中国科学院大学

University of Chinese Academy of Sciences

硕士学位论文

神经机器翻译中语义对齐与曝光偏差问题研究

作者姓名：王树根

指导教师：冯洋　副研究员

中国科学院计算技术研究所

学位类别：工学硕士

学科专业：计算机软件与理论

培养单位：中国科学院计算技术研究所

2020 年 6 月

Research on Semantic Alignment and Exposure Bias in Neural Machine Translation

A thesis submitted to the

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Sciences

in Computer Software and Theory

By

Shugen Wang

Supervisor: Associate Professor Yang Feng

Institute of Computing Technology, Chinese Academy of Sciences

June, 2020

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。尽我所知，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已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，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保存和使用学位论文的规定，即中国科学院大学有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、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

基于深度学习的神经机器翻译方法不需要人工设置的特征，以一种端到端的方式进行训练，由于分布式表示缓解了数据稀疏性的问题，生成的译文更加流畅。神经机器翻译方法提出伊始，由于模型结构简单等特点，受到研究人员的青睐，继而在多种语言对上的翻译质量自动评价指标 BLEU 值均显著超过了传统的方法，因此迅速取代了统计机器翻译方法，成为了当前机器翻译研究的主流范式。

尽管神经机器翻译取得了巨大的成功，其机器翻译的本质及基于深度学习的特性使得这种方法具有持续改进的空间。本文主要针对神经机器翻译中的语义保持性问题和曝光偏差问题展开研究工作，分别提出融入显式的语义对齐过程、从曝光偏差问题源开始进行隐状态的融合来逐渐磨合训练过程和测试阶段上下文的一致性来改善模型提高翻译质量，并进一步提出评估方法对曝光偏差问题是否有效并给出定性结论的方案。本文的详细研究内容如下：

1. **神经机器翻译中显式语义对齐问题研究** 现有的翻译框架在训练过程中通常只关注优化词级别的匹配损失，而忽略了译文意思是否与原文一致，即语义是否已对齐，这在以语义为转换介质的翻译应用中是存在问题的。本文通过引入显式的语义对齐过程来使模型更好地纠正由语义差异引起的翻译错误，在训练过程中对源端和目标端语义进行约束以防止模型在无语义约束的情况下拟合数据，从而提高翻译性能。本文基于提出的一种能够解释现有一个句子对应多种翻译现象的句子语义空间概念模型 S3CM，设计了对齐度量并引入基于 n 元文法的语义抽取器、语义映射网络和可广播集成网络组件来建立多对多的语义对齐框架 SAMT，在解决了训练过程中的语义坍缩问题后提升了机器翻译性能。

2. **基于状态融合和输出矫正缓解曝光偏差问题的研究** 在训练过程中，为了预测下一个 Token，模型接受源端及其对应的完全正确的部分译文作为上下文。然而，在测试阶段，模型生成下一个 Token 的上下文与训练过程有差异，完全正确的部分译文被由模型从头开始生成的部分句子替代。由于模型在训练时没有接触过存在错误的环境，这种差异将引发错误累积，从而导致模型性能退化。本文提出引入隐状态级别纠偏过程对引发曝光偏差问题的源头的内部隐状态进行

融合来帮助已收敛的模型减轻其对于解码上下文的敏感程度。进一步地，我们利用普通解码器每层输出来构建辅助监督信号约束噪声解码器的相应输出，并且根据输出与问题源的网络深度区别地设置约束强度，称之为逐层输出矫正。

3. **基于 n 元文法匹配精度的曝光偏差性能评估方案** 现有的这方面的工作通常通过观测到机器翻译性能的提升来说明减轻了曝光偏差问题，然而曝光偏差问题只是机器翻译性能下降的充分条件。如果无法很好的评估新方法对改进原问题是否有效，将对原问题的研究产生掣肘。因此我们提出进一步探索如何更好地评估模型抗曝光偏差的能力的方法。具体地，鉴于曝光偏差问题源于训练过程中和解码阶段模型预测下一个词所基于的上下文不一致，我们提出先令模型运行在一种介于训练和测试之间的环境之上，然后收集运行结果计算出 n 元文法匹配精确度进行对比的过程来定性评估模型在抗曝光偏差能力方面的性能改进。

关键词： 语义对齐；曝光偏差；内部状态融合；逐层输出矫正；曝光偏差评估

Abstract

Neural machine translation (NMT) method based on deep learning does not require features set manually, and its models are trained in an end-to-end way. As distributed representation alleviates the data sparsity problem, NMT would generate more fluent translation. At the beginning of NMT method, as it is characterized by simple model structure and other characteristics, NMT was favored by researchers. Then, NMT's BLEU, the automatic evaluation index of translation quality, significantly outperforms traditional methods in multiple language pairs. Therefore, it quickly replaced statistical method and became the mainstream paradigm of current machine translation research.

Although NMT has achieved great success, its nature of machine translation and characteristics based on deep learning make this method have room for continuous improvement. This thesis mainly focused on semantic retention and exposure bias in NMT to carry out research. To improve the model and translation quality, this thesis proposed to integrate explicit semantic alignment process, and to integrate hidden state starting from the problem source of exposure bias for gradual running-in training process and context consistency in testing phase, respectively. Besides, this thesis further proposed whether the evaluation solution would be effective for exposure bias problem and gave the scheme of qualitative conclusions. The detailed contributions of this research are as follows:

1. **Research on Confronting Explicit Semantic Alignment in Neural Machine Translation** The existing NMT frameworks usually only focus on optimizing word-level matching loss in the training process, but ignore whether the meaning of translation is consistent with its original, i.e., whether the semantics are aligned, which exists problems in the application of translation with semantics as the conversion medium. This thesis introduced an explicit semantic alignment process to make model better correct translation errors caused by semantic differences and in the training process, the semantics of the source end and target end are constrained to prevent the model from fitting data without semantics constraints, thus improving translation performance. Based on

the proposed sentence semantic space conceptual model (S3CM), which can explain the translation phenomena in which one sentence corresponds to semantic alignment in multiple, this thesis designed alignment metric criterion and introduced n-gram-based semantic extractor, semantic mapping network and integrated network components for broadcasting to establish a many-to-many semantic alignment framework, SAMT, which improves the performance of machine translation after solving the semantic collapse in the training process.

2. **Research on Addressing Exposure Bias with Internal States Fusion and Layer-wise Output Rectification in Neural Machine Translation** In the training process, to predict the next token, the model accepts the source as well as its corresponding absolutely correct partial translation as the context. However, in the testing phase, the context of the next token generated by model is different from that in the training process, with the absolutely correct partial translation being replaced by the partial sentences generated from the beginning of the model. Since the model has not been exposed to the environment with errors during training, this discrepancy will lead to error accumulation, which would lead to the degradation of model performance. In this thesis, the hidden state-level correction process was introduced to fuse internal hidden states of the source causing exposure bias to help the converged model reduce its sensitivity to the decoding context. Furthermore, we use the output of each layer of the general decoder to build an auxiliary supervision signal for constraining the corresponding output of the noise decoder, and set the constraint intensity differently according to the network depth between the output and the problem source, which is called layer-wise output rectification.

3. **Evaluation Solution to Exposure Bias Performance Based on n-gram Matching Precision** The existing work in this area shows that the exposure bias problem has been alleviated usually by observing the improvement of machine translation performance. However, the exposure bias problem is only a sufficient condition for the degradation of machine translation performance. If we still can not evaluate well the effectiveness of the new method for improving the original problem, the research on such a problem will be limited. Therefore, we propose a method to further explore how

to better evaluate the model's capability of anti-exposure bias. Specifically, since the exposure bias problem originates from the inconsistent context on which the model predicting the next word is based in the training process and the decoding stage, we propose a process that lets the model run in an environment between training and testing first, and then collect the running results to calculate n-gram matching precision for comparing to qualitatively evaluate the model's performance improvement on the anti-exposure bias.

Keywords: Semantic Alignment; Exposure Bias; Internal States Fusion; Layer-wise Output Rectification; Evaluation on Exposure Bias

目 录

图形列表

表格列表

第 1 章　引言

1.1　研究背景及意义

翻译是借助于语义从一种自然语言转换为另一种自然语言的任务，而机器翻译研究如何利用计算机来完成不同语言间的翻译 (宗成庆, 2008)。

随着国际化程度的不断提高，传统的依赖掌握多种语言的专家进行人工翻译的方式已经无法满足当前日益增长的跨语言交流需求。研究人员在上世纪 40 年代伊始便提出了机器翻译的概念，拟借助计算机来完成两种语言的自动翻译过程。继而，基于规则和基于中间语言的转换翻译方法、基于记忆的翻译方法、基于实例的翻译方法以及基于统计的方法不断被提出 (宗成庆, 2008)、完善来提升机器翻译的性能，至此机器翻译达到了实用的水准。近年来，随着计算机能力的迅速提升以及数据量的快速增长，基于深度学习的方法被应用到了自然语言处理领域，使得机器翻译这种经典的序列到序列应用 (Sutskever 等, 2014; Cho 等, 2014b) 的性能得到了进一步的提升。

与前面提到的机器翻译方法相比，基于深度学习的神经机器翻译不需要人工设置的特征，以一种端到端的方式进行训练，且由于分布式表示缓解了数据的稀疏性，生成的译文更流畅。神经机器翻译模型通常基于编码器-解码器框架 (Cho 等, 2014a)，其中编码器将源端（称为源端句子、源端序列）编码成实值的向量表示，称之为编码器输出，之后解码器根据编码器输出和已生成的部分目标端（目标端句子、目标端序列）依次地解码出完整地目标端序列[1]。由于解码时搜索空间巨大，该框架一般采用基于集束搜索（BeamSearch）或基于采样（Sampling）的方法进行解码。在编码器-解码器框架下，三种主要的神经机器翻译模型架构分别为：RNNSearch (Bahdanau 等, 2015)，ConvS2S (Gehring 等, 2017), Transformer (Vaswani 等, 2017)。RNNSearch 使用基于门控循环单元（Gated Recurrent Unit，GRU）的循环神经网络结构（Recurrent Neural Network，RNN）对源端进行双向编码，并将由当前位置相应的正反向得到的编码拼接起来作为当前位置 Token 的表示，解码时，解码器根据解码出的前一个 Token、解码器当前

[1]由于神经机器翻译是一种序列到序列的应用，且序列由 Token（词的某种形式，可能不具备语言学意义）构成，不失一般性地，本文使用源端序列序列替代源端或源端句子，称目标端或目标句子为目标端序列，使用 Token 嵌入指代词嵌入（word embedding）。

的状态以及通过多层感知机方式（Multi-Layer Perceptron，MLP）动态计算出的上下文向量（context vector）来预测下一个词，其中这种动态计算上下文的方式被称为注意力机制（Attention Mechanism），该模型中变长的双向源端表示和注意力机制缓解了长距离依赖问题。由于 RNN 固有的自回归特性导致模型无法在序列内进行并行训练，ConvS2S 直接摒弃了这种循环单元，整体基于卷积神经网络结构（Convolutional Neural Network，CNN），通过可学习的位置编码和加深网络深度来弥补时序信息缺失的弱点和增强对底层信息的感受野（receptive field），该方法增强了模型训练的可并行性并进一步提高了机器翻译性能。紧接着，完全基于注意力（Attention）机制的 Transformer 模型在进一步简化了模型结构的同时增强了模型的可并行性，并且取得了更优异的性能。

总而言之，神经机器翻译在多个数据集和多种语言对上都表现出了巨大的优势，并迅速取代了统计机器翻译成为了学术界研究和工业界应用的主流范式。尽管如此，神经机器翻译中存在着一些不足之处，如得到的翻译内容与原文意思不一致、模型在训练和测试时接收的数据不完全一致等问题，这些弱点将影响着翻译的质量，因此亟待进一步研究。

1.2 国内外研究现状分析

作为自然语言处理领域一种能够快速落地且具有实用价值的应用，神经机器翻译这种典型的序列到序列的试验台近年来的受到了持续的关注。本节将从神经机器翻译的基本理论模型、当前主流的模型架构、评价神经机器翻译性能的主要指标等几个方面来分析国内外在该研究方向上的发展现状，并介绍前沿的研究方向。

1.2.1 神经机器翻译模型

给定一个样本 $(\mathbf{x}, \mathbf{y})$，$\mathbf{x} = \{x_1, \cdots, x_{T_x}\}, \mathbf{y} = \{y_1, \cdots, y_{T_y}\}$，神经机器翻译直接通过如下公式建模在给定源端句子情况下的目标端句子的概率：

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T_y} P(y_t|\mathbf{y}_{<t}, \mathbf{x}; \Theta)$$

其中 $\mathbf{x}$、$\mathbf{y}$ 分别为源端序列和目标端序列，它们是由对应词汇表索引构成的整数索引向量，T_x 和 T_y 分别对应它们的长度（含 Token 的数量），Θ 为模型参数。这种类似于神经语言模型的工作最早可追溯到 Bengio 等 (2001)，因此训练通常被

定义为在给定的训练数据集上对模型参数 Θ 进行优化来最大化训练数据的似然，其中一般使用基于随机梯度下降的方法来对参数进行更新。

1.2.2　主流的神经机器翻译架构

发展至今，神经机器翻译日臻成熟，在模块设置上，注意力机制成了事实上的标准配置；在 Token 粒度选择上更偏向于使用子词单元 (Wu 等, 2016; Sennrich 等, 2016)。研究人员先后提出了 3 种典型的架构：

- RNNSearch (Bahdanau 等, 2015)：基于循环神经网络门控单元、使用多层感知机注意力机制的架构；
- ConvS2S (Gehring 等, 2017)：引入可学习位置编码并具备注意力机制的完全基于卷积网络的架构；
- Transformer (Vaswani 等, 2017)：引入相对位置编码的完全基于（多头）注意力机制的架构。

这 3 种架构在捕获序列的长距离依赖的能力和可并行性训练方面呈向好的态势，因此本文在模型改进方面的研究工作均基于当前最先进（State-Of-The-Art，SOTA）的 Transformer 架构。

Transformer 整体架构如图 1.1所示[2]。该架构由均堆叠了 N 个基本块的编码器和解码器组成。对于编码器来说（图 1.1左侧组件），每个基本块均由多头注意力（Multi-Head Attention）子层和前馈神经网络（Feed Forward）子层构成，这些子层均对输出进行了残差（Add）和层正规化（Norm）操作。对于解码器，其整体结构与编码器大致相同，只是在多头注意力子层和前馈神经网络子层之间引入了用于与源端进行交互的编码器-解码器多头注意力子层。在训练过程中，为了防止探测到未来的信息，解码器的多头注意力子层（Masked Multi-Head Attention）的未来权重部分被屏蔽。注意到，不像循环神经网络中存在自回归的特性使得任意时间步的状态均包含了历史信息，Transformer 的并行特性导致获取的表示缺少时序信息，因此引入了位置编码（Positional Encoding）来弥补这一缺陷。具体而言，Transformer 利用三角函数的周期性来捕获词与词之间的相对位置信息：

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

[2]对于另外两种架构的细节请直接查看原论文。

其中 pos 为当前 Token 相对于序列起始的位置，i 为对应的 Token 嵌入的维度，d_{model} 表示模型的维度。Token 嵌入加上相应的位置编码共同构成了网络的输入。

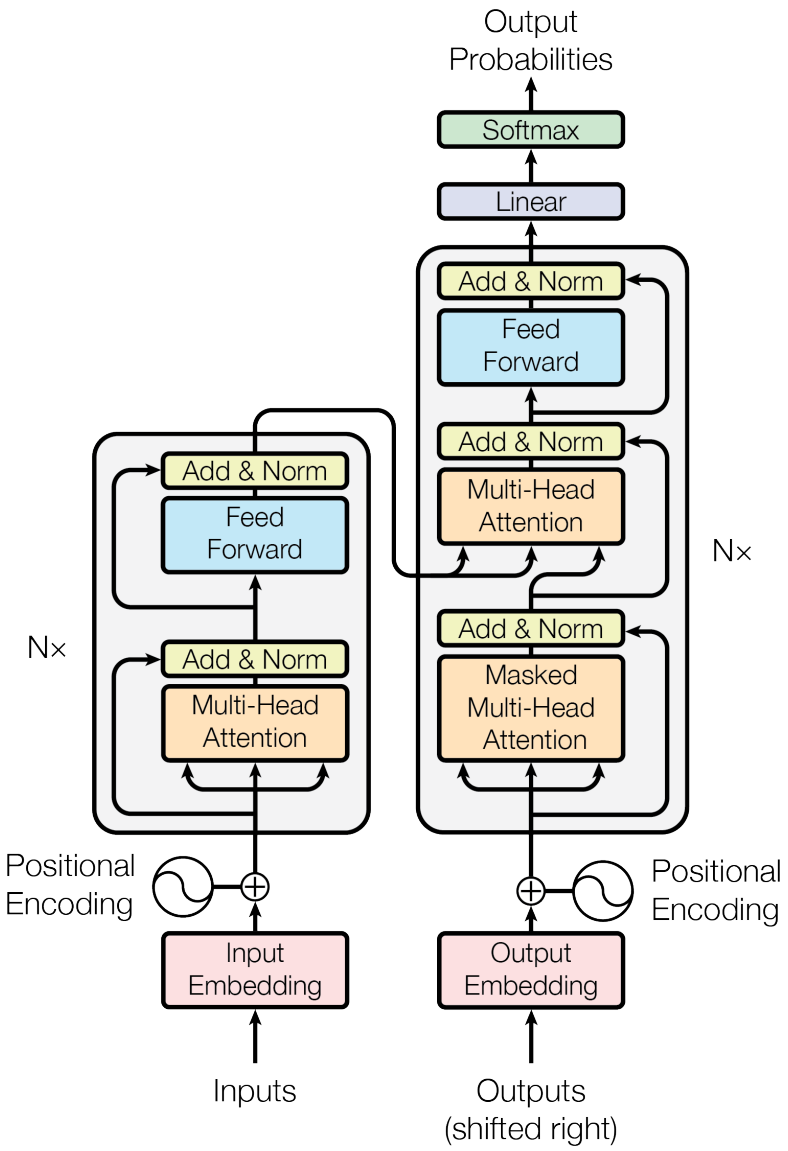

图 1.1 Transformer 架构图, 引自Vaswani 等 (2017)

Figure 1.1 Transformer Architecture, cited from Vaswani 等 (2017)

