模型后复述出参考译文和候选译文的概率，以及从源端输出候选译文的概率来评估输出译文的质量。而 BLEURT 和 COMET 则都是通过利用评价指标作为目标对模型进行预训练的方式，得到了一个将句子的分值作为输出的一个模型。

5.4 基于评价指标的训练策略

本节将会详细介绍模型的训练策略，包括利用 BERTScore 的架构构建评价指标训练模型的基本框架，以及其在非自回归模型训练场景下的一系列改进。

5.4.1　模型设计思路

在模型设计上，本工作主要沿袭了前文所采用的 BERTScore[71] 的结构。整个模型用于计算分值的流程如图（5-1）所示：

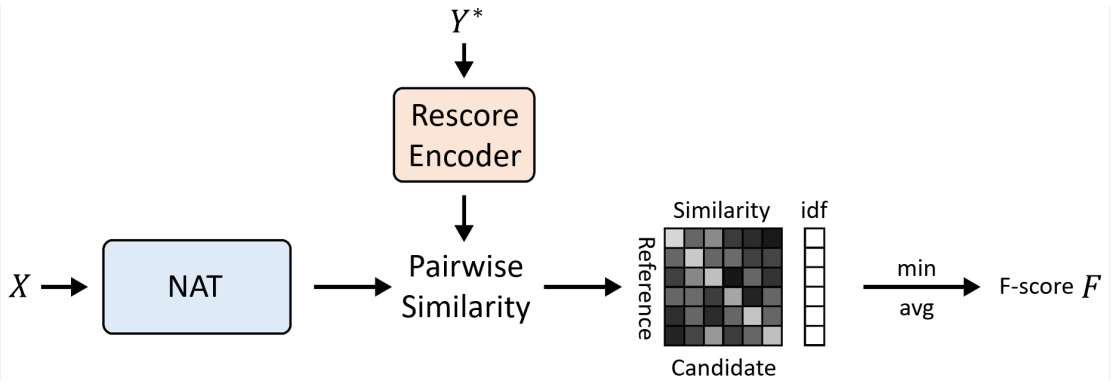


图 5-1 计算翻译质量的模型结构和分数计算流程

Figure 5-1 The structure of the model and the score calculation process

从上图来看，除了图中左侧的非自回归模型以外，本工作还引入了一个类似于 BERT 模型的打分编码器模块，非自回归模型的输出可以用于得到候选译文的隐状态向量 $\langle h_1, ..., h_n \rangle$，而打分编码器模块则是用于计算参考译文的隐状态向量，即 $\langle h_1^*, ..., h_n^* \rangle$。而在得到了上述的隐状态向量之后，本工作将会计算候选译文和参考译文各个词之间的相似性 $\text{Sim}(h_i, h_j^*)$，具体的计算方式上可以采用隐状态的余弦距离来表示相似度，即：

$$\text{Sim}(h_i, h_j^*) = \frac{h_i^{\top} h_j^*}{\|h_i\| \|h_j^*\|}. \tag{5-1}$$

而进一步的，本工作可以采用计算最大匹配的方式，分别寻找候选译文和参考译文中各个词所对应的相似度最大的词来计算精确度和困惑度，并进一步的结合得到 F-score 分值，具体如下式所示：

$$P(\mathbf{y}, \mathbf{y}^*) = \frac{1}{|x|} \sum_{h_i \in \langle h_1, ..., h_n \rangle} \max_{h_j^* \in \langle h_1^*, ..., h_n^* \rangle} \text{Sim}(h_i, h_j^*), \tag{5-2}$$

$$R(\mathbf{y}, \mathbf{y}^*) = \frac{1}{|x|} \sum_{h_i^* \in \langle h_1^*, ..., h_n^* \rangle} \text{IDF}(y_i^*) \max_{h_j \in \langle h_1, ..., h_n \rangle} \text{Sim}(h_j, h_i^*), \tag{5-3}$$

$$F(\mathbf{y}, \mathbf{y}^*) = \frac{2 \cdot P(\mathbf{y}, \mathbf{y}^*) \cdot R(\mathbf{y}, \mathbf{y}^*)}{P(\mathbf{y}, \mathbf{y}^*) + R(\mathbf{y}, \mathbf{y}^*)}. \tag{5-4}$$

需要注意的是，上式中的 IDF 又被称为反文档频率，即统计文档中相应的词出现在文档中的频率的负对数，该值越高代表这一词更能表示语义信息。

而相应的，本工作可以利用得到的 F-score 的分值设计训练目标进行训练。

而就模型的训练流程而言，在已有一个非自回归模型的基础上，本工作首先要对打分解码器进行预训练。而正如前文所提到的，如果直接采用极大似然损失去训练打分解码器的话则会带来严重的源端信息泄露，使得各个词的隐状态无法融合上下文的信息。相应的，本工作也采用了类似 BERT[49] 的训练方式，即通过利用掩码遮盖 15% 的输入词，并基于其他的词来预测掩码盖住的词，从而使得打分解码器能够学到上下文的信息。在训练流程上，则是先采用前述方式对打分解码器进行预训练，再利用预训练之后的解码器来引导模型非自回归翻译模型的输出。

5.4.2 主要改进

从理论上讲，由于 F-score 本身是一种表示候选译文与参考译文准确性的比较好的结构，而且这种结构能够直接由模型输出计算得来。因此，将其用于非自回归训练能够得到比较好的结果。但在实际训练过程中，采用上述指标训练模型的结果确是模型解码的输出为大量的高频词（如逗号）的重复，即没有学习到相关的指标的知识。本节将会针对在非自回归上出现的这一问题提出相应的改进策略。

5.4.2.1 核心问题

正如前文所述，作为理想的神经网络翻译评价指标的 F-score 却没能起到引导模型输出的作用。究其原因，可以追溯为以下两者的矛盾，即探索与利用之间的矛盾、以及准确与流利之间的矛盾。具体来说包括了以下一些：

- 探索与利用之间的矛盾：探索-利用困境（Exploration-exploitation dilemma）[80] 是增强学习领域的一个重要概念，具体来说，探索是指的根据模型的输出分布通过采样得到更高的输出，而利用是指的是模型取输出分布中概率最大的样本来得到输出。就多臂老虎机[81] 这样的增强学习任务而言，由于任务本身并没有标准的标签，所以随着任务空间的加大，纯粹采用取概率最大的输出进行训练和取依照概率分布进行采样的输出进行训练都会使得期望损失函数升高。而对有监督的翻译而言，由于数据集中给出了一种参考译文，因此如果让模型强制以参考译文作为目标的话，虽然学习的结果仍然会存在很多的问题，但仍然能够得到一个质量尚可的输出；反之，如果是在整个词表空间内探索的话，模型大概率会采到偏离合理译文的输出，这将使得模型难以学到正确的译文。