Transformer 的另一个重要的模块多头注意力如图 1.2所示：

对于给定的查询 Q，键 K，值 V，多头注意力模块首先通过不同的线性变换将它们映射到多头（图 1.2中的 h 个）子空间，然后分别对映射后的数据计算注意力（Attention），再把这些得到的注意力进行拼接，最后通过对拼接后的注

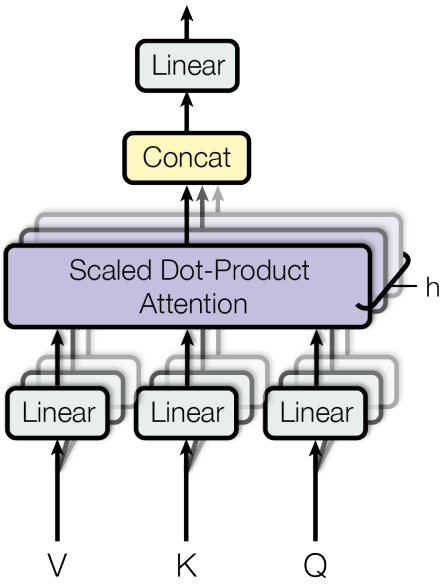

图 1.2 多头注意力模块, 引自Vaswani 等 (2017)

Figure 1.2 Multi-Headd Attention Module, cited from Vaswani 等 (2017)

意力重新映射得到最终的输出。整个过程可以通过公式形式化为：

$$q = Q\mathbf{W_Q} = [q_1; q_2; \cdots; q_h]$$

$$k = K\mathbf{W_K} = [k_1; k_2; \cdots; k_h]$$

$$v = V\mathbf{W_V} = [v_1; v_2; \cdots; v_h]$$

$$o_i = \mathbf{Attention}(q_i, k_i, v_i)$$

$$o = [o_1; o_2; \cdots; o_H]\mathbf{W_O}$$

其中 $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}, \mathbf{W_O} \in \mathbb{R}^{d_m \times d_{in}}$ 均为线性变换的权重矩阵，d_m 和 d_{in} 分别代表模型维度和输入的 Token 嵌入维度，h 是头的个数。**Attention** 为放缩点积注意力（Scaled Dot-Product Attention），其计算方式如下：

$$\mathbf{Attention}(Q, K, V) = \text{softmax}(\frac{Q \cdot K^T}{\sqrt{d_m}})V$$

Transformer 模型架构对学习率敏感，使用 Adam (Kingma 和 Ba, 2015) 优化器时，学习率调整策略按照如下公式设置：

$$lrate = d_{model}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

其中 $warmup\_steps$ 被设置为 4000。

1.2.3 神经机器翻译性能评价指标

为了评估一个方法是否有效，人们通常会引入性能评估指标。在机器翻译中，性能指的是源端句子通过机器翻译系统得到的译文质量的好坏。主观上，人们通过流畅度、充分性及语义保持性来评估译文的质量。事实上，为了降低人工评价的成本，通常使用基于 n 元文法匹配的混合精确度指标 BLEU (Papineni 等, 2002) 来自动评估译文质量，该指标值介于 0–1 之间，且数值越大，则表明系统生成的译文质量越高，BLEU 值通常以百分数形式在文献中出现。BLEU 值的计算方式如下：

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

其中 p_n 和 w_n 表示 n 阶 n 元文法匹配精确度和对应的权重，BP 为长度惩罚因子，计算方式为：

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

c、r 分别为系统译文长度及其余所有参考译文长度进行衡量得到的长度。本文评估时使用 *multi-bleu.perl*[3] 脚本。

1.2.4 神经机器翻译研究方向

当前，神经机器翻译的探究大致分为以下几个方向：

先验知识融合

神经机器翻译模型的训练完全基于数据驱动的模式，即自动从大量的数据中学习出数据的特征。由于自然语言本身具有一定规律（语法结构等），有效的将这些先验知识融合到模型或者作为辅助的监督信号将帮助模型快速地学习到额外的能力，从而使模型生成更加的流畅、充分和准确的译文。例如， Chen 等 (2018a) 等提出使用依存树（Dependency Tree）中词间的距离替代序列中的距离来指导局部注意力 (Luong 等, 2015)。 Dinu 等 (2019) 提出一种在运行时将自定义术语注入神经机器翻译的方法。 Zhu 等 (2020) 提出一种先使用 BERT 提取输入序列的表示，然后通过注意力机制将表示与 NMT 模型的编码器和解码器的每一层融合的 BERT 融合模型的算法。

[3] https://raw.githubusercontent.com/moses-smt/mosesdecoder/master/scripts/generic/multi-bleu.perl

训练准则

神经机器翻译通常使用基于极大似然估计（MLE）的损失函数作为训练目标，这种基于词匹配的训练准则能够很好地拟合训练数据，但是忽略了词与上下文之间的关系。例如， Shen 等 (2016) 等提出使用最小风险训练原则作为优化目标进行训练。 Wieting 等 (2019) 基于语义相似性方面的最新工作，引入了一种用于优化 NMT 系统的替代奖励函数。

低资源语言翻译

神经机器翻译模型需要在训练数据量达到一定规模的时候才能达到可实用的水准，但由于双语平行语料库构建成本大，很多小语种之间缺乏大规模的平行语料。 Pourdamghani 等 (2018) 等提出在单语数据上通过提高语义相似的词的相似度来改善词的对齐质量。 Gu 等 (2018b) 等提出利用迁移学习方法将多种源语言中的词汇和句子级表示共享到一种目标语言中，词汇部分通过通用词汇表示来共享，以支持多语言词级共享。 Murthy 等 (2019) 等提出在迁移学习的基础上重新排序辅助语言句子以匹配源语言的词序来训练父模型。

鲁棒性

神经机器翻译对输入敏感，微小的扰动将严重的影响端到端间的中间表示，进而导致性能下降。为了获得稳健的翻译能力，研究人员尝试通过对数据增加噪音、修改模型架构或训练准则等方式来增强对该方面的关注度。 Cheng 等 (2018) 等提出通过对抗稳定性训练来提高神经机器翻译模型的鲁棒性。 Vaibhav 等 (2019) 等提出通过在不一样的干净的数据中模拟自然产生的噪声来增强机器翻译系统的鲁棒性。

可解释性

神经机器翻译基于黑盒模式的特性使得人们仅能够观察到端到端的结果，而无法对其输出结果进行可证明或可视化的有效推断，这使得模型算法开发过程不便调试且结果不受控制，从而限制了神经机器翻译的发展。 Ding 等 (2017) 等提出使用分层相关传播 (Layer-wise Relevance Propagation，LRP) 来计算每个上下文词对基于注意的编码解码器框架中的任意隐藏状态的贡献。 Bau 等 (2019) 等根据不同的模型不需要任何昂贵的外部监督也能学习到相似的属性直觉，提出

了如何在神经机器翻译模型中发现重要神经元的无监督方法。

非自回归机器翻译

传统的自回归神经机器翻译在预测下一个词时，需要对整个翻译历史进行回顾，这种串行的解码方式一定程度上限制了翻译速度。Gu 等 (2018a) 等提出能够避免自回归特性的模型来并行产生输出，从而使推理过程中的延迟降低一个数量级。Wang 等 (2018) 等提出在在局部上减轻、全局上保持自回归特性，使得模型能够在每个时间步并行产生多个连续词。Guo 等 (2019) 将课程学习引入到 NAT 的微调中，在微调过程中设计课程，以逐步将训练从自回归生成转换为非自回归生成。Chenze Shao 和 Zhou (2019) 提出以最大程度地减少模型输出和参考语句之间的 n-gram 袋（Bag-of-Ngrams）差异的 NAT 训练方法。Shu 等 (2020) 提出一个具有连续潜在变量和确定性推理程序的潜在变量非自回归模型，与现有方法相反，使用确定性推理算法来找到目标序列，该目标序列将对数概率的下限最大化。非自回归机器翻译探索以非完全串行解码的方式进行生成的可能性，当前该方法相对于自回归机器翻译来讲存在着较大的性能损失。

篇章翻译

在翻译中，将文档作为一个整体考虑有助于解决歧义和前后及指代不一致等问题。Wang 等 (2017) 等提出了一种跨句子语境感知方法，并研究了历史语境信息对神经机器翻译性能的影响。Maruf 和 Haffari (2018) 提出使用记忆网络 (Weston 等, 2015) 同时考虑源文档和目标文档上下文来改善篇章翻译模型。Miculicich 等 (2018) 等提出以结构化和动态的方式捕捉上下文的层次化注意力模型。Voita 等 (2019) 提出一种单语的 DocRepair 模型来纠正句子级翻译之间的不一致，该模型对一系列句子级别的翻译执行自动后编辑，从而在彼此上下文中优化句子的翻译。Li 等 (2020) 提出一种显式句子压缩方法来增强 NMT 的源句子表示的方法。

语音翻译

传统的神经机器翻译以文本作为输入并输出文本，然而除了文字材料外，另一种常见的翻译场景是跨语言对话，输入为音频。一种方法可通过自动语音识别、机器翻译、语音合成这种流水线的形式来完成，另一种是端到端（直接语音到语音）的尝试。Duong 等 (2016) 等使用语音-字幕这种双语数据来进行语音-词

对齐任务。 Salesky 等 (2019) 等使用序列到序列模型将嘈杂、不流畅的语音翻译成不流畅的文本，并使用收集的 Fisher-Spanish 数据集的“拷贝编辑”参考来消除影响, 这种方式能够直接生成流畅的翻译。

同声传译

为了降低时延，达到实时翻译效果，研究人员进一步提出了同声传译的概念 (Cho 和 Esipova, 2016)，这种情景下不需要等待整个待翻译的句子全部输入便能逐步的得到译文。 Dalvi 等 (2018) 等通过修改神经机器翻译解码器使其与动态构建的编码器和注意力一起工作来解决同时翻译的问题。 Alinejad 等 (2018) 等提出了一种新的通用预测动作来预测输入中未来的词以提高质量并最小化同声翻译中的延迟，并使用强化学习的方法进行训练。 Zheng 等 (2019a) 提出一个从并行文本生成的先验 READ/WRITE 序列中学习自适应策略简单的监督学习框架。为了应对集束搜索在同声传译的挑战， Zheng 等 (2019b) 提出了一种推测性波束搜索算法将幻觉化到未来的几步，以便从目标语言模型中隐含地受益，从而做出更准确的决策。

1.3　主要研究目标

神经机器翻译尽管在 BLEU 值等自动评价的性能指标上显著超过了传统的机器翻译方法，甚至在限定领域的人工评价中可以与人工翻译结果相媲美，但仍然面临着诸多挑战。本文针对源自翻译的本质及监督式深度学习模型固有及延伸的问题进行研究，即对神经机器翻译中的语义保持性问题和训练、测试阶段上下文不一致的曝光偏差问题进行研究，接下来将对问题做出简要的描述和举例说明。

1.3.1　神经机器翻译中的语义保持性问题

传统的训练方法使用基于最大似然估计的优化目标，这种仅关注词级别匹配损失的训练准则忽略了系统译文与原文意义是否保持一致，这使得译文即使在没有漏翻（under-translation）和过翻（over-translation）的情形下，可能依然不理想。例如，我们在基线系统的译文中找到了如下案例: 在表 1.1案例中，系统译文与 **#2** 参考译文完全一致，根据参考译文计算得到的 BLEU 值达到了该评价指标的上界 1.0。但是原文中的加粗的“经历”在前三个参考译文中都没有被显式的

翻译原文	这 是 今年 入 春 以来 朝鲜 **经历** 的 第一 场 沙尘暴 。
参考译文	**#1** this is the first sandstorm to hit north korea since this spring . **#2** this is the first sandstorm in north korea since spring this year . **#3** this is the first spell of sandstorms that hit dprk since this spring . **#4** this is the first sandstorm north korea **experiences** after spring comes this year .
基线译文	this is the first sandstorm in north korea since spring this year .

表 1.1 基线系统翻译结果破坏语义保持性的示例

Table 1.1 Example of Baseline Translation where Semantics are Not Maintained

翻译出来，主观的来说，对原文中的 “经历” 进行显式翻译将使得译文更加地忠实原文。

1.3.2 神经机器翻译中的曝光偏差问题

神经机器翻译中预测下一个词的所基于的上下文在训练过程中和测试阶段不一致。在训练过程中，为了预测下一个词，模型能够观察到源端句子及已正确生成的部分序列。然而在测试阶段，由于没有参照，一般使用完全由模型生成的部分句子来代替已正确生成的部分序列。由于在训练的过程模型仅在完全没有错误的环境下学习，而没有被显式的教导如何应对已生成的部分句子中存在错误的情况，因此模型基于这种可能存在错误的部分句子来预测下一个词会导致错误的进一步累积，最终造成模型解码性能下降。

翻译原文	中国 星期五 允许 他们 **取道 菲律宾** 前往 汉城 。
参考译文	**#1** on friday , china allowed them to travel to seoul **via the philippines** . **#2** on friday , china allowed them to go to seoul **via philippines** . **#3** china allowed them friday to go to seoul **through the philippines** . **#4** china allowed them to fly for seoul **through the philippines** on friday .
基线译文	china allows them to travel to seoul on friday .

表 1.2 曝光偏差问题导致基线翻译不完整的示例

Table 1.2 Example of Incomplete Baseline Translation Caused by Exposure Bias

在表 1.2的案例中，所有的参考译文均将 “允许” 译为 “allowed”，并且我们通过进一步搜索训练语料发现，这些新闻领域的句子中的 “允许” 均被译为 “allowed” 或者 “allow”。因此对于基线译文未对原文中的 “取道 菲律宾” 进行翻译

（加粗部分内容）这种现象，我们将其归因于前面词语的翻译错误导致的累计错误。

此外，由于该问题只是导致机器翻译性能下降的充分条件，从而无法评判该问题是否得到了缓解，由于无法获得更多的细节，将不利于对该问题的进一步研究，因此我们还对如何评估方法在缓解曝光偏差问题的效果方面进行了研究。

1.4　论文贡献

在全球化的背景下，日益增长的跨语言的交流需要使得机器翻译具有广阔的应用前景。基于现有工作，本文工作针对 1.3节提出的几个问题进行了研究：

- 约束语义保持性问题：提出首先建立合适的语义空间概念模型，并基于该模型在训练过程中引入显式的语义对齐过程来约束语义偏离现象。
- 缓解曝光偏差问题：提出从偏差源开始对模型进行纠正来缓解该问题，具体做法为在更细粒度的隐状态级别对有偏和无偏的状态进行融合并以接受完全正确的部分序列作为输入得到的输出来构建辅助监督信号来指导接受非完全正确的序列的输出。
- 评估曝光偏差问题：提出新的方案以期更好地评估该问题，测量并证明新方法的引入使得模型在抗曝光偏差方面的能力得到了提升。

1.4.1　神经机器翻译中的显式语义对齐问题研究

现有的翻译框架通常只关注优化词级别的匹配损失，而忽略了译文意思是否与原文一致，即语义是否已对齐，这在以语义为转换介质的翻译应用中是存在问题的。因此，本文认为引入显式的语义对齐监督可以使模型更好地纠正由语义差异引起的翻译错误，从而提高翻译性能。具体来说，针对基于最大似然估计的词级别匹配训练准则对序列中的各个词无差别考虑（忽略了序列上下文）从而导致词匹配度高而语义无法保持的问题，本文提出在训练过程中对源端和目标端语义进行约束以防止模型参数在无语义约束的情况下拟合数据。

更具体地，本文基于能够归纳现有现象的句子语义空间概念模型建立一个语义对齐框架。在这个框架中，我们使用基于 n 元文法的语义抽取器从原始 Token 嵌入中提取出句子的语义表示，这为解码器引入另一种源端信息流。为了将抽取到的语义信息融入到生成过程中，我们使用可广播集成网络将源语义表示融合到解码器中。考虑到无法保证直接提取的语义总是在相同的语义空间下或者被

投影到相同的区域，我们提出使用具有函数近似功能的语义映射网络来桥接源端语义表示和目标端表示来使它们在同一区域。有了这些基础后，我们进一步通过提出设计一种称之为语义散度的参数化度量来约束句对间的语义关系，即句对的语义对齐。

1.4.2 基于状态融合和输出矫正缓解曝光偏差问题的研究

在训练过程中，为了预测下一个 Token，模型接受源端及其对应的完全正确的部分译文作为上下文。然而，在测试阶段，模型根据不同的上下文生成下一个 Token，即没有了完全正确的部分译文，取而代之的是由模型从头开始生成的部分句子。由于模型在训练时完全没有接触过这种存在错误的环境，这种差异将引发错误累积，从而导致模型性能退化。针对预测下一个词的所基于的上下文在训练过程和测试阶段不一致这个曝光偏差问题，本文拟从引发问题的源头引入隐状态级别纠偏过程，帮助已收敛的模型减轻其对于解码上下文的敏感程度。具体地，我们提出两种在隐状态级别上减轻这种差异的方法：内部状态融合和逐层输出矫正。

对于内部状态融合方法，从一个已收敛的模型开始，我们按照正常的程序先运行一次解码器（称为普通解码器）以获得曝光偏差来源的偏置状态、每层的输出和最终预测，该解码器接受地完全正确的部分句子为输入馈送（input feeding）。然后，我们运行概念上的噪声解码器（该解码器与普通解码器进行了参数绑定），其输入馈送是存在错误的 (例如，将来自普通解码器的最终预测输出作为输入馈送)，在这个过程中，我们会先将无偏状态（本次获取的状态）与前面运行普通解码器获得的偏置状态按比例融合，然后再继续正常的解码过程。进一步地，我们利用普通解码器每层输出作为一种辅助的监督信号来约束噪声解码器的相应输出。我们根据每层偏差的程度施加不同的约束强度，越靠近偏差源的约束强度越小，称之为逐层输出矫正。

1.4.3 基于 n 元文法匹配精度的曝光偏差性能评估方案

现有的这方面的工作通常通过观测端到端的机器翻译性能提升来说明减轻了曝光偏差问题，然而这种说法不具有很强的说服力。因为曝光偏差问题只是机器翻译性能下降的充分条件，新方法对改善原问题是否有效尚未可知，如果无法很好的评估一个方法对原问题是否有效及改善程度，则对原问题的研究会有所