- 准确与流利之间的矛盾：以 BERTScore 为代表的基于神经网络的评价指标主要是针对的模型对上下文的敏感性不足，导致对同义词等打分存在偏差的问题来进行改进，在多个任务上也取得了不错的结果。但需要说明的是，由于 BERTScore 采用的是编码器结构，该结构得到的输出更倾向于得到句子中各个词的含义，而对于句子上下文的连贯程度并不敏感，加之 BERT 模型在训练时是用的较为干净的单语语料，相应的，对多峰性严重，较为支离破碎的非自回归模型的输出，模型往往无法给出较为准确的隐状态，这样学习的结果是使得对于一部分初始的时候相似度较大的词，模型倾向于进一步加大相应位置的相似性，

这样会使得非自回归模型的多峰性会进一步加大。因此，需要引入一些神经网络的建模方式，使得模型得到的分值中能够包含流利度的信息。

5.4.2.2　针对探索空间的改进

正如前文所述的那样，对于非自回归机器翻译任务而言，由于其任务的探索空间过大，导致模型难以朝一个正确的训练目标进行收敛。因此，一种可行的解决方式是压缩模型探索的空间。而就非自回归机器翻译任务来说，设词表大小和输出句子长度分别为 $|V|$ 和 T，那么非自回归任务的模型搜索空间则是可以表述为 $|V|^T$，这对任务的学习带来了一定的难度。而相应的，在本工作中采取的方法是对模型探索的范围进行压缩，具体来说，若有训练样本 $(\mathbf{x},\mathbf{y}) \in D$，设词集合 $A_{(\mathbf{x},\mathbf{y}),\theta}$ 满足：

$$A_{(\mathbf{x},\mathbf{y}),\theta} = \{y | y \in \{argmax_i P(y_i|\mathbf{x};\theta), i = 1, ..., T\} \cup \mathbf{y}\}. \tag{5-5}$$

相应的，本工作可以将各个位置中采样到各个词的概率表示为：

$$P(y_i = w|x) = \begin{cases} \frac{P(y_i=w|x;\theta)}{\Sigma_{w' \in A} P(y_i=w'|x;\theta)} & w \in A_{(\mathbf{x},\mathbf{y}),\theta} \\ 0 & w \notin A_{(\mathbf{x},\mathbf{y}),\theta} \end{cases}. \tag{5-6}$$

从上式中可以看到，模型在训练时的采样的空间从原来的整个词表，变为了只采样各个位置输出概率最大的词和参考译文的词的交集。相应的，任务的搜索空间就从原来的 $|V|^T$ 变成了最多 $(2 \cdot T)^T$，由于 $T << |V|$，相应的模型的探索空间也得到了大幅的压缩。

5.4.2.3　针对流利度的改进

正如前面所提到的，由于采用以 BERTScore 为代表的基于神经网络的评价指标并没有提供有关流利度的惩罚信息，而这又是非自回归翻译模型中比较突出的问题，相应的，这对模型的输出译文质量带来了显著的影响，为了解决这一问题，一种可行的做法引入一个语言模型来表示上下文信息。通常来说，一个流利的句子会被认为较为自然易懂，而语言模型恰好能够建模句子中下文出现的概率，这使得其成为了建模句子流利度的不二选择。具体来说，可以通过一个句子经过模型的困惑度来表示出句子的流利度，即：

$$\text{ppl}(\mathbf{y}) = \exp\left(\frac{1}{T}\Sigma_{i=1}^{T} \log P(y_i|\mathbf{x}, y_{<i};\theta_{LM})\right). \tag{5-7}$$

虽然语言模型有着适合建模句子流利度的特点，但需要注意的是，由于语言模型也是利用从真实句子中采样得到的训练集来进行的训练，由于训练集样本有限以及样本噪声等因素的影响，模型对一些句子输出的困惑度会显著的偏离真实情况，比如对一些高频的输出（如多个逗号的重复）会给出一个较高的分值。因此，本工作在 ppl 的基础上会乘上一个约束项，即：

$$\text{repeat_penalty}(\mathbf{y}) = \begin{cases} \text{len}(\{y \in \mathbf{y}\})/T & \text{len}(\{y \in \mathbf{y}\})/T < 0.7 \\ 1 & \text{len}(\{y \in \mathbf{y}\})/T \geq 0.7 \end{cases}. \tag{5-8}$$

通过上述的惩罚系数，本工作利用了一些人工的规则对一些重复词居多的句子进行了惩罚，从而避免了语言模型引导到过于简单的输出的现象。最终本工作得到的分值如下形式所示：

$$\text{Score}_{\text{LM}}(\mathbf{y}) = \text{ppl}(\mathbf{y}) \cdot \text{repeat_penalty}(\mathbf{y}). \tag{5-9}$$

$$\text{Score}(\mathbf{y}, \mathbf{y}^*) = F(\mathbf{y}, \mathbf{y}^*) + \text{Score}_{\text{LM}}(\mathbf{y}). \tag{5-10}$$

5.4.3 训练目标

在以上的训练目标基础上，本工作可以采用如下的几种训练目标，包括：

- 直接优化：通过上述的计算过程可以计算得到的分值可以直接反传给模型进行计算，因此可以直接将非自回归模型输出的隐状态作为打分模块的输入来得到分值，并将其反传给非自回归模型进行训练。此时，模型的损失函数为 $L = -\text{Score}(\mathbf{y}, \mathbf{y}^*)$。
- 作为增强学习的激励信号：由于直接优化时只能利用隐状态而非具体的词进行训练，相应的另一种可行的办法是通过一定规则下的采样并计算模型的奖赏，再通过策略梯度算法来估计模型在相应的指标上的质量。其好处是能够针对句子做较为直接的优化，但在训练时会有一定的方差。
- 最优参考译文训练：从之前的一系列工作中可以看出，虽然非自回归翻译模型训练最终的目标是使得非自回归模型能够学到分数最高的译文，但同时需要考虑非自回归模型是否能够学到相应的知识。因此本工作中可以将适合模型学习的译文用以下的一个约束优化问题来表示：

$$\begin{aligned} \max_{\mathbf{y}} \quad & \text{Score}(\mathbf{y}, \mathbf{y}^*), \\ \text{s.t.} \quad & P(\mathbf{y}|\mathbf{x}; \theta) \geq \tau. \end{aligned} \tag{5-11}$$

上式可以通过拉格朗日乘子法来计算，去除常数项后，可以得到所要寻找的训练目标如下式所示：

$$\text{argmax}_{\mathbf{y}} \text{Score}(\mathbf{y}, \mathbf{y}^*) + \lambda P(\mathbf{y}|\mathbf{x}; \theta). \tag{5-12}$$

由于上述任务本身并不满足非凸的性质，并且模型的参数 θ 随着模型的训练过程始终在变化，因此上述任务在实际训练时，是通过神经网络 $\theta(\mathbf{y}, \mathbf{y}^*)$ 来近似的寻找上述最优的目标句子 $\mathbf{y}'$，再将模型输出分布与真实分布来计算极大似然损失来引导模型的训练。

5.4.4　训练流程

上述模型中由于涉及到的模块众多，因此在训练的过程中采用了类似于之前训练上下文解码器的方式，即将多个模块分别进行训练。具体来说其包含了以下的几个步骤：

(1) 对非自回归模型进行预训练。

(2) 固定非自回归模型参数，对目标语言编码器进行训练。

(3)（可选）固定非自回归模型参数，对目标语言模型进行训练。

(4)（可选）固定非自回归模型和目标语言编码器和语言模型的参数，对目标句子求解器进行训练进行训练。

(5)（可选）将词嵌入模块进行拆解，使得语言编码器和非自回归模型采用不同的词嵌入网络模块。

(6) 固定目标语言编码器和目标语言模型，对非自回归解码器进行微调。

5.5　实验结果与分析

本部分的主要内容是对本工作中初步实验的设置，以及初步的实验结果进行介绍。并相应的对实验的结果进行一定的分析。

5.5.1　实验设置

本部分将会对实验的训练数据、模型参数设置、训练设置以及评价指标等方面进行介绍。

5.5.1.1　训练数据

本工作在 WMT14 英语到德语任务（WMT14 En→De ）上进行了实验，相应的本工作采用了这一任务的数据集。在本实验中，本工作采用的数据集并没有对训练的数据进行清洗，因此训练数据规模仍然与原始的数据一致，即约 450 万句。对于这些数据集，本工作采用了 32000 步的 BPE 模型[51] 来对语料进行预处理，同时对上述几个任务本工作也共享了源端和目标端的词表。

5.5.1.2　知识蒸馏

在知识蒸馏方面，本工作对上述数据集均采用了通过知识蒸馏的数据进行训练。对于本实验来说，本工作都采用了 Kim 等[13] 中的做法，即将非自回归训练时的目标端换成了一个通过自回归的模型解码后得到的译文。需要说明的是，对于基于 CMLM 模型微调的实验，本工作采用了 Transformer-Big 模型训练得到的自回归翻译模型进行蒸馏；而对于基于标准 NAT 模型训练的模型，本工作则采用了基于 Transformer-Base 的模型进行训练。上述蒸馏过程中都采用了大小为 5 的集束解码得到相应的目标端。

5.5.1.3 基线模型

考虑到非自回归模型的结构相似性，本工作相应的与以下两类非自回归模型进行对比：

- 基础的非自回归模型：如标准 NAT[2] 模型和 CMLM 模型[18]。
- 对非自回归模型的改进模型：如 Bags-of-ngram 模型[14]、GLAT 模型[21]、基于 CMLM 微调的 AXE 模型[36] 和基于 CMLM 的 OAXE 模型[52]。

5.5.1.4 模型结构

在本实验中，本工作采用的非自回归模型的结构与标准的 Transformer-Base 结构一致，即 6 个编码器层和 6 个解码器层。而就新引入的模块而言，本次实验中引入的目标端语言编码器包含 4 个编码器层，目标端语言解码器则包含 4 个解码器层，且不包含交叉注意力机制，而计算最优译文的模块则包含 2 个解码器层。而就每层具体的结构而言，词嵌入的维数为 512 维，中间层隐状态维数为 2048 维，而注意力机制的头数为 8 个。源端和目标端共享词嵌入模块，且两个解码器的解码器输出投影层与目标嵌入层共享相同的参数。

5.5.1.5 其他实现细节

本工作的模型和训练过程的代码实验都是在基于 Pytorch 框架[55] 下的 fairseq[56] 工具包上完成。在训练阶段，本工作的实验都是在一个以及预训练过的模型上实现的。在训练的批处理上，本工作每次都使用含有 64000 个词的批来微调本工作的模型。在各个微调任务中，本工作采用的学习率都是 $5e-4$，并随着平方根倒数时间而衰减。而就优化器方面，本实验采用的是 Adam 优化器[54]，其中对基于 CMLM 的模型，$\beta=(0.9,0.999)$；而对于其他的模型，$\beta=(0.9,0.98)$。而在评估质量时，本工作对 5 个最佳检查点进行平均，并相应的设置 5 个长度候选。而就评价指标而言，本实验中本工作对翻译质量的评价主要是采用的是翻译中常用的 BLEU 指标。在本实验中，对 WMT14 英语到德语双向任务，本工作采用 multi-bleu.perl 来对模型指标进行测量。

5.5.2 主要的实验结果

本工作在前述的 3 种训练策略进行了反复的实验，下表中展示了本工作的几种方法在 WMT14 En→De 数据集上的结果：

表 5-1 几种利用评价指标训练模型的策略的结果

Table 5-1 The results of several strategies for training models using evaluation indicators

	WMT14 En→De	**加速比**
Vanilla NAT	19.28	**×15.6**
CMLM	18.05	×15.0
Bags-of-Ngram	20.90	×10.7
CMLM+AXE	23.53	×15.3
GLAT	25.21	×15.3
CMLM+OAXE	**26.10**	×15.3
CMLM(ours)	16.33	×14.1
+ 直接优化	×	\
+ 增强学习的激励信号	×	\
+ 最优参考译文训练	23.22	×14.1

从上表中可以看出，采用前述的两种策略进行训练时结果用“×”表示，这代表模型的输出译文质量将会直接收敛到 0，具体表现为输出大量重复的高频词，这也表明对于非自回归翻译（乃至语言翻译）这样的标签较弱的任务而言，完全采用增强学习目标这样的弱标签会使得模型难以收敛到训练目标。而只有采用通过引导模型去接近一个模型能够生成且正确率较高的译文，才能使得模型的输出译文质量得到提升。

而从本方法的结果来看，虽然本方法相比器一些经典的改进方法（如 AXE 、OAXE 和 GLAT ）之间存在着一定的差距，但是，这在一定程度上是由于实现的基线模型输出译文质量较低，以及所采用的语言模型和编码器是经过基于平行语料中的单语语料进行的训练，通过进一步的充分研究，本方法能够实现一定的提升。