掣肘。即如果不能很好地测量和证明是新方法是否及如何增强了模型在抗曝光偏差方面的能力，将不利于对曝光偏差问题的进一步分析。因此我们提出探索如何更好地评估模型抗曝光偏差的能力的方法。由于曝光偏差问题源于训练过程中和解码阶段模型预测下一个词所基于的上下文不一致，我们提出通过运行在一种介于训练和测试之间的环境之上的过程来评估模型在抗曝光偏差方面的性能改进。

1.5　章节组织

本文的组织结构如下：

第 1 章介绍了神经机器翻译的背景及意义、研究现状和研究方向，阐述了本文研究针对的问题及研究必要性，并给出了本文的贡献概览及论文组织结构。

第 2 章提出了在神经机器翻译训练过程中引入显式的语义对齐过程，对训练过程施加语义对齐约束来促使模型在遵循句子语义空间概念模型 S3CM 的前提下拟合数据，建立语义对齐框架，以解决传统的词匹配级别的训练目标忽略翻译后的意思原文是否一致的问题。

第 3 章针对神经机器翻译在训练过程和测试阶段预测下一个 Token 的上下文不一致的现象（已经部分生成的目标端序列），在对问题进行扩展后，提出通过从问题源开始对已收敛的模型进行内部状态融合来提升其抗曝光偏差能力，并通过进一步构造的辅助监督信号的逐层输出矫正的方法进行了增强。

第 4 章针对曝光偏差问题出现的必然性及其严重影响模型性能的事实，但如何评估一个方法对缓解其是否有效这个悬而未决的问题，提出通过收集收敛的模型运行在上下文错误程度介于训练和测试之间且可控的的坏境下的预测结果来计算基于 n 元文法匹配精确度的方案来获得内部细节以定义衡量方法是否有效和如何得出定性结论。

第 5 章对全文内容进行总结，指出了研究存在的不足，并指明了进一步的研究方向。

第 2 章　神经机器翻译中显式语义对齐问题研究

2.1　引言

近代著名翻译家严复曾总结出过翻译的标准，即“信、达、雅”。这三点的关系由浅入深：“信”就是一定要尊重原文，一定要把原作者索要表达的意思翻译准确，不能按照瞎编胡改；“达”则是建立在“信”的基础上，进一步将原文的内涵表达出来，只要能够表达出原文的意思即可，不用斤斤计较于字数；“雅”要求一定要翻译成规范，符合正统的文体，其针对的是译文的水平，即好的译文不仅能很好的传达作者的想法能让人受益，也体现着译者的水平。现阶段机器翻译技术水平已经基本能够实现“信”和“达”的翻译要求。近年来新兴的神经机器翻译技术更是让译文表现出了优越的流畅性。若将机器翻译置于“信、达、雅”的翻译标准中，则本质上是一种通过语义进行的序列到序列的转换应用。句子的语义指的是整个句子的意思，任何翻译良好的译文都应该尊重原文或具有与原文相同的内涵。

现阶段有监督的神经机器翻译框架使用 < 原文-译文 > 对形式的样本进行训练，通常只关注如何优化基于最大似然估计的词级别匹配损失，使模型持续的拟合训练数据，而忽略了对译文意思是否与原文一致的约束，即语义是否已对齐。基于训练样本均已良好翻译的假设，本项研究认为引入显式的语义对齐监督可以使模型在保持原文和参考译文意思一致的前提下拟合训练数据，因此能够更好地纠正由语义差异引起的翻译错误，进而提高翻译性能。据我们所知，这是在神经机器翻译中显式进行语义对齐的第一次尝试。

一方面一个源端句子本身有多个语义上完全等同的翻译，如“今天天气怎么样？”既可翻译成“what is the weather like today?”也可翻译成“How is the weather today?”；另一方面译文会受到原文所在上下文及译者的文化背景等的影响，如电影中的不同字幕译本；加上诸如一词多义等歧义性现象在自然语言中普遍存在的因素，因此一个句子所对应的译文在大多数情况下不止一个。鉴于这个事实，在分布式表示的情境中[1]，我们提出一种能够对现有的一个句子可能存在多种翻

[1]当前尚未有被普遍接受的对语义的定义，对语义的进行研究的工作大致可以分为以下两类：一类是对词法进行分析的形式语义，如语义角色标注、依存句法树分析等；另一类是基于连续向量表示的分布式语义，如词嵌入应用、知识图谱中的实体关系表示等。一方面语义无法被解开单独表示，一方面语义确实又

译的事实进行解释的句子语义空间概念模型（Sentence Semantic Space Conceptual Model，S3CM）。

在本章接下来的内容中，我们将首先介绍当前已有的相关工作，并详细的介绍 S3CM 模型和基于该模型如何设计用于约束训练过程中语义对齐的度量准则。然后进一步阐述我们为实现语义对齐目标而引入的多个必要模块的动机及实现细节，由此建立起一个显式的语义对齐框架。我们也将描述在实现语义对齐目标所遇到的难点，针对异常结果的分析及解决问题所采取的解决方案。最后，我们将在不同规模的多个数据集的多个语言对上进行试验，通过分析结果表明，训练过程中，对神经机器翻译进行显式的语义对齐有助于减轻语义保持性被破坏的现象，从而提高翻译性能，并且在具有多个参考译文的数据集上的 BLEU 提升更为明显。

2.2 相关工作介绍

自神经机器翻译成为主流的机器翻译范式以来，如何引入句子级的信息来提升翻译性能的研究便层出不穷。现有的应用语义来提高机器翻译性能及语义保持性相关的工作大致可以归为以下两类：一类通过建模覆语义覆盖度来指导整句，另一类对融入了语义信息的训练过程进行不同层面的约束。

2.2.1 建模语义覆盖度的方法

Tu 等 (2016) 等针对基于注意力机制的模型倾向于忽略过去的对齐信息而经常导致过度翻译和翻译不足的问题，提出通过维护一个覆盖向量来跟踪注意力历史，使得覆盖向量被馈送到注意力模型以帮助调整对未来的关注程度，这使得模型能够考虑更多关于未翻译的源端词。Zheng 等 (2018) 等针对现有的神经机器翻译模型在解码阶段没有明确地对已经翻译的和没有翻译的进行建模的问题，进一步提出了将源信息分成已翻译的过去内容和未翻译的未来内容两部分，这两部分由两个额外的递归层建模。基于这种新机制，过去和未来的内容被应用到了注意力模型和解码器状态，这为解码器提供了已翻译和未翻译内容的知识。

存在着，如翻译这种借助于语义从一种自然语言转换为另一种自然语言的应用能够捕捉到语义，因此机器翻译亦成为了一种最流行的语义研究的试验台。

2.2.2　融入语义信息进行约束的方法

Song 等 (2019) 等研究了抽象语义表示（Abstract Meaning Representation，AMT）的作用并通过图循环网络（Graph Recurrent Network，GRN）融入了语义角色标注信息而提高了机器翻译性能。Zhang 等 (2016)（VNMT）将输入经过编码器得到的表示进行 mean-pooling 后再经过前馈网络变换并采样得到的隐变量视为语义，通过最小化 KL 散度来令其采样所服从的近似先验、后验分布相互逼近来使基于源端得到的先验语义和同时基于源端及目标端得到的后验语义相同。但 Shah 和 Barber (2018) 认为 VNMT 使用的隐变量和编码器的隐藏状态起着相同的作用。此外，在强大的自回归解码器下，VNMT 性能严重下降。 Yang 等 (2019) 提出直接最小化源端表示和目标表示之间的差异，其中源端表示和目标表示分别通过对转换后的编码器输出应用 mean-pooling 和位置目标嵌入来得到，但这种做法缺乏对语义的显式提取过程和对齐层次要求并且忽略了自然语言的歧义。

2.3　显式语义对齐

在本节中，我们将详细的介绍我们提出的句子语义空间概念模型以及基于该模型设计的语义散度（Semantic Divergence，SD）参数化度量和为实现显式语义对齐（Explicit Semantic Alignment）所引入的三大组件：用于从原始 Token 嵌入中提取句子级信息的语义抽取器（Semantic Extractor），将源端语义表示融合到解码器中的可广播集成网络（Broadcastable Integration Network）和映射目标语义表示到与源端语义表示的相同层级不可知语义空间的语义映射网络（Semantic Mapping Network）。我们将引入显式语义对齐约束的神经机器翻译模型命名为 SAMT。

2.3.1　多对多的句子语义空间概念模型

基于 2.1提到的一个句子通常有多个语义等价的翻译并且歧义在自然语言中广泛存在的事实，句子级语义对齐的基本框架应当是多对多形式的[2]。在分布式表示情境中，我们提出能够概括这种一对多现象的句子语义空间概念模型：句子的语义表示不是简单的固定维度的向量，而是一片由与中心向量相距一定角度

[2]一个原文可能有多个不同的目标译文，反之亦然。

和距离范围的一簇向量构成的空间区域。直观上的来理解，在二维空间，一个句子的语义表示可以视作为一个圆和一个角的重叠部分，如图 2.1所示。若紫色射线为中心向量，则 θ 为相对于中心向量偏离的角度，r 为相对中心向量偏离的距离，两条灰色射线张成的角（任意一条灰色射线与中心向量形成的角度为 θ）与虚线构成的圆（圆以中心向量的端点为圆心，r 为半径）重叠部分的向量即为句子的语义空间区域。在三维空间，一个句子的语义表示可理解成一个球体和一个无底圆锥体的重叠部分。

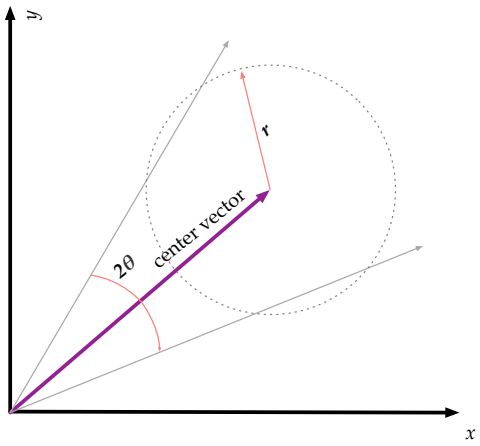

图 2.1 二维空间下的句子语义空间概念模型

Figure 2.1 Sentence Semantic Space Conceptual Model in Two-Dimensional Space

2.3.2 语义散度：句子语义对齐准则

根据 2.3.1节介绍的句子语义空间概念模型，我们在设计衡量两个语义向量彼此接近程度的语义散度（Semantic Divergence）对齐准则时，同时考虑角度 θ 和半径 r，并强烈偏重于角度 θ。对于任意的源端-目标端语义表示向量对 $(\mathbf{s}, \mathbf{t})$，它们的语义散度 $\mathcal{L}_{\text{SD}}(\mathbf{s}, \mathbf{t})$ 计算过程如下：

$$sim = \frac{\mathbf{s} \cdot \mathbf{t}}{\|\mathbf{s}\|_2 \|\mathbf{t}\|_2}$$

$$dis = \|\mathbf{s} - \mathbf{t}\|_2$$

$$\theta = \arccos(sim)$$

$$\mathcal{L}_{\text{SD}}(\mathbf{s}, \mathbf{t}) = S(\upsilon) \cdot dis$$

其中 $S(\upsilon)$ 是用于增大角度 θ 的影响力的连续放缩函数，υ 是对 θ 进行某种初等变换后的变量。$S(\upsilon)$ 应该具备以下属性：

- 随 θ 单调递增；
- 随 θ 迅速变化（尤其是当 $\theta > \frac{\pi}{2}$ 时）；
- 能够获得很大的值，以使语义散度在整个训练准则中变得重要。

我们提出了两种实现该函数的方法。

在第一种称之为 SAMT#sim 的方法中，我们直接将由两个语义表示向量计算得到的相似性值视为一个变量。然而，这样得到的变量与 θ 具有相反的单调性，并且我们也很难找到一个能够显著地放缩一个定义域在 [-1，1] 上的变量的初等函数。因此我们先对该变量进行变换，然后再用指数函数对其进行放缩。计算过程如下:

$$var = -sim + 2.0$$

$$S(sim) = e^{var \cdot 6.0} + b$$

对于变量 sim，我们首先对其取相反数来获得与 $S(v)$ 一致的单调性，然后在此基础上通过加 2.0 来完成对该定义域向右移动的操作：[-1,1] $\Rightarrow$ [1,3]。进一步地，我们对变化后的变量 var 乘上 6.0，以使函数值能够对自变量微小的变化敏感，并且能获得较大的函数值。偏移量 b 用于调整初始值，在所有的实验中，我们将其设置为-1.0。

直接由角度 θ 来决定放缩函数的值似乎比使用相似性 sim 更为直观。从这个角度出发，我们提出了第二种方法，记作 SAMT#ang。众所周知，两个矢量之间的角度 θ 的范围为 [0,π]。根据2.3.1节提出的句子语义空间概念模型，我们能够接受两个语义向量的角度在 (0，$\frac{\pi}{2}$) 范围内，但是一旦超过 $\frac{\pi}{2}$，两个向量正交甚至方向背离，我们就无法容忍。换句话说，对于不同范围的 θ，放缩函数对其的变化率会有差异：对于区间 [$\frac{\pi}{2}$,π] 上的 θ，其产生的损失更为巨大。因此，在 SAMT#ang 方法中，我们将 $S(v)$ 设计成分段函数，并以 $\frac{\pi}{2}$ 为区间分隔点。具体计算过程如下：

$$S(\theta) = \begin{cases} 300 \times (1.5 - \cos(\theta)) & \theta < \frac{\pi}{2} \\ e^{\theta \times 3.0 + 1.0} + b & \theta \geq \frac{\pi}{2} \end{cases}$$

在上式中，当 θ 位于 [0，$\frac{\pi}{2}$] 时，$\cos(\theta)$ 的值域为 [-1，0] 中。考虑到即使两个向量在同一个方向，只要它们相等，它们之间的欧式距离就不为零，为了保留初始惩罚，我们将偏移量设置为 1.5（大于 1.0 的值）。本文设置其他常量的目的为进

一步使函数对自变量小的变化敏感以及获得大的函数值。b 用于保持分段函数值在 π 处连续。

所有的变换和超参数常数都是通过观察放缩函数的曲线[3]来启发式地选择的，为了清楚起见，我们将它们设置为常数形式。

2.3.3 基于 n 元文法的语义抽取器

为了提取句子级语义信息，我们设计了一个用于从原始 Token 嵌入中提取句子级抽象信息的基于 n 元文法的语义抽取器模块。我们很少能够用任何一个词来概括一个句子的意思，因此在直觉上，一个句子所蕴含的信息量比它的子结构—词所蕴含的要丰富。基于这一点，在分布式表示情境中，我们假设 Token 嵌入空间（Token Embedding Space）是句子语义空间（Sentence Semantic Space）的子空间。更形式化地来讲，若令 d_t 和 d_s 分别表示 Token 嵌入空间和句子语义空间的维度，则 $d_t \leq d_s$。因此，为了实现对句子语义表示的提取，从最初始的 Token 嵌入开始，我们需要首先将其从局部 Token 嵌入子空间维度扩充到能够张成全局句子语义空间的维度。这可以通过用对检索到的 Token 嵌入应用线性变换来实现的，即乘上一个形状为 $d_t \times d_s$ 的权重矩阵。之后，我们将得到的这些非收缩的表示连同对通过沿序列长度维度对它们进行平均得的表示一起馈送到基于 n 元文法的语义网络中。

在基于 n 元文法的语义网络中，我们首先通过 1 维卷积计算最常用的 1 元文法、2 元文法、3 元文法和 4 元文法的初始表示，即相应地使用大小为 1、2、3 和 4 的卷积核。所有卷积网络的输入通道数量均相同，但输出通道数量呈指数递减。详细来说，为提取 n 阶 n 元文法表示，若 Token 表示的维数是 d，则我们将卷积核的输入通道数量设置为 d，输出通道数量设置为 $d/2^n$。为了在不同的空间中获得 n 元文法候选表示，我们将对初始表示应用不同的非线性激活函数，分别为：$relu$，elu[4]，$tanh$ 和 $tanhshrink$[5]。一个词语在单独列出时可能是多义的，但是将其放置在给定的上下文中，它的意思就会变得清晰起来，高阶的 n 元文法可以起到这样一个上下文的作用，因此我们根据 n 元文法的阶数启发式地选择了这些激活单元。注意到，句子的意思在某种程度上对某些特定的词很敏感，因

[3] 与任何数据集无关的。

[4] https://pytorch.org/docs/1.0.0/nn.html#torch.nn.ELU

[5] https://pytorch.org/docs/1.0.0/nn.html#torch.nn.Tanhshrink

此，我们选择 *relu* 作为一元文法（1-gram）的激活函数，因为 *relu* 被证明与一个深层叠加的 sigmoid 网络（深度置信网络，Deep Belief Network, DBN）等价，后者可以很好地建模输入和输出的联合分布。就二元文法（2-gram）而言，它在大多数神经语言处理任务中几乎与一元文法同等重要，但有时稍微更有意义，因此我们需要保留少量的负的激活值，而 *elu* 是合适的。至于三元文法（3-gram），根据我们不完全的观察，它在大多数情况下是无意义的，所以我们将其限制在一个有限的范围内，*tanh* 满足我们的要求。对于四元文法（4-gram）时，这个高阶的 n 元文法具有更精确的表示能力，所以相对三元文法来说，施加的限制可以被稍微解除，*tanhshrink* 成为我们的选择。

由于大多数高阶的 n 元文法在语言学意义上是非法的，为了提取更自然的语义表示，我们使用多层感知机（Multi Layer Perceptron，MLP）来检测候选 n 元文法表示与非收缩的平均表示之间的相关性，根据候选的 n 元文法表示与非收缩的平均表示间的相关强度来为它们分配权重。这可以通过软注意机制（Soft-Attention Mechanism）来完成的，其中将非收缩的平均表示作为查询（Query），候选的 n 元文法表示则既作为键（Key）又作为值（Value）。然后，有效的 n 元文法表示通过计算这些候选表示的加权和求得。最后，我们将得到的有效的 1 元文法、2 元文法、3 元文法和 4 元文法表示拼接起来，并对拼接后的向量再次应用一个窄缩的线性变换重新投影成语义维度，在经过一个非线性激活函数后得到整句的语义表示。

以源句为例，其语义抽取的过程可以表述如下:

$$\mathbf{h} = W_s\mathbf{emb}(\mathbf{x}),\ \mathbf{a} = \frac{1}{T_x}\sum_{i=1}^{T_x}\mathbf{h}_i$$

$$\mathbf{g}_1 = \mathbf{attn_1}(\mathbf{a}, relu(\mathbf{conv_1}(\mathbf{h})))$$

$$\mathbf{g}_2 = \mathbf{attn_2}(\mathbf{a}, elu(\mathbf{conv_2}(\mathbf{h})))$$

$$\mathbf{g}_3 = \mathbf{attn_3}(\mathbf{a}, tanh(\mathbf{conv_3}(\mathbf{h})))$$

$$\mathbf{g}_4 = \mathbf{attn_4}(\mathbf{a}, tanhshrink(\mathbf{conv_4}(\mathbf{h})))$$

$$\mathbf{s} = tanh(W_r concat(\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4))$$

以上方程式中，为了简略起见，我们对线性变换省略了偏置项。**h** 和 **a** 是非收缩的源端表示和非收缩的平均表示。$W_s \in \mathbb{R}^{d_s \times d_t}, W_r \in \mathbb{R}^{d_s \times d_G}$ 为权重矩阵，其中

$d_G = \sum_{j=1}^{4}(d_s / 2^j)$。**emb** 表示 Token 嵌入模块。$\mathbf{attn_k}$ 和 $\mathbf{conv_k}$ 是注意模块和卷积网络，其中 $k = 1, 2, 3, 4$。**s** 则是最终提取到的源端语义表示。

语义抽取器的具体细节如图 2.2所示。圆圈代表向量表示，在同一水平线上，

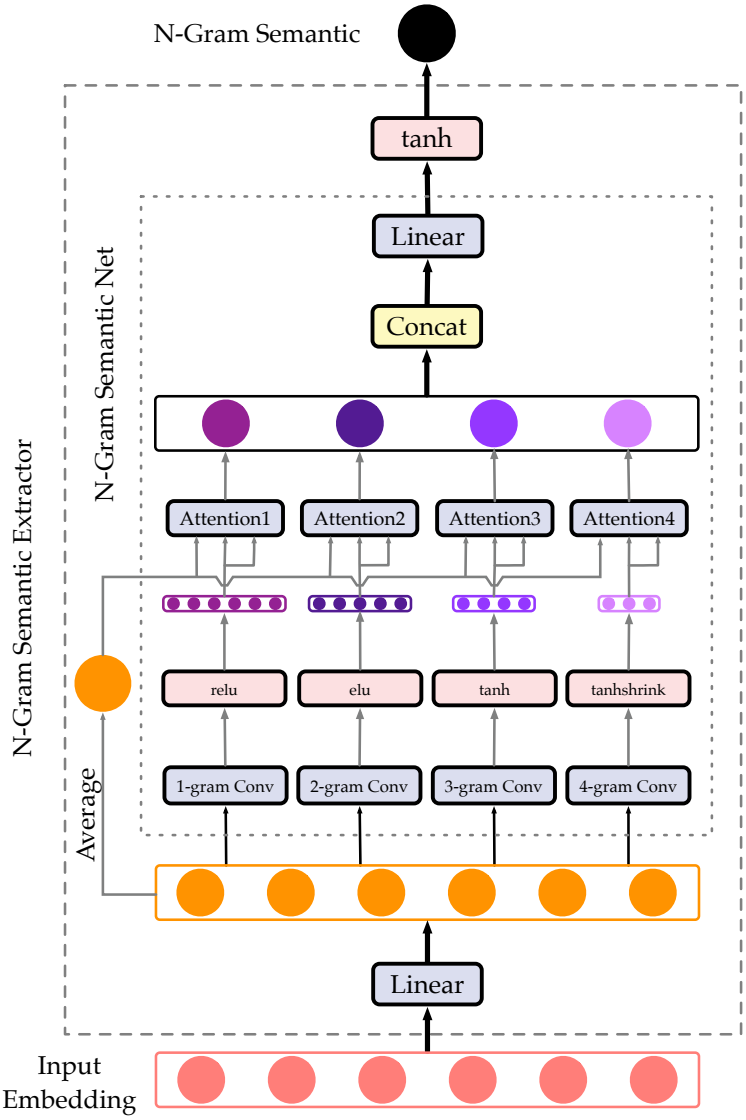

图 2.2 基于 n 元文法的语义抽取器

Figure 2.2 N-Gram Based Semantic Extractor

圆圈的颜色越深代表对应的表示维度越高。

2.3.4 融合句子级语义信息的可广播集成网络

获取到了句子的语义表示后，为了利用其指导生成译文的过程，我们需要将它集成到解码器中。这样，除了从编码器获得的抽象表示之外，我们的模型还将引入原始的语义信息流到解码器中，如图 2.3所示。

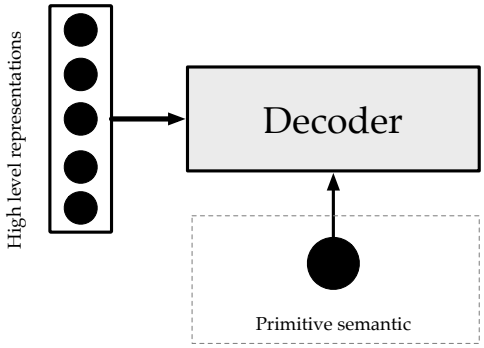

图 2.3 流向解码器的信息流

Figure 2.3 Information Flows to Decoder

很显然，我们可以把这个集成过程放在解码器的任何位置，只要有一个兼容函数模块能够融合语义表示和编码器输出、解码器输入中的至少一个。在这项工作中，我们将这个融合过程放置在解码器的自我注意力模块和编码器-解码器注意里模块之间，这样语义信息能够扩散到部分生成序列的相关表示中。在具体做法上，我们简单地先将语义表示重复到与当前生成的序列相同的长度，然后将前一个模块的输出与具有与当前序列长度的语义表示向量连接起来。为了与源端表示（编码器输出）进行交互，我们将以上混合生成表示和句子语义表示重新投影到源端表示维度。由于编码器输出或解码器输入的长度是可变的，而语义表示本身是固定的，我们需要在长度维度上对语义表示进行广播。因此我们称此兼容函数模块为可广播集成网络。

2.3.5　用于函数近似的语义映射网络

尽管源端和对应的目标语义来源于相同维度的 Token 嵌入空间，并且以相同的方式被抽取，但是并不能保证它们总是处于相同的语义空间区域。为了应用语义对齐准则，我们首先需要将源端语义表示和目标端语义表示映射到相同的层次。由于测试阶段，我们仅能够获取到源端的信息，这本项研究工作中，我们选择将目标端语义表示映射源端语义表示所在区域，这样做避免了对源端语义表示进行映射，因此既可以减少模型参数量又能减轻解码的计算量。深层神经网络被证明具有函数近似的功能 (Goodfellow 等, 2016)，即隐层单元数量足够多或者足够深的神经网络能够以任意精度逼近一个函数，利用此性质，我们提出使用一个语义映射网络将目标端语义表示映射到源端语义表示空间。在语义映射网络中，我们通过堆叠前馈网络来增加深度深度。为避免参数数量增加过快，我们

采用具有瓶颈结构的双层前馈网络作为用于堆叠的基本块。语义映射网络的结构如图 2.4所示，其由具有双残差连接的瓶颈结构的子层堆叠而成，图中矩形的彩色圆圈数量表示向量的维数。

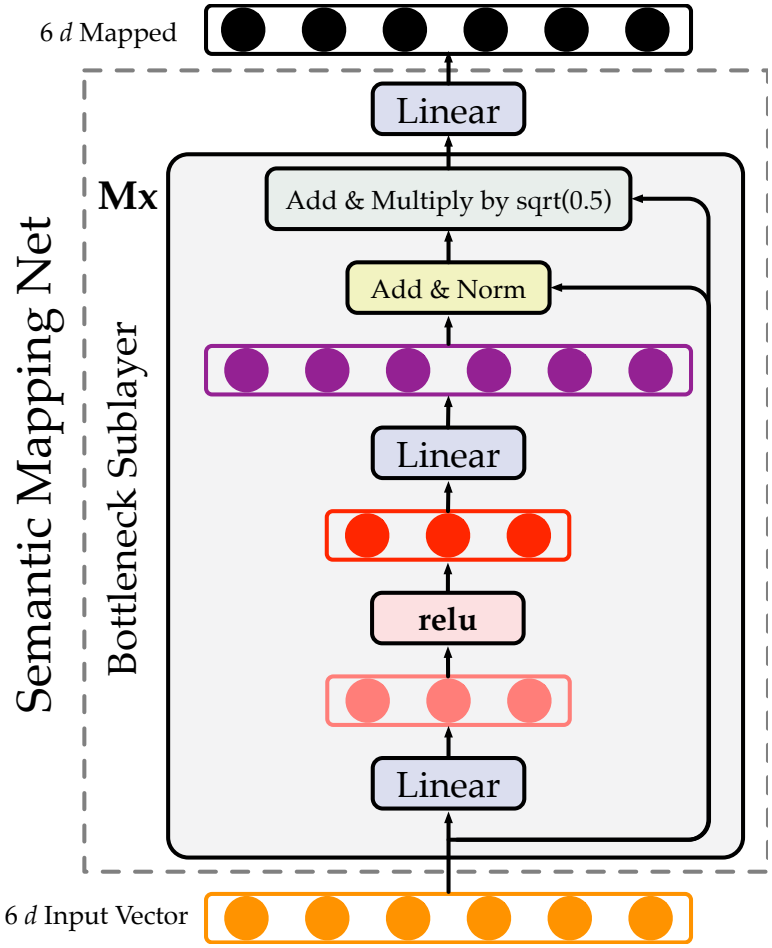

图 2.4 瓶颈结构的语义映射网络

Figure 2.4 Bottleneck Structure Semantic Mapping Network

在瓶颈结构的子层中，输入向量首先经由一个形状为 $W_{in} \in \mathbb{R}^{d_i \times d_s}$ 的矩阵使维度收缩至原来的一半，即 $d_s = 2d_i$，之后流向非线性激活的 *relu* 单元，然后又通过另一个形状为 $W_{out} \in \mathbb{R}^{d_s \times d_i}$ 的矩阵将维度扩展为输入向量的初始维度。为了在利用深度的同时减轻协变量偏移（Covariate shift）现象，按照 Vaswani 等 (2017) 的做法，我们对瓶颈结构的输入和输出添加了残差连接 (He 等, 2015) 并对残差结果应用层标准化（Layer Normalization） (Ba 等, 2016) 技术。我们参考了 Gehring 等 (2017) 的实现细节，再次对瓶颈结构的输入和层标准化的输出添加残差连接，并将结果乘上 $\sqrt{0.5}$。最后为了获得最终的映射结果，在流经过所有瓶颈结构的子层后，我们对输出进行重新投影。

2.3.6　显式语义对齐框架模型架构

在前面的内容中，我们提出句子语义空间概念模型后，分别引入了用于实现显式语义对齐的各种组件，并详细介绍了其引入动机和具体实现细节。以 Transformer 架构为基础，我们绘制了 SAMT 的整体模型结构，如图 2.5所示。

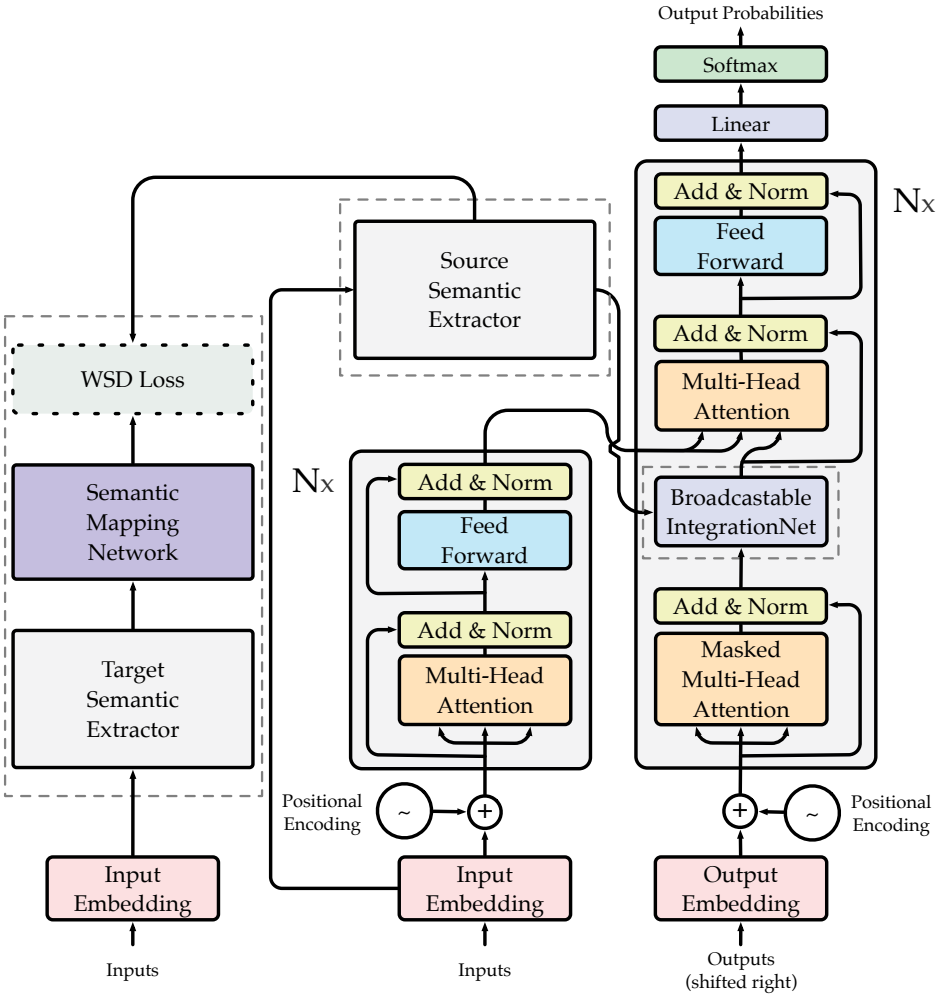

图 2.5 显式语义对齐模型架构 SAMT 概览

Figure 2.5 Overview of Explicit Semantic Alignment Architecture, SAMT

上图中，虚线框内的组件是本工作引入的模块，其余的为原始的 Transformer 架构。“WSD Loss” 为包裹的语义散度（Wrapped Semantic Divergence，WSD）损失[6]是最终训练目标的重要组成部分，其由语义散度和正则化项组成，对抽取得到的源端语义和转换为同一层次语义空间的目标端语义表示进行计算。

值得一提的是，由于在语义抽取器中引入了 n 元文法，使得语义抽取器即使

[6]直接应用语义散度作为训练准则几乎不起效，详细原因参见 2.3.7节。

没有自回归结构也具有一定的捕获时序的能力，因此输入可以不融入位置编码。为了提取确定的语义表示，语义抽取器中不引入随机因数。

2.3.7 研究难点：语义坍缩

在引入了所有合理的模块后，再应用语义散度训练准则，我们就可以以端到端的方式来训练新的模型了。然而，由于 Transformer 中强大的自回归解码器，加上模型结构中存在大量的几乎无监督信号的自由参数，我们尝试融入的语义信息会被迅速忽略，即一个小批量中所有样本学习到的语义表示完全相同意味着它们几乎毫无用处。通过进一步的调试分析，我们发现这种语义坍缩（Semantic Collapse）现象是由于训练过程中映射后的目标端语义表示总是过于接近源端语义表示导致的。

为了学习有用的语义表示，必须要防止源端语义表示和目标端语义表示在训练过程中始终过于接近。具体来说，对于通过将源端表示向量减去目标端语义表示计算得到的差，我们希望其保持稠密，即差向量中的非零元素尽可能的多。在机器学习社区中，向量中非零元素的个数是被普遍接受的 L0 范数的定义，并且对模型参数应用 L0 正则化将产生稀疏解。借鉴于参数稀疏惩罚的思想，我们推测在差向量上施加类似于 L0 正则化可能能够满足保持向量稠密的要求。由于 L0 范数不可微的特性，我们选择 L0 范数的最佳凸近似 L1 范数作为正则化约束函数。同时，为了学习到信息量更多的语义表示，我们还向源端语义表示和目标端语义表示施加了 L1 约束来防止抽取到的语义表示向量倾向于学成 0。因此，额外的约束项的形式如下:

$$\mathcal{L}_1(\mathbf{x}, \mathbf{y}; \Theta) = \mathcal{L}_1(\mathbf{s}, \mathbf{t}) + \lambda \cdot (\mathcal{L}_1(\mathbf{s}) + \mathcal{L}_1(\mathbf{t}))$$

其中 $\mathbf{s}, \mathbf{t}$ 分别是源端和目标端语义表示向量，它们的抽取计算公式如下:

$$\mathbf{s} = \mathbf{SourceSemanticExtractor}(\mathbf{x}; \Theta)$$

$$\tilde{\mathbf{t}} = \mathbf{TargetSemanticExtractor}(\mathbf{y}; \Theta)$$

$$\mathbf{t} = \mathbf{SemanticMappingNetwork}(\tilde{\mathbf{t}}; \Theta)$$

以上措施能够减轻源端语义坍缩问题，但是目标语义坍缩问题依然存在。我们通过修改 L1 损失的实现方式解决了这个问题。具体来说，我们对 L1 损失设置阈值机制，一旦计算出的 L1 损失偏离了预设的阈值，将会被严重地惩罚。算

法 1给出了算法的整体流程。修正的 L1 损失与语义散度损失以一种对抗的方式一起工作：如果语义散度损失变小，则差向量的 L1 损失迅速增加，反之亦然。我们推测是放缩惩罚的异步性使得一对向量一直保持着差异，从而避免了它们始终过于接近的窘境，进而解决了语义坍缩问题。

算法 1 修正的 L1 损失算法 $\mathcal{L}_1(\boldsymbol{v}1, \boldsymbol{v}2)$

Require: d 维向量 $\boldsymbol{v}1, \boldsymbol{v}2$ （$\boldsymbol{v}2$ 是可选的），阈值 η, 用于数值稳定的 ϵ

if $\boldsymbol{v}2$ 未设置 **then**
    $loss = \sum_{i=1}^{d} |v1_i - v2_i|$
else
    $loss = \sum_{i=1}^{d} |v1_i|$
end if
计算缩放后的损失 $loss = loss \,/\, \eta$
if $loss > 1.0$ **then**
    计算平方 $loss = loss^2$
else
    计算倒数 $loss = 1.0/(loss + \epsilon)$
end if
返回 $loss$