5.5.3 分析实验

本节主要是针对本工作的方法，以及实验得到的一系列结果进行分析。

5.5.3.1 消融实验

本实验主要是分析本方法中的两个技巧——约束探索空间和引入语言模型计算困惑度的效果。具体来说，本实验控制其他参数不变，仅调整模型训练时是否压缩采样空间和训练时的评价指标，在 WMT14 En→De 进行了实验，结果如下表所示：

表 5-2 约束探索空间和引入语言模型的消融实验

Table 5-2 Ablation study for sample constraint and calculating perplexity

约束探索空间	训练评价指标		BLEU
	F-score	困惑度	
			16.33
√	√		×
√		√	15.42
	√	√	×
√	√	√	**23.22**

首先，从引入探索约束的角度来看，如果在相同的条件下不引入采样的约束，模型最终无法输出相应的译文，这说明了引入探索约束的重要性。其次，本工作探索了在 F-score 之外采用语言模型困惑度的结果。从模型的训练结果来看，首先，采用 F-score 训练的模型最终无法得到相应的 BLEU 值，这说明直接采用 F-score 无法提供对非自回归翻译至关重要的上下文（或者流利度）的信息；同时，本工作也采用了仅利用语言模型的困惑度进行实验，结果表明模型也能输出一定的译文，但其 BLEU 值相比原始的模型有所降低，说明仍然需要 F-score 模型来提供上下文准确性相关的信息。

5.5.3.2 在其他评价指标上的结果

由于本工作的模型采用了类似于 BERTscore[71] 结构的神经网络评价指标对模型进行训练，因此一个显而易见的假设是本工作中的方法能够输出的译文能够在 BERTScore 上有所提升。对此本工作利用模型 bert-base-multilingual-cased 进行了相应的实验，结果如下表所示：

表 5-3 利用评价指标训练的模型在多个评价指标上的结果

Table 5-3 Results of models trained with evaluation indicators on multiple evaluation metrics

	WMT14 En→De	
	BLEU	BERTScore
CMLM(ours)	16.33	0.4669
+ 最优参考译文训练	**23.22**	**0.5686**

而上述结果也表明，本工作的方法不仅能够在基于 *n*-gram 的指标 BLEU 上得到提升，在基于神经网络的评价指标 BERTScore 上也得到了提升，这也在一定程度上印证了本工作的方法达到了相应的目标。

5.6　本章小结

本工作主要是提出了一种通过网络评估指标引导非自回归翻译的方法。针对非自回归模型本身在拟合真实分布上存在着一定的理论上限的问题，本工作尝试通过引入一种类似于神经网络评价指标 BERTScore 的网络结构来将非自回归模型输出译文的质量作为训练的目标。而对于非自回归任务采用神经网络评价指标训练时存在的探索空间过大和 BERTScore 指标对流利度不敏感的问题，本工作进一步的通过约束模型采样输出的范围和引入语言模型困惑度来改进模型的训练。本工作在 WMT14 英语到德语任务上进行了实验，结果表明虽然直接利用基于神经网络的评价指标训练模型存在一定的难度，但通过一个合理的训练目标设计，模型的输出译文质量能够得到显著提升。而如果能够引入更强的基线模型和计算评价指标的神经网络结构，本工作的模型能够实现一个更高的输出译文质量。

第 6 章　总结和展望

6.1　总结

机器翻译是一种在理论层面和应用层面都具有广泛的价值的一种技术。而近年来，随着计算机硬件性能的提升和深度学习技术的进一步发展，神经机器翻译逐渐取代了基于规则和统计的机器翻译，成为了翻译领域的技术主流。一般而言，神经机器翻译多采用编码器-解码器的网络架构，其生成译文的过程也是采用的自回归的过程，即预测下一个词时需要给出之前已经输出的译文，这一串行的过程一方面会使得训练和解码的用时大幅增加，另一方面也使得训练和解码的过程变得不一致，从而产生曝光偏差问题。而 Transformer 模型的出现使得模型在训练阶段得以实现了并行的训练，但其在解码阶段自回归生成的问题仍然没有解决。而非自回归机器翻译则是针对这一问题提出的一种重要的解决方式，其在利用编码器对输出句子长度进行预测的基础上，对各个位置的输出词进行独立的预测，从而实现了并行的解码输出。

虽然非自回归机器翻译成功的实现了并行的解码，但其解码得到译文相较于非自回归的译文在质量上存在着较大的差距，其具体表现在解码得到的译文中有着大量的重复和漏译错译现象。从理论上来说，这一问题产生的原因是由于非自回归翻译任务仅根据源端译文和相应的词的位置进行预测，缺乏其他位置的上下文信息，导致任务难度相对变大；而极大似然损失函数则未能很好的解决翻译任务中存在一句话对应多种翻译的问题，使得模型难以给各个位置的输出词一个准确的损失，导致非自回归机器翻译任务普遍面对着多峰性问题。本工作主要是针对非自回归模型的训练目标本身存在的缺点和其在非自回归机器翻译中应用时出现的一系列问题，循序渐进又各有侧重的设计了对非自回归模型的训练进行改进的策略，具体的工作上包括了以下的 3 个方面：

- 本工作引入了一种结合上下文信息的训练目标设计方法。具体来说，非自回归模型本身在经过一系列预训练后具有一定的内生的上下文知识，但非自回归模型的极大似然损失函数中缺乏对结合模型输出上下文的信号。针对这一问题，本工作引入了一个建模在给出上下文的条件下模型在相应位置的输出分布的先见解码器模型，在训练时，通过将模型的输出词作为先见解码器的输入来计算各个位置基于相应上下文信息的分布，并使得模型在降低极大似然损失的同时能够降低模型输出分布和先见解码器的输出分布之间的 Jensen 散度。同时本工作针对这一工作设计了一系列的训练策略，并在多个数据集上进行了实验，而实验的结果表明，本工作的方法在多个数据集上都实现了超过 2 个点的显著的 BLEU 值提升，且在准确度和流利度上都能够实现一定的提升，这说明通过在训练目标中利用模型先验知识引入上下文信息能够显著缓解非自回归翻译中的多峰性问题，改善模型性能。
- 本工作提出了一种通过多样本保序回归的损失函数设计的方法。具体而

言，虽然以 DA-Transformer 为代表的一系列模型在表达能力已经有了很大的提升，但极大似然损失强制非自回归模型拟合于参考译文的独热分布，与丰富多样的真实译文分布存在着偏差。对此，本工作通过从模型的输出分布中采集样本，并引入损失函数来确保样本的输出概率大小排序与样本的质量相一致。同时，本工作针对非自回归模型学习多峰分布存在一定困难的特点，设计了不同的保序集合来引导模型的后续训练。本工作在 WMT14 英语到德语和 IWSLT14 德语到英语上进行了实验，结果显示本工作的方法能够实现一定的 BLEU 值的提升，同时在多样性任务上实现了显著的提升。这一方面说明本工作的方法能够起到改善非自回归模型性能的作用，另一方面也展示出本工作的方法对输出分布的质量的改善作用。

- 本工作提出了一种神经网络评价的引导训练方法。其背景是，虽然经过一系列的改进，非自回归模型在拟合上下文上有了一定的提高，但非自回归模型本身在拟合真实分布上存在着一定的理论上限的问题。相应的，本工作尝试通过引入一种类似于神经网络评价指标 BERTScore 的网络结构来将非自回归模型输出译文的质量作为训练的目标。而对于非自回归任务采用神经网络评价指标训练时存在的探索空间过大和 BERTScore 指标对流利度不敏感的问题，本工作进一步的通过约束模型采样输出的范围和引入语言模型困惑度来改进模型的训练。本工作在 WMT14 英语到德语任务上进行了实验，结果表明虽然直接利用基于神经网络的评价指标训练模型存在一定的难度，但通过一个合理的训练目标设计，模型的性能能够得到显著提升。而如果能够引入更强的基线模型和计算评价指标的神经网络结构，本工作的模型能够实现一个更高的性能。

通过上述的一系列策略，本工作实现了针对不同的场景下的非自回归的机器翻译的性能提升。而需要特别说明的是，这些方法虽然理论上在自回归模型中都能够加以应用，但其要完全得到真实的输出需要进行串行解码的过程，这使得模型通常都无法直接采用这些策略进行训练，而需要采用一定的变通实施（比如提前解码出固定的样本，用模型估计当前位置下的期望指标等）的方式，这会对模型带来一定的训练偏差。而非自回归模型的并行解码特性、以及训练过程和解码过程相一致的特点，使得这些策略在训练时得到使用成为可能。这一系列工作也展现出了非自回归机器翻译模型理论研究的潜力。

6.2 展望

近些年来随着非自回归模型技术的演进，其模型的性能已经能够接近甚至超过自回归翻译模型的性能，但其本身要实现更广泛的应用，还需要解决一系列的理论问题：

- 结合前述几种方法的改进：上述几种模型针对不同的非自回归模型采用了不同的策略进行改进，虽然这些方法有其实现模型的客观条件（如保序回归损失函数的应用依赖于 DA-Transformer 建模分布的提升），但倘若通过合理的方式加以建模，上述三种方法是可以加以结合并实现性能提升的，这也是后续探索的

一种可行方向。

- 减小显存开销的情况下提升生成质量：以 DA-Transformer 为代表的基于降采样的非自回归翻译模型虽然可以实现在单轮迭代的情况下且在不需要知识蒸馏的情况下达到与自回归模型相当的性能水平，但其在解码过程中需要一个长度为源端句子长度数倍的解码器输入，这一方面会产生大量的显存开销，另一方面也会使得模型解码输出的长度受到限制。因此，采取一定的措施在减小显存开销的情况下实现较高质量的输出是一个潜在的研究方向。
- 实现可变输出下的模型输出：非自回归的模型往往是利用的编码器-解码器模型结构，根据源端的输入给出一个固定长度的输出。但在同声传译、篇章翻译和对话等场景下，可以看到输入信息的长度是逐渐增长的，此时非自回归模型快速生成的特点就难以得到充分发挥。如何将非自回归的优势应用在可变长度的场景之中是接下来需要解决的一个问题。

除此以外，非自回归模型的一系列的特性也为我们进行一些生成式的模型的理论研究（如生成式扩散模型）创造了条件，这些都是后续可以探索和研究的方向。

参考文献

[1] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [C/OL]//Bengio Y, LeCun Y. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. http://arxiv.org/abs/1409.0473.

[2] Gu J, Bradbury J, Xiong C, et al. Non-autoregressive neural machine translation [C/OL]//6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. https://openreview.net/forum?id=B1l8BtlCb.

[3] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks [C]// Ghahramani Z, Welling M, Cortes C, et al. Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, NIPS 2014, December 8-13 2014, Montreal, Quebec, Canada. 2014: 3104-3112.

[4] Kalchbrenner N, Blunsom P. Recurrent continuous translation models [C/OL]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, 2013: 1700-1709. https://aclanthology.org/D13-1176/.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]//Guyon I, von Luxburg U, Bengio S, et al. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. 2017: 5998-6008.

[6] Ranzato M, Chopra S, Auli M, et al. Sequence level training with recurrent neural networks [C/OL]//Bengio Y, LeCun Y. 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016. http://arxiv.org/abs/1511.06732.

[7] Weaver W. Translation [C]//Proceedings of the Conference on Mechanical Translation. 1952.

[8] Council N R, et al. Language and machines: Computers in translation and linguistics [J]. 1966.

[9] Brown P F, Pietra S D, Pietra V J D, et al. The mathematics of statistical machine translation: Parameter estimation [J]. Comput. Linguistics, 1993, 19(2): 263-311.

[10] Gale W A, Church K W. A program for aligning sentences in bilingual corpora [J]. Comput. Linguistics, 1993, 19(1): 75-102.

[11] Koehn P, Och F J, Marcu D. Statistical phrase-based translation [C/OL]//Hearst M A, Ostendorf M. Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003. The Association for Computational Linguistics, 2003. https://aclanthology.org/N03-1017.

[12] Schuster M, Paliwal K K. Bidirectional recurrent neural networks [J/OL]. IEEE Trans. Signal Process., 1997, 45(11): 2673-2681. https://doi.org/10.1109/78.650093.

[13] Kim Y, Rush A M. Sequence-level knowledge distillation [C/OL]//Su J, Carreras X, Duh K. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,

EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. The Association for Computational Linguistics, 2016: 1317-1327. https://doi.org/10.18653/v1/d16-1139.

[14] Zhou C, Gu J, Neubig G. Understanding knowledge distillation in non-autoregressive machine translation [C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. https://openreview.net/forum?id=BygFVAEKDH.

[15] Ding L, Wang L, Liu X, et al. Understanding and improving lexical choice in non-autoregressive translation [C/OL]//9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. https://openreview.net/forum?id=ZTFeSBIX9C.

[16] Sun Z, Yang Y. An EM approach to non-autoregressive conditional sequence generation [C/OL]//Proceedings of Machine Learning Research: volume 119 Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. PMLR, 2020: 9249-9258. http://proceedings.mlr.press/v119/sun20c.html.