2.3.8　损失函数：训练目标

由于语义散度设计没有经过严格的推理，加上分布式语义表示的不可知论性，为了使模型能够工作，我们需要加上最大似然估计的训练准则。在训练过程中，语义对齐目标在早期阶段占主导地位，然后逐渐采用最大似然准则来拟合数据。我们通过模拟退火技术来逐渐增强语义散度约束强度。将以上因素考虑在内，我们得出了如下的训练目标，其中 T 表示 Token 数量:

$$\mathcal{L}(\Theta) = \frac{1}{T}\Big(\mathcal{L}_{\text{MLE}}(\Theta) + \alpha \cdot \mathcal{L}_{\text{WSD}}(\Theta)\Big)$$

$$\mathcal{L}_{\text{MLE}}(\Theta) = -\sum_{m=1}^{N}\sum_{i=1}^{T_y} \log P(y_i^m | \mathbf{y}_{<i}^m, \mathbf{x}^m; \Theta)$$

$$\mathcal{L}_{\text{WSD}}(\Theta) = \sum_{m=1}^{N}\Big(\mathcal{L}_{\text{SD}}(\mathbf{s}^m, \mathbf{t}^m) + \mathcal{L}_1(\mathbf{x}^m, \mathbf{y}^m; \Theta)\Big)$$

2.4 实验结果与分析

为了验证本项工作提出的模型的有效性，我们设计在多个数据集的多个语言对上进行实验，通过对实验结果进行分析得出方法有效的结论。在本节中，我们将依次详细介绍实验所使用的数据集、用于对比的基线系统、运行的参数设置、训练用到的方法和技巧，然后列出实验结果并进行分析。由于本项工作引入了多个额外组件，我们还进一步的按照控制变量的实验准则进行消融实验来分析各个组件对整个模型的影响。为了对整个模型工作的过程有更加直观的理解，我们亦绘制了训练过程中的损失和评估指标的变化趋势曲线。同时我们也设计额外的实验来验证所引入的连续的句子语义实际起到的作用并分析其重要程度。本节还对超参数的设置进行了实验和分析。最后，我们通过展示经过合理挑选的示例来从侧面说明相对于基线系统的改进。

2.4.1 实验数据

我们在小规模的英语 ⇔ 罗马尼亚语（En⇔Ro）、中等规模的中文 ⇒ 英语（Zh⇒En）以及大规模的英语 ⇒ 德语（En⇒De）翻译任务上进行了实验。对于这些任务，我们对参考译文进行子单元拆分（Tokenize），并使用由 multi-bleu.perl 脚本计算出的 BLEU 值 (Papineni 等, 2002) 来评估翻译质量。

对于 Zh⇒En 数据集，训练数据从 LDC 语料库中提取，由 1.25M 个句子对组成[7]，含 27.9M 个中文 Token 和 34.5M 个英文 Token。我们选择 NIST 2002 (MT02) 数据集作为验证集，NIST 2003 (MT03)、NIST 2004 (MT04)、NIST 2005 (MT05)、NIST 2006 (MT06) 和 NIST 2008 (MT08) 数据集作为测试集，它们分别包含了 878、919、1788、1082、1664、1357 个句子。我们主要在此数据集进行实验。

对于 En⇔Ro 数据集，我们使用由 Lee 等 (2018) 预处理过的语料库，该语料库包含了 608K 个句子对。我们使用他们划分的均由 1999 个句子组成的验证集和测试集。

对于 En⇒De 数据集，我们使用由 tensor2tensor (Vaswani 等, 2018) 预处理过的 WMT16 语料库，该语料库包含了 4.5M 个句子对。我们使用 newstest2013 作为验证集，newstest2014 作为测试集，它们相应地包含了 3000、3003 个句子。

[7]这些句子对主要是从 LDC2002E18、LDC2003E07、LDC2003E14、LDC2004T07、LDC2004T08 和 LDC2005T06 的 Hansards 部分抽取的，共 1,252,977 个句对。

2.4.2　对比基线

我们使用1.2.2节中提到的当前最先进的 Transformer 结构作为基线系统。

2.4.3　运行配置

我们按照 Sennrich 等 (2016) 的做法，将单词处理成子词单元。我们对 Zh⇒En 数据集的源语言和目标语言句子分别应用了操作数为 30K 的字节对编码（Byte Pair Encoding，BPE）技术，产生了 29M 个中文 Token 和 35.1M 个英文 Token。我们分别取源端和目标端语料中出现频数最高的 32768 个 Token 作为源端和目标端词汇表，从而得到分别由 32768 个和 29408 个 Token 构成的源端、目标端词汇表，并分别覆盖了该数据集中 99.83% 和 100% 的 Token[8]。我们分别对预处理过的 En⇔Ro 数据集、En⇒De 数据集构建联合词汇，具体来说，先将源端语料和目标端语料拼接起来，然后对拼接的语料应用操作数为 40K、32K 的字节对编码技术，并进一步将词表大小限制为 40K 和 32768，从而生成了含有 34976 和 32768 个词类的联合词汇表。

我们在一个重新实现的开源工具包 fairseq (Ott 等, 2019) 上实现了本文基于 Transformer 的所有方法。除非另有说明，我们所有的模型均使用 fairseq[9]的默认设置。在所有实验中，样本均以相同的顺序呈现，以保证训练过程中不会引入被证明对性能和收敛速度都有积极的影响课程学习（Curriculum Learning）(Bengio 等, 2009) 因素。出于对参数数量的考虑，我们仿照 Liu 等 (2019) 的做法在解码器中进行了权重绑定（Decoder WT）(Press 和 Wolf, 2017)[10]。为了迫使模型中的语义抽取器学习得更充分，我们对解码器的输入馈送（input feeding）应用了 $p = 0.2$ 单词丢弃（词丢弃）技术，即依概率 p 将目标端序列的 Token（序列结尾符 EOS 除外）替换为 UNK，这产生了更强的基线。

对于 Transformer#base 模型，我们将编码器和解码器的层数都设置为 6。我们使设置模型的维度为 512，前馈层的内部维度为 2048，多头注意力模块头的数量设置为 8。我们应用了概率为 0.1 的 dropout (Srivastava 等, 2014) 技术。在计算最大似然损失时，我们应用了不确定性为 0.1 的均匀标签平滑技术。我们使用

[8]对于语料库中所有不在词汇表中的 Token，即集外词（Out of Vocabulary，OOV），我们按照通常做法将它们统一映射为 UNK 符号。

[9]https://github.com/pytorch/fairseq

[10]解码器的 Token 嵌入权重矩阵和 softmax 之前的线性层权重共享参数。

Adam (Kingma 和 Ba, 2015) 优化器，并设置超参数 β_1= 0.9, β_2 = 0.98, $\varepsilon = 10^{-8}$。按照 Ott 等 (2018) 的做法，我们使用逆平方根的分段学习率调度方式：学习率在最初的 4000 个训练步线性增加至 7e-4，之后它与训练步数的平方根成比例地衰减。对于 Transformer#big，我们采用与 Ott 等 (2018) 相同的设置。

在新引入的模块中，我们将语义维度设置为模型大小的两倍，即 1024。而对于基于 Tansformer#big 的模型，受限于计算资源，语义维度设置为与模型维度相同。语义映射网络的深度设置为 2。对于参数化的语义散度，λ 设置为 0.1，而阈值 η 设为 2.0。

2.4.4 训练细节

我们对新增加的 WSD 部分损失函数应用模拟退火技术，即训练目标中的参数 α 以 0.05 的斜率线性增加到 1.0，对于基于 Transformer#big 的模型，α 设置为 0.02。为了保留原始信息，我们在新增加的模块中均不引入随机性，即在训练或推理过程中不应用 dropout。训练过程中，每个小批量样本（mini-batch）包含不超过 32768（base 模型为 4096×4×2，big 模型为 2048×4× 4）个源端或目标 Token。我们在每 1000 个训练步过后或这完成了对训练集一轮的遍历时评估模型，以便提前终止训练。除非另有说明，否则我们均对单个模型使用大小为 4 集束宽度（beam width）进行解码。对于所有翻译任务，用于选择生成译文的长度惩罚（length penalty）设置为 0.6。我们在 CUDA 9.2 上使用 4 张 Titan XP 型号的图形显卡上训练所有的模型。

2.4.5 实验结果

对于所有的 base 模型，我们将模型训练至收敛，然后报告在验证集上单模型取得的最高的 BLEU 值的那组结果。

表 2.1报告了 Zh⇒En 数据集翻译任务上 NIST 验证集和测试集的实验结果（大小写不敏感的 BLEU 值）。在表 2.1中，“**#Param.**” 代表模型参数的数量，“⇑” 表示结果显著好于基线 ($p < 0.01$)，而 “↑” 表明 $p < 0.05$。所有显著性检验都是通过使用成对自助重采样（paired bootstrap resampling）(Koehn, 2004) 来测量的，其中实验次数为 1000，采样比例为 0.5。

从表 2.1中可以看出我们提出的所有方法的实验结果的性能均好于基线，其中基于相似性的方法 SAMT#sim 优于基于角度的方法 SAMT#ang。我们把导致

System	Architecture	# Param.	MT02	MT03	MT04	MT05	MT06	MT08	AVG
Existing NMT Systems									
Meng 和 Zhang (2019)	Transformer#big	277.6M	45.32	44.13	45.92	44.06	44.78	35.33	42.84
Meng 和 Zhang (2019)	DTMT#4	208.4M	47.03	46.34	47.52	46.70	46.74	37.61	44.98
Our NMT Systems									
Transformer#1	#base	91.0M	45.33	44.10	45.99	44.76	44.02	34.59	42.69
Transformer#2	#1 + Decoder WT	76.0M	45.77	44.92	45.82	45.66	44.16	35.14	43.14
Transformer#3 (baseline)	#2 + $p_{word\_dropout} = 0.2$	76.0M	46.60	45.32	46.78	45.35	44.89	36.23	43.71
this work	SAMT#ang	100.7M	**47.51**$^{\Uparrow}$	46.80$^{\Uparrow}$	47.75$^{\Uparrow}$	46.33$^{\Uparrow}$	45.18	36.58	44.53
	SAMT#sim	100.7M	47.43$^{\uparrow}$	**46.90**$^{\Uparrow}$	**48.07**$^{\Uparrow}$	**46.89**$^{\Uparrow}$	**46.00**$^{\Uparrow}$	**37.53**$^{\Uparrow}$	**45.08**

表 2.1 NIST Zh⇒En 数据集翻译任务实验结果

Table 2.1 Results on the NIST Zh⇒En Translation Task

这两种方法得到的结果差异的原因归结为变量的异步性：cos 在 $\pi/2$ 区域附近变化剧烈，而变量 θ 本身是一个相对平滑的线性过渡。

#	System	Architecture	#Param.	En⇒Ro	Ro⇒En
Existing NMT Systems					
1	Transformer (Lee 等, 2018)	Transformer#base	-	32.40	32.06
Our NMT Systems					
3	Transformer#1	Transformer#base (Decoder WT)	62.0M	33.20	32.82
4	Transformer#2 (baseline)	#1 + $p_{word\_dropout} = 0.2$	62.0M	33.49	33.02
5	*this work*	SAMT#sim	86.7M	**34.22**$^{\Uparrow}$	**33.61**$^{\uparrow}$

表 2.2 WMT16 En⇔Ro 测试集翻译任务实验结果

Table 2.2 Results on the WMT16 En⇔Ro Translation Tasks (Test Set)

根据在 Zh⇒En 数据集上的实验结果，我们仅在 En⇔Ro, En⇒De 数据集上验证 SAMT#sim 方法的有效性。表 2.2和表 2.3分别列出了在 WMT16 En⇔Ro 和 En⇒De 的实验结果，实验表明在更小规模或者更大规模的数据集上，我们的模型依然优于基线模型。注意到 Zh⇒En 数据集上的参考译文有多个，吻合我们模型的假设，相对来说性能提升更大。

2.4.6　消融实验

由于本项工作引入了多个额外组件，即语义抽取器、语义映射网络、可广播集成网络，还引入了多个可变因素，例如，为解决语义坍缩现象而对语义散度进行了包装和修正，为了分别评估它们对于整个模型的重要性，我们进行了屏蔽了模型的某个组件或因素来测试在 Zh⇒En 数据集上性能变化的消融实验。表 2.4列

Systems	Desc.	# Param.	Test
ConvS2S (Gehring 等, 2017)	single	-	26.43
Transformer (Vaswani 等, 2017)	ensemble	213M	28.40
RNMT+ (Chen 等, 2018b)	single	-	28.49
SNMT (Yang 等, 2019)	-	276.8M	28.92
Our Baseline	single	209.9M	28.40
	ensemble	209.9M	28.60
Ours + SIM	single	243.5M	**28.75**$^{\uparrow}$
	ensemble	243.5M	**28.91**

表 2.3 WMT16 En⇒De 测试集翻译任务实验结果 (big 模型)

Table 2.3 Results on the WMT16 En⇒De Translation Task with the big Model (Test Set)

出了与额外组件、参数量等影响因子有关的消融实验结果。在表 2.4中，以 “-” 开头的系统描述意味着我们从 SAMT#sim 中移除了该模块。特别地，“-Semantic Extractor” 给出的结果是我们使用平均 Token 嵌入或编码器输出作为句子语义表示来进行实验获得的最好结果，包括该表示进行了和没有进行 *tanh* 非线性激活的情况。

#	System Description	# Para.	MT02	MT03	MT04	MT05	MT06	MT08	AVG	Δ
1	Baseline (6 layers)	76.0M	46.60	45.32	46.78	45.35	44.89	36.23	43.71	-
2	SAMT#sim	100.7M	**47.43**	46.90	48.07	46.89	46.00	37.53	**45.08**	**+1.37**
3	- $\mathcal{L}_{\text{SD}}(\mathbf{s},\mathbf{t})$ (Semantic Div Loss)	100.7M	46.53	44.92	46.83	45.87	45.38	36.20	43.84	+0.13
4	- $\mathcal{L}_{1}(\mathbf{x},\mathbf{y};\Theta)$ (Modified L1 Loss)	100.7M	46.47	45.46	47.14	46.32	44.99	36.13	44.01	+0.30
5	- Semantic Mapping Net	97.5M	46.26	45.64	47.42	46.29	45.46	36.76	44.32	+0.61
6	- Semantic Extractor	84.9M	46.08	43.91	46.50	45.61	43.78	34.84	42.93	–0.78
7	Share Semantic Extractor	92.3M	47.17	46.65	48.04	47.42	45.73	36.85	44.94	+1.23
8	Non Modified L1 Loss	100.7M	46.88	46.39	47.14	46.06	45.31	36.15	44.21	+0.50
9	Baseline Deeper (10 layers)	105.4M	46.51	45.19	47.14	45.91	44.78	36.60	43.92	+0.21

表 2.4 NIST Zh⇒En 数据集翻译任务消融实验结果

Table 2.4 Ablation Study Results Results on the NIST Zh⇒En Translation Task

在表 2.4中，从第 1 行到第 4 行的结果来看，我们发现语义散度和修正的 L1 损失只有当组合在一起时才能很好地工作，没有它们中的任意一项，性能都会下降很多。从第 5 行的结果来看，相对于第 1、2 行，为了获得更好的性能，语义映射过程是必要的。而通过比较第 1、2、6 行的实验结果，语义抽取器无疑是至

关重要的，缺失了它会直接导致翻译质量急剧下降，甚至模型不工作。同时观察第 1、2、7 行的结果可以发现，即使共享源端和目标端语义抽取器，模型也能较好的工作，我们猜测可能是因为语义抽取器抽取到了两种语言的某些共同属性。第 8 行的实验结果表明修正的 L1 损失也是一个关键因素，没有这个因素，得到的提升比较有限，造成这种局面的很大原因是 Transformer 中的自回归解码器能力已经很强大以致于约束很弱的语义对齐过程几乎直接被忽略。由于引入了多个新的模块，我们的方法不可避免地引入了一定量的参数 100.7M（相对于基线系统的 76.0M）。我们在更深的架构上 Transformer#base 模型进行了实验，该模型的编码器和解码器均有 10 个块堆叠而成，在参数量上与我们的模型大致相同（略多）。根据第 9 行的结果和 Transformer#big 中报告的结果（表 2.1中的第 1 行），我们可以得出翻译性能的提升是由我们的方法而不是由于引入了额外的参数带来的的结论。

2.4.7　损失与 BLEU 曲线

为了获得对训练过程中训练目标与最终评估模型性能使用的 BLEU 值变化趋势的直观理解，我们分别在图 2.6和图 2.7展示了训练目标中的整体损失与其中主要的两种损失和评估指标随训练轮数变化趋势的折线图，其中横轴代表训练的轮数。从图中可以观察到，新引入的对齐损失（包裹的语义对齐散度：由语义散度和修正的 L1 约束构成）在整体损失中一直处于主导地位并且变化无规则，并且最高 BLEU 值[11]并不是在整体损失最小的情况下获得的，我们猜测这是由于句子语义空间的相对位置在不断变换所造成的。

2.4.8　语义分析

一方面，相对于单词，句子的语义的抽象层次更高，并且对于一个句子来说，大多数时候缺失部分不重要的单词并不会影响我们对整个句子的理解。因此我们认为融入句子语义有助于增强对缺失输入的鲁棒性。因此，我们在 NIST Zh⇒En 测试集上分别对基线模型 Base、最佳的改进模型 SAMT 以及性能与基线模型可比的改进模型 Comp（性能略差）进行解码，其中测试集源端的句子中部分句子被施加了不同程度的随机掩盖。表 2.5列出了源端输入在不同程度的缺失（“p” 表示我们依概率 p 对源端输入应用词丢弃技术来构建缺失输入。）下对整个