[17] Shao C, Wu X, Feng Y. One reference is not enough: Diverse distillation with reference selection for non-autoregressive translation [C/OL]//Carpuat M, de Marneffe M, Ruíz I V M. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022. Association for Computational Linguistics, 2022: 3779-3791. https://doi.org/10.18653/v1/2022.naacl-main.277.

[18] Ghazvininejad M, Levy O, Liu Y, et al. Mask-predict: Parallel decoding of conditional masked language models [C/OL]//Inui K, Jiang J, Ng V, et al. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. Association for Computational Linguistics, 2019: 6111-6120. https://doi.org/10.18653/v1/D19-1633.

[19] Kasai J, Cross J, Ghazvininejad M, et al. Non-autoregressive machine translation with disentangled context transformer [C]//International conference on machine learning, ICML 2020. PMLR, 2020: 5144-5155.

[20] Geng X, Feng X, Qin B. Learning to rewrite for non-autoregressive neural machine translation [C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021. 2021: 3297-3308.

[21] Qian L, Zhou H, Bao Y, et al. Glancing transformer for non-autoregressive neural machine translation [C/OL]//Zong C, Xia F, Li W, et al. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021. Association for Computational Linguistics, 2021: 1993-2003. https://doi.org/10.18653/v1/2021.acl-long.155.

[22] Kingma D P, Welling M. Auto-encoding variational bayes [C/OL]//Bengio Y, LeCun Y. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings. 2014. http://arxiv.org/abs/1312.6114.

[23] Ma X, Zhou C, Li X, et al. Flowseq: Non-autoregressive conditional sequence generation with generative flow [C/OL]//Inui K, Jiang J, Ng V, et al. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7,

2019. Association for Computational Linguistics, 2019: 4281-4291. https://doi.org/10.18653/v1/D19-1437.

[24] Rezende D J, Mohamed S. Variational inference with normalizing flows [C/OL]//Bach F R, Blei D M. JMLR Workshop and Conference Proceedings: volume 37 Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. JMLR.org, 2015: 1530-1538. http://proceedings.mlr.press/v37/rezende15.html.

[25] Shu R, Lee J, Nakayama H, et al. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior [C/OL]//The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020: 8846-8853. https://ojs.aaai.org/index.php/AAAI/article/view/6413.

[26] Bao Y, Huang S, Xiao T, et al. Non-autoregressive translation by learning target categorical codes [C/OL]//Toutanova K, Rumshisky A, Zettlemoyer L, et al. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021. Association for Computational Linguistics, 2021: 5749-5759. https://doi.org/10.18653/v1/2021.naacl-main.458.

[27] Bao Y, Zhou H, Huang S, et al. latent-glat: Glancing at latent variables for parallel text generation [C/OL]//Muresan S, Nakov P, Villavicencio A. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. Association for Computational Linguistics, 2022: 8398-8409. https://doi.org/10.18653/v1/2022.acl-long.575.

[28] Ran Q, Lin Y, Li P, et al. Guiding non-autoregressive neural machine translation decoding with reordering information [C/OL]//Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press, 2021: 13727-13735. https://ojs.aaai.org/index.php/AAAI/article/view/17618.

[29] Dyer C, Chahuneau V, Smith N A. A simple, fast, and effective reparameterization of IBM model 2 [C/OL]//Vanderwende L, III H D, Kirchhoff K. Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, NAACL 2013, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA. The Association for Computational Linguistics, 2013: 644-648. https://aclanthology.org/N13-1073/.

[30] Song J, Kim S, Yoon S. Alignart: Non-autoregressive neural machine translation by jointly learning to estimate alignment and translate [C/OL]//Moens M, Huang X, Specia L, et al. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. Association for Computational Linguistics, 2021: 1-14. https://doi.org/10.18653/v1/2021.emnlp-main.1.

[31] Och F J, Ney H. A systematic comparison of various statistical alignment models [J]. Computational Linguistics, 2003, 29(1): 19-51.

[32] Libovický J, Helcl J. End-to-end non-autoregressive neural machine translation with connectionist temporal classification [C/OL]//Riloff E, Chiang D, Hockenmaier J, et al. Proceedings

of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, Brussels, Belgium, October 31 - November 4, 2018. Association for Computational Linguistics, 2018: 3016-3021. https://doi.org/10.18653/v1/d18-1336.

[33] Huang F, Zhou H, Liu Y, et al. Directed acyclic transformer for non-autoregressive machine translation [C/OL]//Chaudhuri K, Jegelka S, Song L, et al. Proceedings of Machine Learning Research: volume 162 International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA. PMLR, 2022: 9410-9428. https://proceedings.mlr.press/v162/huang22m.html.

[34] Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks [C]//Proceedings of the 23rd International Conference on Machine learning, ICML 2006. 2006: 369-376.

[35] Shao C, Ma Z, Feng Y. Viterbi decoding of directed acyclic transformer for non-autoregressive machine translation [C/OL]//Goldberg Y, Kozareva Z, Zhang Y. Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. Association for Computational Linguistics, 2022: 4390-4397. https://aclanthology.org/2022.findings-emnlp.322.

[36] Ghazvininejad M, Karpukhin V, Zettlemoyer L, et al. Aligned cross entropy for non-autoregressive machine translation [C/OL]//Proceedings of Machine Learning Research: volume 119 Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. PMLR, 2020: 3515-3523. http://proceedings.mlr.press/v119/ghazvininejad20a.html.

[37] Kuhn H W. The hungarian method for the assignment problem [J]. Naval research logistics quarterly, 1955, 2(1-2): 83-97.

[38] LeCun Y, Chopra S, Hadsell R, et al. A tutorial on energy-based learning [J]. Predicting structured data, 2006, 1(0).

[39] Tu L, Pang R Y, Wiseman S, et al. ENGINE: energy-based inference networks for non-autoregressive machine translation [C/OL]//Jurafsky D, Chai J, Schluter N, et al. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. Association for Computational Linguistics, 2020: 2819-2826. https://doi.org/10.18653/v1/2020.acl-main.251.

[40] Shao C, Feng Y, Zhang J, et al. Retrieving sequential information for non-autoregressive neural machine translation [C/OL]//Korhonen A, Traum D R, Màrquez L. Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers. Association for Computational Linguistics, 2019: 3013-3024. https://doi.org/10.18653/v1/p19-1288.

[41] Shao C, Feng Y, Zhang J, et al. Sequence-level training for non-autoregressive neural machine translation [J/OL]. Comput. Linguistics, 2021, 47(4): 891-925. https://doi.org/10.1162/coli_a_00421.

[42] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning [J/OL]. Mach. Learn., 1992, 8: 229-256. https://doi.org/10.1007/BF00992696.

[43] Thomas P S, Brunskill E. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines [J/OL]. CoRR, 2017, abs/1706.06643. http://arxiv.org/abs/1706.06643.

[44] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation [C/OL]//Proceedings of the 40th Annual Meeting of the Association for Computa-

tional Linguistics, ACL 2002, July 6-12, 2002, Philadelphia, PA, USA. ACL, 2002: 311-318. https://aclanthology.org/P02-1040/. DOI: 10.3115/1073083.1073135.

[45] Shao C, Zhang J, Feng Y, et al. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation [C/OL]//The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020: 198-205. https://ojs.aaai.org/index.php/AAAI/article/view/5351.

[46] Shao C, Feng Y. Non-monotonic latent alignments for ctc-based non-autoregressive machine translation [J/OL]. CoRR, 2022, abs/2210.03953. https://doi.org/10.48550/arXiv.2210.03953.

[47] Ma Z, Shao C, Gui S, et al. Fuzzy alignments in directed acyclic graph for non-autoregressive machine translation [J/OL]. CoRR, 2023, abs/2303.06662. https://doi.org/10.48550/arXiv.2303.06662.

[48] Feng Y, Gu S, Guo D, et al. Guiding teacher forcing with seer forcing for neural machine translation [C/OL]//Zong C, Xia F, Li W, et al. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021. Association for Computational Linguistics, 2021: 2862-2872. https://doi.org/10.18653/v1/2021.acl-long.223.

[49] Devlin J, Chang M, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [C/OL]//Burstein J, Doran C, Solorio T. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics, 2019: 4171-4186. https://doi.org/10.18653/v1/n19-1423.

[50] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training [J]. 2018.

[51] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units [C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics, 2016. https://doi.org/10.18653/v1/p16-1162.

[52] Du C, Tu Z, Jiang J. Order-agnostic cross entropy for non-autoregressive machine translation [C/OL]//Meila M, Zhang T. Proceedings of Machine Learning Research: volume 139 Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. PMLR, 2021: 2849-2859. http://proceedings.mlr.press/v139/du21c.html.

[53] Post M. A call for clarity in reporting BLEU scores [M/OL]//Proceedings of the Third Conference on Machine Translation: Research Papers. Belgium, Brussels: Association for Computational Linguistics, 2018: 186-191. https://www.aclweb.org/anthology/W18-6319.

[54] Kingma D P, Ba J. Adam: A method for stochastic optimization [C]//Bengio Y, LeCun Y. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015.

[55] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library [C]//Wallach H M, Larochelle H, Beygelzimer A, et al. Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing

Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. 2019: 8024-8035.

[56] Ott M, Edunov S, Baevski A, et al. fairseq: A fast, extensible toolkit for sequence modeling [C/OL]//Ammar W, Louis A, Mostafazadeh N. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations. Association for Computational Linguistics, 2019: 48-53. https://doi.org/10.18653/v1/n19-4009.

[57] Heafield K, Lavie A. Combining machine translation output with open sourcethe carnegie mellon multi-engine machine translation scheme [J/OL]. Prague Bull. Math. Linguistics, 2010, 93: 27-36. http://ufal.mff.cuni.cz/pbml/93/art-heafield-lavie.pdf.

[58] Liu Y, Liu P, Radev D R, et al. BRIO: bringing order to abstractive summarization [C/OL]//Muresan S, Nakov P, Villavicencio A. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. Association for Computational Linguistics, 2022: 2890-2903. https://doi.org/10.18653/v1/2022.acl-long.207.

[59] Zhao Y, Khalman M, Joshi R, et al. Calibrating sequence likelihood improves conditional language generation [J/OL]. CoRR, 2022, abs/2210.00045. https://doi.org/10.48550/arXiv.2210.00045.

[60] Vijayakumar A K, Cogswell M, Selvaraju R R, et al. Diverse beam search for improved description of complex scenes [C/OL]//McIlraith S A, Weinberger K Q. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. AAAI Press, 2018: 7371-7379. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17329.

[61] Nuesch P. Order restricted statistical inference. [M]. JSTOR, 1991.

[62] De Leeuw J, Hornik K, Mair P. Isotone optimization in r: pool-adjacent-violators algorithm (pava) and active set methods [J]. Journal of statistical software, 2009, 32(5): 1-24.

[63] Huang F, Tao T, Zhou H, et al. On the learning of non-autoregressive transformers [C/OL]//Chaudhuri K, Jegelka S, Song L, et al. Proceedings of Machine Learning Research: volume 162 International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA. PMLR, 2022: 9356-9376. https://proceedings.mlr.press/v162/huang22k.html.

[64] Popovic M. chrf: character n-gram f-score for automatic MT evaluation [C/OL]//Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal. The Association for Computer Linguistics, 2015: 392-395. https://doi.org/10.18653/v1/w15-3049.

[65] Jakob W, Rhinelander J, Moldovan D. pybind11 – seamless operability between c++11 and python [Z]. 2017.

[66] Akoury N, Krishna K, Iyyer M. Syntactically supervised transformers for faster neural machine translation [C/OL]//Korhonen A, Traum D R, Màrquez L. Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers. Association for Computational Linguistics, 2019: 1269-1281. https://doi.org/10.18653/v1/p19-1122.

[67] Xiang J, Li H, Liu Y, et al. Investigating data variance in evaluations of automatic machine translation metrics [C/OL]//Muresan S, Nakov P, Villavicencio A. Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022. Association for Computational Linguistics, 2022: 150-157. https://doi.org/10.18653/v1/2022.findings-acl.14.

[68] Shen S, Cheng Y, He Z, et al. Minimum risk training for neural machine translation [C/OL]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics, 2016. https://doi.org/10.18653/v1/p16-1159.

[69] Holtzman A, Buys J, Du L, et al. The curious case of neural text degeneration [C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. https://openreview.net/forum?id=rygGQyrFvH.

[70] Shen T, Ott M, Auli M, et al. Mixture models for diverse machine translation: Tricks of the trade [C/OL]//Chaudhuri K, Salakhutdinov R. Proceedings of Machine Learning Research: volume 97 Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. PMLR, 2019: 5719-5728. http://proceedings.mlr.press/v97/shen19c.html.

[71] Zhang T, Kishore V, Wu F, et al. Bertscore: Evaluating text generation with BERT [C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. https://openreview.net/forum?id=SkeHuCVFDr.

[72] Thompson B, Post M. Automatic machine translation evaluation in many languages via zero-shot paraphrasing [C/OL]//Webber B, Cohn T, He Y, et al. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. Association for Computational Linguistics, 2020: 90-121. https://doi.org/10.18653/v1/2020.emnlp-main.8.