[11]我们在第 28 轮结束时获得了最好的结果。

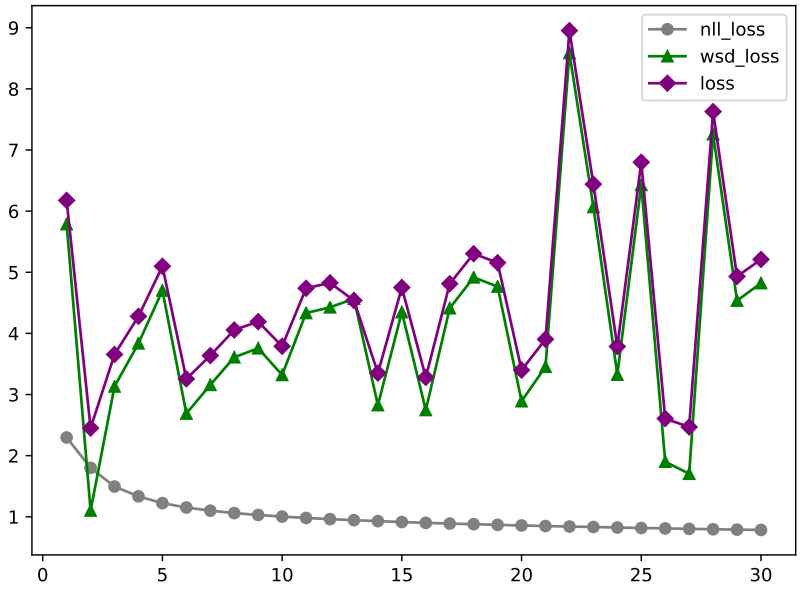

图 2.6 训练损失曲线 (次级别，log 尺度)

Figure 2.6 Curves of Training Loss (word level, log scale)

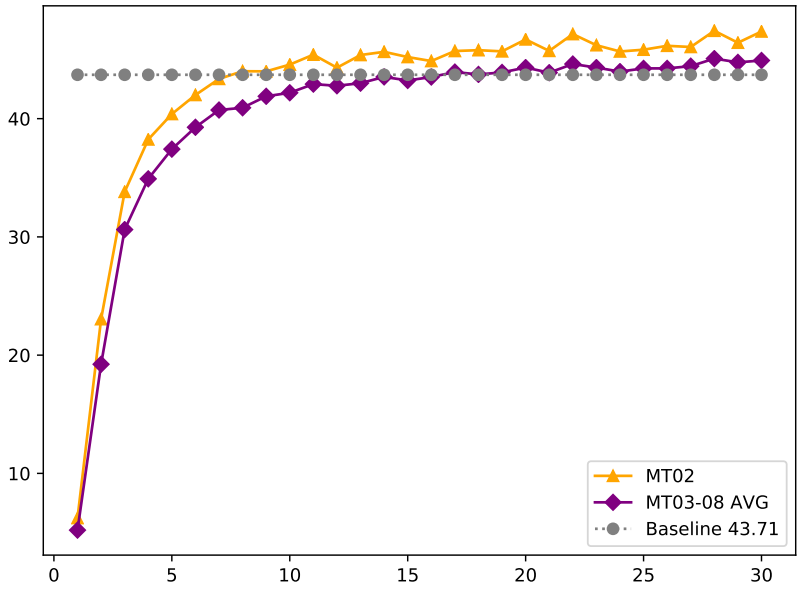

图 2.7 BLEU 变化曲线

Figure 2.7 Curves of BLEU

测试集的覆盖率（C(%) 指源端输入中非 UNK 占的百分比。）以及三个模型在存在不同程度缺失的输入下的解码表现（结果以"AVG MT03-08" 格式出现，即给出的 BLEU 值为 MT03、MT04、MT05、MT06、MT08 上的均值）。表 2.5的结果表明，在不同程度缺失的输入下，甚至于我们的 Comp 模型也一致地优于未集成句子语义的基线模型 Base，这验证了抽取到的语义有助于增强模型对缺失输入的鲁棒性。

Model	$p = 0.00$	$p = 0.05$	$p = 0.10$	$p = 0.15$	$p = 0.20$
C(%)	99%	94%	89%	84%	79%
Base	43.71	40.10	36.45	32.97	29.66
Comp	43.61	40.26	36.74	33.11	29.77
SAMT	**45.08**	**41.35**	**37.64**	**33.93**	**30.41**

表 2.5 NIST Zh⇒En 句子级语义分析实验结果

Table 2.5 Analytical Results of Sentence Level Semantic on NIST Zh⇒En

另一方面，我们也对流入解码器的两种信息流（图 2.3），编码器输出（Token 的高层表示）和语义抽取器输出（句子原始语义）的影响力进行了分析。具体做法是，针对最佳的改进模型，我们在解码过程中分别对编码器输出和语义抽取器输出应用不同概率的 dropout。我们在图 2.8中展示了针对不同概率 dropout 的解码表现结果。从图中可以看出，语义表示不如编码器输出重要，即使被屏蔽了 60%，性能也只表现出轻微的下降。此外，当对语义表示施加幅度小于 0.01 的恒定扰动时，模型解码性能几乎没有变化，但施加从均匀分布中采样得来构建的噪声对于模型解码性能来说却是致命的。根据以上两种现象，我们推测可能是抽取到语义表示中的仅有部分维度起着重要作用。

我们还在 Zh⇒En 数据集上对语义表示的维度[12]如何影响性能进行了实验，实验结果如表 2.6所示。表 2.6的结果表明，语义维度不足或者语义维度过高均无法达到最佳性能。这进一步证实了我们对抽取到的语义表示中存在着一些重要的神经元推测方向的正确性：包含了这些神经元便能起到作用。

[12]注：模型维度为 512。

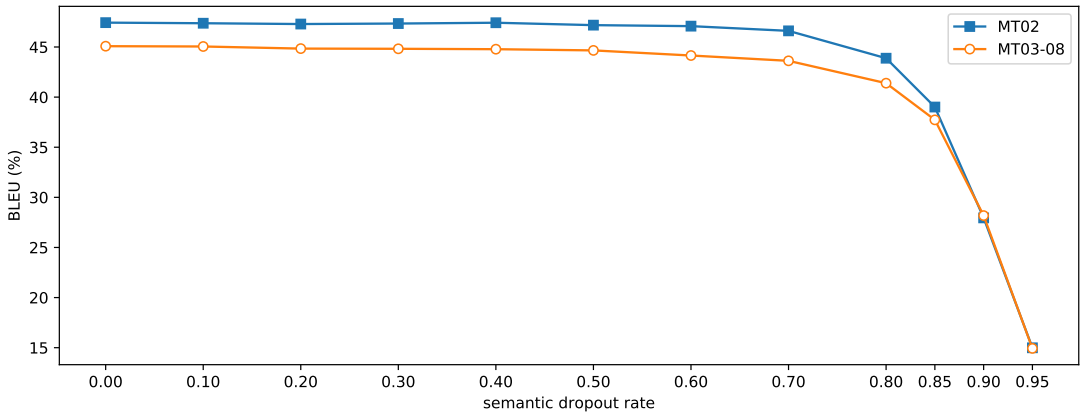

图 2.8 编码器输出和语义表征两种信息流对模型解码性能的影响

Figure 2.8 Influence of Encoder Output & Semantic Representation on Decoding Performance

#	dim	# Para.	MT06	MT02	MT03	MT04	MT05	MT08	AVG
1	512	84.4M	46.92	45.57	47.53	46.72	45.51	36.44	44.35
2	1024	100.7M	**47.43**	46.90	48.07	46.89	46.00	37.53	**45.08**
3	2048	161.5M	47.04	46.45	47.57	46.33	44.82	36.91	44.41

表 2.6 NIST Zh⇒En 不同语义维度实验结果

Table 2.6 Results of Different Semantic Dimensions on NIST Zh⇒En

2.4.9 超参数分析

在整个实验过程中，我们引入了一些重要的超参数，本节将对其中最为重要的两个超参数：用于将源端语义表示和目标端语义桥接至同一层次的语义映射网络深度 depth 和修正的 L1 损失中引入的阈值 η 进行分析。

语义映射网络深度的影响

直觉上，对齐质量对与两个要进行对齐的目标是否处于同一层次很敏感，即语义映射网络能否学成正确的函数空间来有效地将目标端语义表示映射到源端语义表示空间非常关键。而这种函数空间由具有前馈网络性质的语义映射网络来近似，2.3.5节中提到网络的深度是拟合函数精度的关键因素，因此语义映射网络的深度参数很关键。尽管由于语义表示的高维特性导致实际上是否真正对齐很难界定，但我们依然可以从能够被观察到的结果中看出一些端倪。具体来说，训练过程中的对齐好坏影响到解码时抽取到的源端语义表示质量，由于翻译的

本质是在语义下的序列转换应用，在有效的语义有助于提升机器翻译性能的假设下，我们可以通过观察模型的翻译性能来大致推断语义是否对齐的相对质量，更确切地来讲，观测 BLEU 的变化来确定语义对齐的质量。我们在表 2.7给出了在 Zh⇒En 数据集上不同语义映射网络深度[13]的实验结果。实验结果显示，随着语义映射网深度的加深，即从 0 层（不使用语义映射网络）到 4 层，模型性能先增加后降低。这我们的预期猜想：尽管更深的网络可以模拟更复杂的函数，但在有限的训练时间和不完整的数据的限制下，拥有更多参数的模型倾向于过拟合数据，而非学到更准确的函数。

#	depth	# Para.	MT06	MT02	MT03	MT04	MT05	MT08	AVG
1	0	97.5M	46.26	45.64	47.42	46.29	45.46	36.76	44.32
2	1	99.6M	46.98	46.35	48.01	46.43	45.38	37.15	44.66
3	2	100.7M	**47.43**	46.90	48.07	46.89	46.00	37.53	**45.08**
4	3	101.7M	46.89	46.54	48.52	47.02	45.83	36.75	44.93
5	4	102.8M	46.97	46.27	47.35	46.82	46.32	37.61	44.87

表 2.7 NIST Zh⇒En 语义映射网络深度超参实验结果

Table 2.7 Results of Hyper-Parameter Depth in Semantic Mapping Network on NIST Zh⇒En

修正的 L1 损失中阈值的影响

通过观察前期的结果实验，我们发现学习到的的语义表示向量的元素值不能太小，否则编码器输出将完全主导解码过程，致使发生语义坍缩问题。据此我们在 2.3.7节中通过基于阈值机制的修正 L1 损失解决了语义坍缩现象。具体地，我们在算法 1中设置了一个名为 threshold 的超参数 η 来控制向量中的信息量。我们针对该超参数进行了分析实验，实验结果如表 2.8所示，实验结果表明阈值是另一个关键的超参数。

2.4.10　案例分析

本节通过对基线模型（Base）和改进模型（SAMT）在验证集 MT02 上进行解码，总体 BLEU 值（%）分别为 46.60 和 47.43，从解码结果中按照如下程序选出了两个翻译示例：

[13]“depth” 是指语义映射网络中瓶颈子层的堆叠层数，即图2.4中的 “**Mx**” 记号。

#	threshold	# Para.	MT06	MT02	MT03	MT04	MT05	MT08	AVG
1	1	100.7M	46.97	46.51	47.48	46.65	45.96	36.99	44.72
2	2	100.7M	**47.43**	46.90	48.07	46.89	46.00	37.53	**45.08**
3	4	100.7M	46.83	46.44	47.87	46.71	45.96	36.53	44.70
4	8	101.7M	46.99	45.90	47.47	47.21	46.11	36.86	44.71

表 2.8 NIST Zh⇒En 修正 L1 损失中的阈值超参实验结果

Table 2.8 Results of Hyper-Parameter Threshold in Modified L1 Loss on NIST Zh⇒En

- 对两个模型解码出的句子分别评估单句的 BLEU；
- 计算对应每个句子的 BLEU 差值：$\mathbf{BLEU}(s_{SAMT}) - \mathbf{BLEU}(s_{BASE})$ 和其相反数 $\mathbf{BLEU}(s_{BASE}) - \mathbf{BLEU}(s_{SAMT})$；
- 对以计算出的 BLEU 差值进行排序。

我们在执行以上流程中对于以下情况的不进行操作：

- 任意一个模型解码出的句子与参考译文中的任意一个句子相同；
- 任意一个模型解码出的句子在计算 BLEU 值时任意阶的 n 元文法匹配精度为 0。

最后我们按 BLEU 差值选出最佳的样例和最差的样例：

- 最佳：$\max(\mathbf{BLEU}(s_{SAMT}) - \mathbf{BLEU}(s_{Base}))$；
- 最差：$\max(\mathbf{BLEU}(s_{Base}) - \mathbf{BLEU}(s_{SAMT}))$。

表 2.9展示了按照以上程序选出的两个示例，其中第 751 个句子是最佳样例，而第 818 个句子是最差示例。在表 2.9中，“src” 代表翻译原文，“ref1”、“ref2”、“ref3”、“ref4” 是相应的参考译文。对于每个样例的最后一行数据 “Comp”，我们列出了在该样例上两个模型 Base 和 SAMT 格式为 “BLEU，匹配 n 阶的 n 元文法精度”（%）的结果，其中 $n = 1, 2, 3, 4$。

表 2.9中的两个样例都表明了我们所提出的方法有更好的充分性和忠实性，即翻译后的意思与原文一致，即使该句计算得出的 BLEU 值比基线更低（不能以高精度匹配到参考译文）。

751	src	但 克雷蒂安 在 **上 星期四** 到 华盛顿 同 布什 举行 会谈 之后 ， 似乎 已 改变 原本 立场 。
	ref1	but chretien seems to have changed his original position after he went to hold talks with bush in washington d.c. **last thursday** .
	ref2	however , jean chretien seems to have changed his original stance after a meeting with bush **last thursday** in washington .
	ref3	but chretien appears to have changed his stance after meeting with bush in washington **last thursday** .
	ref4	but after chretien talked to bush **last thursday** in washington , he seemed to change his originally stands .
	Base	however , it seemed that chretien changed his original position after holding talks with bush in washington **on thursday** .
	SAMT	however , chretien seems to have changed his original stance after meeting with bush in washington **last thursday** .
	Comp	Base: **42.63**, 80.0/52.63/33.33/23.53, SAMT: **90.71**, 100.0/94.44/88.24/81.25
818	src	这 是 今年 入 春 以来 朝鲜 **经历 的** 第一 场 沙尘暴 。
	ref1	this is the first sandstorm to hit north korea since this spring .
	ref2	this is the first sandstorm in north korea since spring this year .
	ref3	this is the first spell of sandstorms that hit dprk since this spring .
	ref4	this is the first sandstorm north korea **experiences** after spring comes this year .
	BASE	this is the first sandstorm in north korea since spring this year .
	SAMT	this is the first sandstorm that north korea **has experienced** since spring this year .
	Comp	Base: **100.0**, 100.0/100.0/100.0/100.0, SAMT: **54.11**, 86.67/64.29/46.15/33.33

表 2.9 NIST Zh⇒En 验证集 MT02 上的语义对齐翻译样例

Table 2.9 Translation Examples of Semantic Alignment from MT02 on NIST Zh⇒En

2.5　本章小结

在本项工作中，我们注意到传统的神经机器翻译中仅关注词级别的匹配损失而没有对翻译前后的句子意思是否一致进行监督的事实，提出并探讨了神经机器翻译在分布式情境中的显式语义对齐问题。首先，我们通过建立一个能够解释一个句子存在多个翻译的句子语义空间概念模型 S3CM 后设计了语义散度的对齐准则来约束语义对齐。然后，我们引入必要的组件：基于 n 元文法的语义抽取器用于从原始 Token 嵌入中提取源语义信息和目标语义信息，可广播集成网络将源端语义表示信息融合到解码器中指导生成过程和用于将源端、目标端语义表示映射到同一个语义空间的语义映射网络。最后，在缓解了语义坍缩现象后，我们的方法显著提升了机器翻译性能。然而就翻译质量的提高而言，我们的方法还不太令人满意，并且 S3CM 的正确与否也有待考证，但本项工作为语义表示的显式对齐提供了一种方法，这可能会给语义信息起重要作用的应用带来好处。因为这项工作主要集中在建立一个明确的语义对齐框架，所以将来可以更深入地探索单个模块。

第 3 章　基于状态融合和输出矫正缓解曝光偏差问题的研究

3.1　引言

1.2.2节中提到，当前神经机器翻译三种主流的序列到序列模型架构均基于编码器–解码器架构，并采用端到端的方式进行训练。在训练过程中，为了预测下一个 Token，模型将整个源端序列和完全正确的部分目标端序列（Ground Truth）作为上下文，其中部分目标端序列可形式化为：

$$\mathbf{y}^* = \{y_1^*, y_2^*, y_3^*, y_4^*, \cdots, y_{t-1}^*\}$$

其中 y_i^* 表示该序列中的第 i 个 Token，t 为即将预测的第 t 个 Token，$\mathbf{y}^*$ 来源于训练句子对中的目标端句子。然而，在测试阶段，由于没有目标端句子，模型无法获得完全正确的部分目标端序列，因此使用从头开始生成的部分目标端序列来替代，该序列的形式化表示为：

$$\mathbf{y} = \{y_1, y_2, y_3, y_4, \cdots, y_{t-1}\}$$

在 $\mathbf{y}$ 中，y_i 均为解码器接受源端序列和部分生成的目标端序列 $\{y_1, y_2, y_3, y_4, \cdots, y_{i-1}\}$ 作为上下文预测出来的[1]。由于任何一个预测出来的 Token 都有可能是错误或者偏离原文的，一个完全从头开始生成的序列几乎必然存在着累积错误。由于模型在训练时完全没有被教导如何在这种存在错误的环境下进行预测，这种差异可能会导致测试中解码的上下文严重偏离模型在训练时接触的，引发错误累积现象，从而导致性能退化。

自从 Ranzato 等 (2015) 指出这种曝光偏差（Exposure Bias）问题以来，研究人员为缓解[2]它做了大量工作，可大致分为以下两类：

- Token 级别的探索：Venkatraman 等 (2015); Bengio 等 (2015); Goyal 等 (2017); Mihaylova 和 Martins (2019); Zhang 等 (2019) 通过在训练过程中动态地提供存在错误的上下文作为输入，使模型逐渐降低对这种差异的敏感程度；

[1]特别地，当 $i = 1$，部分生成的目标端序列为特殊的序列起始符号（Begin Of Sentence）**BOS**。在实际操作中，$\mathbf{y}^*$ 和 $\mathbf{y}$ 的第一个符号均为 **BOS**，此处为简化说明将其省略。

[2]曝光偏差问题始终存在，因为训练阶段和测试阶段观测的数据是不同的。为了获得更好地性能，训练时倾向于更多地去使用一切能利用的信息，而这些附加信息在测试阶段是不可得的。

- 序列级别的方法： Ranzato 等 (2015); Shen 等 (2016); Shao 等 (2018) 将全局的序列得分作为训练目标的基本单位，通过强化学习等方法进行训练，若训练和测试选择最终序列的方式一致，则可以视为间接地消除了这种差异。