[73] Sellam T, Das D, Parikh A P. BLEURT: learning robust metrics for text generation [C/OL]// Jurafsky D, Chai J, Schluter N, et al. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. Association for Computational Linguistics, 2020: 7881-7892. https://doi.org/10.18653/v1/2020.acl-main.704.

[74] Rei R, Stewart C, Farinha A C, et al. COMET: A neural framework for MT evaluation [C/OL]// Webber B, Cohn T, He Y, et al. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. Association for Computational Linguistics, 2020: 2685-2702. https://doi.org/10.18653/v1/2020.emnlp-main.213.

[75] Doddington G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics [C]//Proceedings of the second international conference on Human Language Technology Research. 2002: 138-145.

[76] Banerjee S, Lavie A. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments [C/OL]//Goldstein J, Lavie A, Lin C, et al. Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005. Association for Computational Linguistics, 2005: 65-72. https://aclanthology.org/W05-0909/.

[77] Snover M G, Dorr B J, Schwartz R M, et al. A study of translation edit rate with targeted human annotation [C/OL]//Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, AMTA 2006, Cambridge, Massachusetts, USA,

August 8-12, 2006. Association for Machine Translation in the Americas, 2006: 223-231. https://aclanthology.org/2006.amta-papers.25/.

[78] Tillmann C, Vogel S, Ney H, et al. Accelerated DP based search for statistical translation [C/OL]//Kokkinakis G, Fakotakis N, Dermatas E. Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997. ISCA, 1997. http://www.isca-speech.org/archive/eurospeech_1997/e97_2667.html.

[79] Morris A C, Maier V, Green P D. From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition [C/OL]//INTERSPEECH 2004 - ICSLP, 8th International Conference on Spoken Language Processing, Jeju Island, Korea, October 4-8, 2004. ISCA, 2004. http://www.isca-speech.org/archive/interspeech_2004/i04_2765.html.

[80] Sutton R S, Barto A G. Adaptive computation and machine learning: Reinforcement learning - an introduction [M/OL]. MIT Press, 1998. https://www.worldcat.org/oclc/37293240.

[81] Katehakis M N, Jr. A F V. The multi-armed bandit problem: Decomposition and computation [J/OL]. Math. Oper. Res., 1987, 12(2): 262-268. https://doi.org/10.1287/moor.12.2.262.

附录

致 谢

转眼之间，三年的硕士求学生涯马上就要结束，我也将结束在北京的四年科研之旅，开启新的征程。值此毕业之际，我要特别感谢所有帮助和支持过我的人，包括一直以来全力支持我的家人们、悉心指导我的老师们、实验室里无私帮助我的师兄师姐师弟师妹们、以及陪伴我走过这三年时光的同学们。

首先我要由衷地感谢我的导师冯洋老师。感谢冯洋老师录取我进入研究组，让我有机会开启一段受益匪浅的求学经历。冯老师有着自然语言处理渊博深厚的学识、勤奋认真的科研态度；同时冯老师也有着高超的管理艺术，制定了井井有条的规章制度，营造了浓厚的学术氛围和优越舒适的科研环境，这些对我都是很大的激励。特别的，冯老师对我在工作中出现的缺点和不足给予了宽容和谅解，对我在学习和生活上给予了真心的帮助，对此我铭记在心。同时，也感谢实验室的各位老师对我在学习和生活上的帮助：感谢实验室的秘书刘琳和程一对科研工作提供的细致周到的支持，感谢管理计算资源的刘宇老师的默默付出。

除此以外，我还要感谢这四年来对我科研提供帮助的师兄师姐们，感谢丁春发、杨郑鑫、谷舒豪、王树根、邵晨泽、李绩成、李泽康、郭登级师兄和刘舒曼、申磊、欧蛟、谢婉莹师姐，你们的帮助和支持让我受益匪浅。同时我也要感谢所认识的师弟师妹们，他们是房庆凯、马铮睿、刘龙祥、黄浪林、桂尚彤、张珂豪、鄢子文、周奘、郭雯钰、雨田、卜梦钰、杨哲、郭守涛师弟和赵彤钰、卫李赋凌师妹。特别的，我还要感谢同级的张绍磊、张倬诚和田畅同学，他们展现出的过人的才华、执着的科研态度和积极进取的生活态度给我给我留下了深刻印象。此外，我也要感谢在有关项目中并肩作战的同学们，面对时间紧张和疫情冲击的双重考验，大家克服了难以想象的困难，这是我始终铭记并且难以忘怀的。

北京是一个充满生机活力的城市，春花秋月，夏日冬雪，我有幸能在这里经历四年的风风雨雨，见证从个人到研究组经历的一系列变化。在此，我要感谢中国科学院和计算技术研究所提供的奖助学金资助；感谢计算所提供的宽敞的办公环境，以及计算所所级科研支撑平台及技术服务团队在自然语言处理方面的支持；特别的，我要感谢北京这座城市，能够让我认识优秀的诸位。

此外，这四年在外求学，我要特别感谢我的家人们，无论经历过多少的困难，距离有多远，你们都是我的坚强后盾。

最后，感谢在百忙之中评阅论文并给出意见的各位老师，也衷心的祝愿大家后面的日子能够平安喜乐、健康幸福。

2023 年 6 月

作者简历及攻读学位期间发表的学术论文与其他相关学术成果

作者简历:

2016 年 9 月——2020 年 6 月，在电子科技大学计算机科学与工程学院获得学士学位。

2020 年 9 月——2023 年 6 月，在中国科学院计算技术研究所攻读硕士学位。

已发表（或正式接受）的学术论文:

(1) **Xuanfu Wu**, Chenze Shao, Yang Feng. Generating diverse translation from model distribution with dropout. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020. Association for Computational Linguistics, 2020: 1088-1097.

(2) Chenze Shao, **Xuanfu Wu**, Yang Feng. One reference is not enough: Diverse distillation with reference selection for non-autoregressive translation. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022. Association for Computational Linguistics, 2022: 3779-3791.

(3) Jicheng Li, Pengzhi Gao, **Xuanfu Wu**, Yang Feng, Zhongjun He, Hua Wu, Haifeng Wang. Mixup decoding for diverse machine translation. Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021. Association for Computational Linguistics, 2021: 312-320.

参加的研究项目及获奖情况:

(1) 百度校企合作项目——深度神经机器翻译关键技术研究

(2) 国家重点研发计划——基于神经网络的汉泰机器翻译研究（2017YFE0192900）

(3) 2022 年中国机器翻译大会汉语到泰语测评任务第一名