所有的这些方法要么根据模型的能力逐渐在 Token 级别对数据进行重新排序以直接增强对存在错误的上下文的适应性，要么提出设计良好的序列级奖励来迫使模型学习到最优序列，从而间接地减小了存在错误的上下文对解码的影响。

本项工作中，我们在将曝光偏差问题扩展为训练过程与测试阶段解码上下文不一致的情形[3]下，进一步思考如何在不改变任何训练程序（training procedure）的前提下，仅通过对模型内部进行变化来达到减轻该问题对于模型性能的影响的通用且易于操作的方法。新提出的方法能够在不改变模型结构的前提下使已经收敛的模型再度“进化”，在保持性能不下降的同时，有效地抵御曝光偏差问题所造成的负面影响。

在本章接下来的内容中，我们将首先介绍前人在缓解曝光偏差问题方面相关的工作。然后深刻地分析传统曝光偏差问题产生的来源，并就此提出对问题改进的方法，即隐藏状态级别的内部状态融合方法。接着，我们对应用了状态融合的整体训练过程进行分析，发现并利用类似二次解码过程中的副产品构建辅助监督信号来进一步增强模型的性能，称该方法为逐层输出矫正。我们同样在不同规模、多个数据集的多个语言对上进行了实验，实验结果表明，内部状态融合和逐层输出矫正方法的应用能够保持模型性能不降，并在大多数情况下[4]能够取得显著的性能提升。此外，我们还深入分析了本项研究内容与前人工作的异同，并与一些引入了噪音的其他工作的区别和联系进行了说明。最后，我们对本项工作进行了总结。

3.2 相关工作介绍

3.1节中提到，自 Ranzato 等 (2015) 指出基于循环神经网络的序列建模训练存在曝光偏差问题起，便产生了一系列为缓解该问题的研究工作。它们大致可以分为以下两类：一种是通过在训练过程中动态地构造存在错误的上下文作为输入馈送来使模型逐渐习得对这种差异不太敏感的能力的 Token 级探索；另一种

[3]包括但不仅限于作为输入馈送的部分生成序列，如训练过程中可能还能用到一些其他的监督信号，但在测试阶段不可获得。

[4]偏差源直接相关的模块与直接进行结果预测的模块没有短路径联系。

是不只使用 Token 级的优化目标，而将生成序列的整体得分这种更大粒度的长远目标作为基本单位来训练模型，相当于间接地弥补了这种差异的序列级方法。

3.2.1　Token 级探索

Venkatraman 等 (2015) 提出将多步预测导致复合误差的问题公式化为模拟学习，并通过使训练数据作为“验证者（demonstrator）”对多步预测过程中产生的误差进行修正。Bengio 等 (2015) 使用一种计划采样 (Scheduled Sampling) 课程学习策略，将训练过程从使用真正的前一个 Token 的完全指导方案逐步改变为使用生成 Token 较多的指导方案。 Goyal 等 (2017) 进一步对 argmax 操作应用连续松弛来创建可微近似，以应用于计划采样方法。此外， Mihaylova 和 Martins (2019) 和 Zhang 等 (2019) 分别将计划采样技术移植到 Transformer 架构。

3.2.2　序列级探索

由于翻译性能是在句子这个整体层面进行评估的，研究人员从另一个方向进行尝试，即考虑从整个句子中获得的整体奖励（如 BLEU 值），而不是词层面的匹配，从而避免直接面对该问题，这也可以被视作为一种缓解曝光偏差问题的方法。Ranzato 等 (2015) 提出了一种通过强化学习方法直接优化测试时使用的度量的序列级训练算法。 Shen 等 (2016) 提出了一种能够直接针对甚至不可微的度量来优化模型参数最小风险训练准则。 Shao 等 (2018) 提出了一种可以避免了强化学习框架的基于概率 n-gram 匹配的可微序列级训练目标的方法。 Zhang 等 (2019) 通过强制解码构建存在错误的输入馈送，以制定更好的课程学习策略。

3.3　基于内部状态融合的隐状态级方法

在本节中，我们通过对曝光偏差问题产生的直接原因进行详细分析，进而找到实际影响到的模块（即问题来源）来分析可能造成的影响。针对问题来源，我们提出通过融合生产偏差的内部状态这种隐状态级别的技术来不断减轻这种训练和测试阶段的差异，即内部状态融合（Internal States Fusion）。

3.3.1　曝光偏差问题来源分析

曝光偏差问题被定义为训练过程和测试阶段解码下一个单词的上下文不同，而上下文由源端序列表示和解码下一个 Token 时已生成的部分目标端序列两部分组成，源端序列表示通过源端输入流经编码器而来，在训练过程和测试阶段并

无差别[5]，因此该问题直接来源于作为输入馈送的两种不同的部分生成序列。因此最先接受这种在训练过程和测试阶段存在差异的输入馈送作为输入的模块便首当其冲，经过该模块后的输出的隐状态在训练过程和测试阶段也存在着偏差，并且如果这种差异没有被及时地处理，随着经过的模块的增加，存在偏差的输出单元也越来越多，该问题可能越发严重。

尤其是第一个接触这种存在差异的输入馈送的模块在整个模型中有很大影响力时，即该模块是整个模型架构中的一个重要模块，该模块差异巨大的输出势将严重影响到最终的决策结果。

3.3.2 针对问题来源的改进

本研究工作基于当前最先进的 Transformer 架构，因此解码器第一层中的掩蔽多头注意（Masked Multi-Head Attention）模块是接受产生的不同输入馈送（上下文）的第一个模块，根据 3.3.1节的分析，该模块在训练过程和测试阶段内部将会产生偏差。由于多头注意力是 Transformer 中最重要的模块之一 (Domhan, 2018)，若在此处产生的偏差未及时弥补，之后的模块均会（直接或间接地）接收到存在错误的输入，导致神经网络内部的偏差错误累积，从而对最终预测产生影响。并且我们认为从越接近问题源的模块开始减轻偏差问题，越有利于模型学得对存在错误的上下文不降低敏感性的能力。我们着力于桥接神经网络的内部单元—隐状态，即在训练过程中将完全正确和存在错误的隐状态进行不断融合，以便模型在测试阶段中更好地适应存在错误的输入（从头开始生成的部分目标端序列）。

为了获得用于桥接的内部状态，出于对训练速度的考虑[6]，我们采用二次解码的方式来分别产生完全正确和存在错误的内部状态。为了更直观的理解，我们引入了概念上的普通解码器（Vanilla Decoder）和噪声解码器（Noise Decoder）。普通解码器就是未经修改的 Transformer 解码器，而噪声解码器则在 Transformer 解码器中第一层的第一个掩蔽多头注意力模块进行了状态融合操作，即内部状态融合方法，我们将在接下来的内容中详细描述关于这部分的修改，需要注意的

[5]准确来说，训练过程中一般会通过引入 dropout 技术来防止模型过拟合，而测试阶段为了获得确定性的输出不会引入随机因素，因此这两个阶段还是存在着差异，不过这种差异的产生原因不会影响对该问题的分析。

[6] Zhang 等 (2019) 的句子级方法在训练过程中引入了解码步骤，破坏了训练过程的并行性。

是这两个解码器的参数是绑定的（tied）。

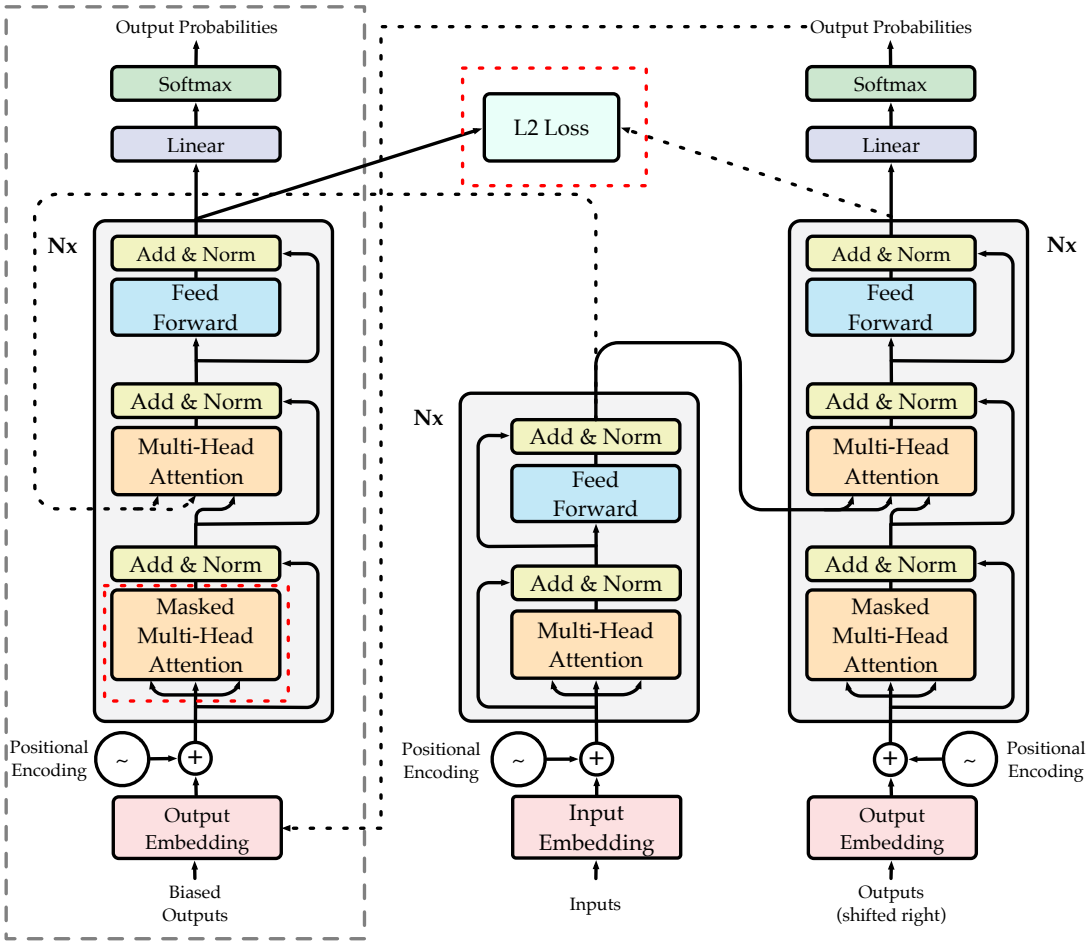

图 3.1 基于状态融合和输出矫正的模型架构

Figure 3.1 States Fusion & Output Rectification Based Model Architecture

我们依次运行普通解码器（图 3.1最右端组件）和噪声解码器（图 3.1最左端组件）。我们首先运行普通解码器，它接受完全正确的参考序列作为输入馈送，源端表示和输入馈送在经过普通解码器后能够获得曝光偏差来源的偏差状态、解码器每一层的输出和最终的预测序列。然后，我们运行噪声解码器，其输入馈送是存在错误的[7]。在这个过程中，第一层的第一个掩蔽多头注意力模块在计算放缩点积注意力（Scaled Dot-Product Attention）前将对内部状态进行更新，即将经过多头子空间映射后的子空间状态与运行普通解码器时缓存的相同级别的子空间状态逐元素地按比例进行融合[8]，如图 3.2所示。其中左侧为普通解码器中的掩蔽多头注意力模块，右侧为噪声解码器中的掩蔽多头注意力模块，“Biased

[7]为了方便起见，我们这里使用普通解码器最终的预测序列。

[8]我们仅在解码器第一层的第一个掩蔽多头注意力模块中进行子空间状态融合操作。

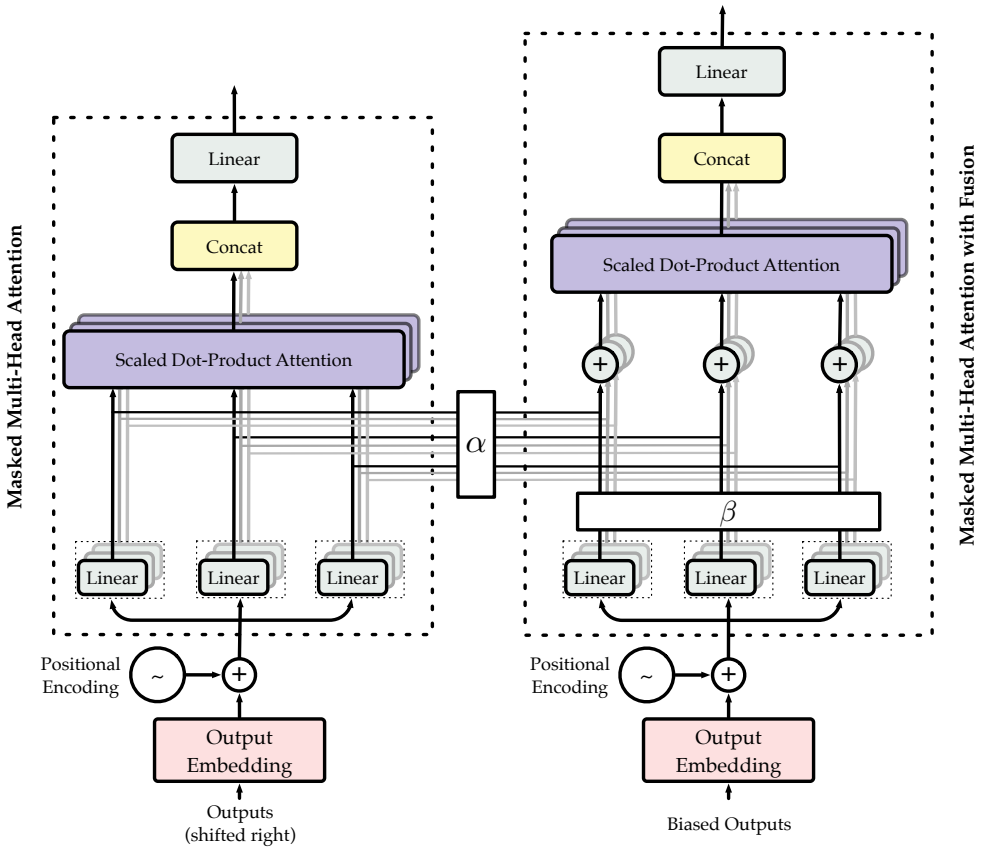

图 3.2 内部状态融合详细图例阐释

Figure 3.2 Detailed Illustration of Internal States Fusion

Outputs" 代表存在错误的输入馈送。

根据 1.2.2节的描述，整个内部状态融合过程可以形式化为如下[9]:

$$
\begin{aligned}
q^* &= e^*\mathbf{W_Q}, \quad q' = e'\mathbf{W_Q} \\
k^* &= e^*\mathbf{W_K}, \quad k' = e'\mathbf{W_K} \\
v^* &= e^*\mathbf{W_V}, \quad v' = e'\mathbf{W_V} \\
q &= \mathbf{F}(q^*, q') \\
k &= \mathbf{F}(k^*, k') \\
v &= \mathbf{F}(v^*, v') \\
o &= \mathbf{Attention}(q, k, v)\mathbf{W_O}
\end{aligned}
$$

[9]对于任何本章的变量 $\mathbf{s}$，我们将普通解码器的输入或输出表示为 $\mathbf{s}^*$，将与噪声解码器相关的输入或输出表示为 $\mathbf{s}'$。所有这些符号都是矢量类型，除非另有说明。

其中 e^* 和 e' 由下式获得:

$$e = \text{dropout}(\mathbf{emb}(\mathbf{y}) + \mathbf{pos}(\mathbf{y}))$$

这里的 **emb** 和 **pos** 分别为 Token 嵌入和位置编码模块，**y** 表示输入馈送。**F** 是一个融合函数，其形式如下:

$$\mathbf{F}(s^*, s') = \alpha \cdot s^* + \beta \cdot s'$$

其中 α 和 β 为标量系数，我们在本章的所有实验中均设置 $\beta = 1 - \alpha$，除非另有说明。

3.4　构建辅助监督信号进行增强的方法

注意到在 3.3节中引入了两个几乎完全一样的解码器对：普通解码器和噪声解码器，它们在宏观上的区别在于接受了不同的输入馈送，即来自训练样本的完全正确的和构建的存在错误的部分目标端序列。本节，我们进一步利用来接受完全正确的输入的自普通解码器的每一层输出的特征作为辅助监督信号，并根据特征与噪声源之间的距离对来自噪声解码器的相应输出施加不同强度的约束，这被称为逐层输出矫正（Layer-wise Output Rectification，**LOR**）。

3.4.1　训练过程中监督信号分析

有监督的神经机器翻译模型一般根据交叉熵（Cross Entropy）准则进行训练。具体地，该准则收集预测正确的 Token 对应的对数概率（log probabilities）并求，然后通过基于梯度下降的方法进行优化。然而，在解码器中，从输入馈送到最终预测间构成了一条非常长的路径，如果将该路径展开，将是一个很深的神经网络结构。一方面深度可以增强网络的表达能力，但另一方面输入在流经多个解码器块和输出层产生最终预测结果，与给定的正确结果进行对比计算损失求出梯度后，由于路径太长，来自顶部输出层的监督信号会变得太弱以至于无法很好地反向传播到底部的组件中。这是训练深度神经网络必然会遇到的问题，如果能有效地利用某些中间的输出作为辅助监督信号将可以一定程度地改善这个窘境。很显然，我们在 3.3节中已经获得了构建辅助监督信号的条件。

3.4.2 利用解码器对构建辅助监督信号

在运行二次解码的过程中，我们能够获得任意位置的输出，其中就包括解码器每一层输出的特征。注意到普通解码器运行时接受的输入是完全正确的，但噪声解码器的输入可能存在错误，因此从普通解码器得到的输出相对来说会比从噪声解码器得到的结果更接近理想状况。受知识蒸馏（Knowledge Distillation）(Hinton 等, 2015) 形式上的启发，我们提出进一步利用从普通解码器中得到的这些内部状态作为辅助监督信号（教师）来指导处于相同层次的噪声解码器状态（学生）。因此，我们利用普通解码器的每层的输出特征来约束噪声解码器的相应位置的输出。具体地说，我们通过 L2 损失来迫使噪声解码器的输出接近于相同层次普通解码器的输出。形式化地，若令 h_i^* 和 h_i' 分别表示普通解码器和噪声解码器的第 i 层的输出特征，其中 i 为标量，则对第 i 层施加约束计算损失的公式如下:

$$loss_i = \|h_i^* - h_i'\|_2$$

此外，由于每层的特征输出相对于曝光偏差问题源的距离不同，我们推测其偏离理想情况的程度也不同。经验上，大量的参数能够起到一个对两种差异较大状态进行缓冲的作用，因而离问题源越近的地方，产生的偏差程度越大，而离问题源越远的地方，随着大量参数的引入，这种偏差间隙能够被逐步的减小。因此我们对不同的层输出的特征损失施加不同强度的约束，具体来说，偏差程度越大，约束强度越小。我们通过为每个层输出特征计算的 L2 损失分配精细的权重来实现这种机制，每层输出特征的 L2 损失对应的权重由以下双重指数方式计算得到:

$$\tilde{w}_i = a \cdot 2^{i-1}$$

$$w_i = \exp(\tilde{w}_i)$$

上式中所有变量均为标量，a 是第一重指数的放缩系数，$\tilde{w}_i$ 为候选权重，w_i 是分配给第 i 层的权重。因为我们为每层分配的权重不同且按层数单调变化，故称这种方法为逐层输出矫正。

3.5　模型架构与训练目标

综合 3.3节和 3.4节所述，我们的整体模型架构如图 3.1所示。在图 3.1中，从左到右分别是噪声解码器、编码器和普通解码器。带虚线的组件是本项工作引入的新组件，其余组件为 Transformer 原始架构，我们对修改的部分用红色虚线框进行了标记。从图中可以看出，噪声解码器与最右侧的普通解码器在架构上完全一致，除了在噪声解码器的 “Masked Multi-Head Attention” 被红色虚线框进行了标记，而这个部分正是 3.3节中介绍的内部状态融合发生的位置，噪声编码器和普通编码器均为概念上的组件，它们始终共享参数，另一个被标记的修改 “L2 Loss” 便是 3.4节中的逐层输出矫正方法图示。

本工作的整体训练有两部分构成：基于最大似然估计 **MLE** 用于拟合数据的准则和利用辅助监督信号对各层输出特征进行约束的 **LOR** 正则项，整体训练目标为以下公式:

$$\mathcal{L}(\Theta) = \frac{1}{T}\left(\mathcal{L}_{\text{MLE}}(\Theta) + \gamma \cdot \mathcal{L}_{\text{LOR}}(\Theta)\right)$$

$$\mathcal{L}_{\text{MLE}}(\Theta) = -\sum_{m=1}^{N}\sum_{i=1}^{T_y} \log P(y_i^m | \mathbf{y}_{<i}^m, \mathbf{x}^m; \Theta)$$

$$\mathcal{L}_{\text{LOR}}(\Theta) = \sum_{m=1}^{N}\sum_{i=1}^{D} w_i \cdot \|h_i^{m*} - h_i^{m\prime}\|_2$$

其中 T 表示 Token 的数量，N 表示训练语料库中的训练样本数量，D 为解码器堆叠的层数，γ 是一个用于整体平衡 **LOR** 强度和传统的 **MLE** 的系数，它们均为标量。

3.6　实验结果与分析

我们也在多个数据集的多个语言对上进行了实验，实验结果表明，本项工作提出的方法有效。在本节中，我们将依次介绍实验所使用的数据集、用于对比的基线系统、运行的参数设置、二次训练的方法，然后列出实验结果并进行分析。由于整个方法分为两个部分，并引入了噪音从已经收敛的模型开始训练，我们通过屏蔽其中的一个进行消融实验来分析其在整体方法中的作用。最后，我们通过展示示例来从侧面说明相对于基线系统的改进。

3.6.1 实验数据

我们采用与 2.4.1节相同的实验数据集进行实验，即 608K 英语 ⇔ 罗马尼亚语（En⇔Ro）、1.25M 中文 ⇒ 英语（Zh⇒En）以及 4.5M 英语 ⇒ 德语（En⇒De）数据集。

3.6.2 对比基线

我们也使用1.2.2节中提到的当前最先进的 Transformer 结构作为基线系统。

3.6.3 运行配置

实验前，我们对所有数据集按照 2.4.3节的方法进行了预处理。对于 Zh⇒En 数据集上的实验，我们将融合系数 α 设置为 0.1，而其它的设置为 0.4[10]。我们将所有实验的第一重指数的放缩系数 a 设置为 0.1，每层权重随层数变化的变化曲线如图 3.3所示，横轴为层数，纵轴为权重。**LOR** 平衡系数 γ 设置为 0.001，而对于 En⇒De 数据集为 0[11]。

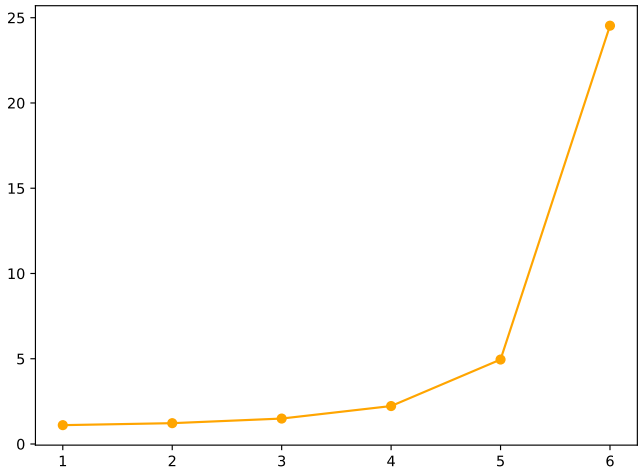

图 3.3 双重指数约束强度曲线

Figure 3.3 Double Exponential Constraint Strength Curve

3.6.4 实验结果

对于所有的 base 模型，我们还是将模型训练至收敛，然后报告在验证集上单模型取得的最高的 BLEU 值的那组结果。对于 Zh⇒En 数据集上的结果，我们

[10]非训练数据集仅含单个参考译文，为保持结果稳定，我们在融合状态时将完全正确的比例调大。

[11]因收敛的模型所经历过的训练步骤数较多，且模型在该数据集上的预测准确率相对其他的数据集来说要低，导致监督信号较弱甚至于无效。

报告大小写不敏感的 BLEU 值，对于其它数据集则报告大小写敏感的 BLEU 值。同 2.4.5节一样，“⇑” 表示结果显著好于基线 ($p < 0.01$)，而 “↑” 表明 $p < 0.05$。所有显著性检验也都是通过使用成对自助重采样（paired bootstrap resampling）(Koehn, 2004) 来测量的，其中实验次数为 1000，采样比例为 0.5。

#	System	Architecture	MT02	MT03	MT04	MT05	MT06	MT08	AVG
Existing NMT Systems (single model)									
1	Feng 等 (2019)	Transformer (base)	-	44.74	46.27	44.16	43.29	34.72	42.63
2	*this work*	Transformer (base)	45.33	44.10	45.99	44.76	44.02	34.59	42.69
3		Fusion & Rectification	**46.86**⇑	**45.87**⇑	**47.52**⇑	**46.38**⇑	**45.34**⇑	**36.04**⇑	**44.23**(+1.54)
Existing NMT Systems (average model)									
4	Zhang 等 (2019)	Transformer (base)	-	46.89	47.88	47.40	46.66	-	47.21
5	Zhang 等 (2019)	Transformer (wo)	-	47.42	48.34	47.89	47.34	-	47.75 (+0.54)
6	Zhang 等 (2019)	Transformer (so)	-	48.31	49.40	48.72	48.45	-	48.72 (+1.51)
7	*this work*	Transformer (base)	47.26	46.43	47.72	46.93	46.31	-	46.84
8		Fusion & Rectification	47.99	47.02	48.29	47.27	47.04	-	47.41 (+0.57)

表 3.1 NIST Zh⇒En 数据集翻译任务实验结果

Table 3.1 Results on the NIST Zh⇒En Translation Task

表 3.1列出了 NIST Zh⇒En 验证集和测试集上的实验结果。第 1 组数据为文献中可比较的基线结果[12]。由于我们提出的方法不改变模型结构组织，基线模型和改进模型拥有相同的参数量，91.0M。表 3.1第 2、3 组实验结果表明，通过应用我们提出的两种方法，模型在能够取得显著的性能提升。从 5-8 组实验数据来看，应用本方法取得的提升与 Zhang 等 (2019) 的词级方法取得的提升大致相当，但不及其句子级方法。

表 3.2中报告了我们的方法在小规模的 En⇔Ro 和大规模的 En⇒De WMT16 公开数据集上的实验结果。表 3.2在其它数据集上的结果也均优于基线，表明我们的方法能够在多种规模的多个数据集的多种语言对上取得一致的性能提升[13]。

虽然在 Zh⇒En 数据集上性能提升显著，但在 WMT16 数据集（En⇔Ro，En⇒De）上仅表现出微弱的优势。我们推测这种差距是由于解码器权重绑定 (Press

[12]表 3.1中引用结果 Feng 等 (2019)， Zhang 等 (2019) 来自同一个研究组。为了与 Zhang 等 (2019) 进行对比，我们也对模型进行了平均 (5 个模型) 并使用 *mteval-v11b.pl* 脚本进行评分，略微不同的是：我们未限制训练样本长度，他们使用 30K 的词表。

[13]表 3.2中，“-” 表明没有提供结果或该项结果不具有可比较性。红色标记的数据表明结果是通过多模型集成（ensemble）得到的。

#	System	En⇒Ro	Ro⇒En	En⇒De
	Existing NMT Systems			
1	Lee 等 (2018)	32.40	32.06	-
2	Vaswani 等 (2017)	-	-	27.30
3	Feng 等 (2019)	32.85	-	27.21
4	Zhang 等 (2019) #BASE	-	-	27.34
5	Zhang 等 (2019) #SS	-	-	28.05
6	Zhang 等 (2019) #OR	-	-	28.65
	Our NMT Systems			
7	baseline	33.20	32.82	27.26
8	*this work*	**33.22**	**33.22**$^{\uparrow}$	**27.91**$^{\Uparrow}$
9	#4 -Decoder WT	32.21	31.88	-
10	*this work*	**33.24**$^{\Uparrow}$	**32.72**$^{\Uparrow}$	

表 3.2 WMT16 En⇔Ro、En⇒De 数据集上的实验结果

Table 3.2 Results on the WMT16 En⇔Ro & En⇒De Translation Tasks

和 Wolf, 2017) 技巧的应用导致的，即解码器的 Token 嵌入矩阵和进行 softmax 归一前的线性层的权重是共享的。我们提出从曝光偏差问题的源头上减轻问题，这个技巧的应用破坏了这个条件，因为这样和偏差源直接相关的模块与直接进行结果预测的模块之间存在着短路径联系，即曝光偏差源直接、迅速地影响到了模型。我们在摒弃了这个技巧[14]后，进一步在 En⇔Ro 数据集上进行了实验。表 3.2中的第 9、10 行的几组对比数据证实了我们的推测。在这种情况下，我们通过观察训练过程中的损失曲线发现：Zh⇒En 和 En⇔Ro 都是容易使模型陷入过拟合的数据集，缓解曝光偏差问题可以让模型得到较大幅度的性能提升。

3.6.5 消融实验

为了深入探究 3.3节、3.4节方法及其它因素对整体模型的影响，我们设计并进行了消融实验，并给出了主要的实验结果，如表 3.3所示。带缩进的 **Description** 表示我们是在最近的非缩进基础上进行的实验。“-” 代表我们在训练过程中从整体模型中移除该部分。“noise input” 是指我们通过对目标序列应用概率为预测错误率的词丢弃技术来构建存在错误的输入馈送，而不是采用普通解码器最终预测。“from scratch” 表明应用我们的方法完全重头开始训练。“geometric mean” 表示我们使用几何平均方式融合内部状态，即 $\alpha = \beta = \frac{1}{\sqrt{2}}$。

[14]表 3.2中，“-Decoder WT” 指我们不使用解码器权重绑定技巧。

首先，为了评估整体方法中不同部分的重要性，我们屏蔽了模型的每个部分，并测试了在 Zh⇒En 数据集翻译任务上的性能变化。表 3.3中第 1、2、3、4 组的结果表明，我们的内部状态融合方法能够独立工作，而分层输出矫正的应用进一步提高了翻译质量。然后，为了消除引入噪声的相关技术（词丢弃）可能带来的正则化影响，我们设计在没有梯度累积的境中运行解码器，然后以此构建存在错误的输入馈再次送到解码器中进行训练。表 3.3中的第 5 组数据与基线结果（第 1 组数据）相比，表明纯粹地而没有按照一定策略的地引入噪音无助于提升翻译性能。我们还尝试了以另一种方式构建含有错误的上下文（输入馈送），即简单地将按照模型当前时刻动态预测的错误率对训练样本的目标端应用词丢弃技术。第 6 组数据相对基线结果也有明显提升，该实验结果表明，探索其他构建含有错误的上下文的方法是有意义的。此外，从第 7 组实验结果可以看出，即使从头开始重新训练模型， 3.3节的方法也能够改善翻译质量。第 8 组数据表明状态的融合方式不仅限于线性插值。

#	Description	MT02	MT03	MT04	MT05	MT06	MT08	AVG	Δ
1	basline	45.33	44.10	45.99	44.76	44.02	34.59	42.69	
2	*this work*	46.86	45.87	47.52	46.38	45.34	36.04	44.23	**+1.54**
3	-fusion	37.54	34.93	38.42	34.93	34.95	25.33	33.71	–8.98
4	-rectification	45.84	45.73	46.72	45.53	44.74	35.24	43.59	+0.90
5	-fusion & rectification	43.75	41.77	44.15	41.75	41.87	32.91	40.49	–2.20
6	noise input	46.48	45.64	46.87	46.65	44.89	35.99	44.01	+1.32
7	from scratch	45.56	44.89	46.38	45.14	43.93	36.17	43.30	+0.61
8	geometric mean	46.14	45.21	46.95	45.14	44.69	35.72	43.54	+0.85

表 3.3 NIST Zh⇒En 数据集翻译任务消融实验结果

Table 3.3 Ablation Study Results on the NIST Zh⇒En Translation Task

由于我们的模型引入了二次解码的模式，训练速度相对原始 Transformer 模型有所降低，在同样的配置下遍历完整个数据集的时间为原来的 1.5 倍。

3.6.6　案例分析

表 3.4展示了在 NIST Zh⇒En 验证数据集 MT02 上分别由基线模型（BASE）和应用了内部状态融合与分层输出矫的模型（SFOR）翻译的两个示例，这两个模型在整个验证集上的 BLEU 值分别为 45.33 和 46.86。“src” 为源端句子，“ref1”，

“ref2”，“ref3”，“ref4” 则代表源端句子的黄金参考，即 4 个参考译文。用**红色**标记的内容为源端短语及其对应的翻译，而用**蓝色**标记的部分表明其已经偏离了源端短语。**橙色**的字体代表那些应该被明确翻译出来以忠实于原文的词语。

给出的两个示例均表明，我们的模型给出了更准确和忠实的翻译。第一个例子中，虽然 “**pull out**” 在语义上等同于 “**military retain**”，但这种有偏差的上下文会导致翻译不足，即 “**两 城镇**” 应该被显式地翻译为 “**two towns**” 或 “**two cities**”。第二个例子中，所有包含了时间状语 “上周四” 的黄金参考都将其放在 “and” 前一部分的末尾。然而基线模型将它翻译在句子的开头，这种非预期的顺序导致源短语 “**北京 西班牙 使馆**” 被错误地翻译成了 “**the chinese embassy in spain**”，本意应为 “**the spanish embassy in beijing**”，这是完全错误的。此外，“**坚决**” 应该被翻译成类似于 “**resolutely**” 或 “**firmly**” 之类的词来强调原句子的主语所持的态度。我们把这两种不忠实原文的翻译现象的原因归结为由曝光偏差导致的累积错误。

315	src	白宫 称 以色列 从 西岸 **两 城镇** **撤军** 仅 是 开始
	ref1	white house says israeli **military withdrawal** from **two** west bank **cities** only a beginning
	ref2	white house welcomes israel ’s first **withdrawal** from west bank
	ref3	white house : israel ’s **withdrawal of troops** from **two cities in** west bank only a beginning
	ref4	the white house call israeli **withdrawal** from **2 town in the** west bank a beginning
	BASE	white house says israeli **pullout** from the west bank only starts
	SFOR	white house says israel ’s **military withdrawal** from **two** west bank **towns** only starts
664	src	这 批 朝鲜人 上 星期四 闯入 **北京 西班牙 使馆** ， **坚决** 表明 要 投奔 韩国 。
	ref1	the group of north koreans stormed **the spanish embassy in china** last thursday and **resolutely** expressed their intention to go to south korea .
	ref2	this group of north koreans has intruded **the spanish embassy in beijing** and they **asked to be** allowed to go to south korea .
	ref3	these dpr koreans broke into **the spanish embassy in beijing** last thursday saying they wanted to go to republic of korea .
	ref4	these north koreans broke into **the spanish embassy to beijing** last thursday , **firmly** indicating that they wanted to seek asylum in the south .
	BASE	last thursday , the korean people intruded into **the chinese embassy in spain** and made clear their intention to go to korea .
	SFOR	this group of north korean people intruded into **the spanish embassy in beijing** last thursday and **resolutely** indicated they would go to korea .

表 3.4 NIST Zh⇒En 验证集 MT02 上的曝光偏差翻译样例

Table 3.4 Translation Examples of Exposure Bias from MT02 on NIST Zh⇒En

3.7 与相关工作及引入噪音工作的区别和联系

和 Mihaylova 和 Martins (2019) 相比，我们的方法能够在没有精心设计的课程学习策略的情况下工作，并实现一致的性能提升。和 Zhang 等 (2019) 相比，我们的方法能够在不改变训练安排的情况下对已收敛的模型进行增强，并且不破坏训练过程的并行性，对噪声的敏感性不强。

为了弥补训练和测试之间解码上下文的差距，在训练阶段引入噪声是必须

的，也是不可避免的，因为解码确实会出错。据我们所知致力于增强鲁棒性及基于生成对抗网络（Generative Adversarial Networks，GAN）(Goodfellow 等, 2016)的神经机器翻译模型工作均引入了噪音因素。但是本项工作引入噪音的动机与做法上和它们有很大的不同：

- 致力于增强鲁棒性的工作引入噪音来抵抗含噪音的输入（源端具有小的扰动），一般在训练过程中对**源端**进行模拟噪音的操作，以期模型相对于无扰动的情形，在源端输入存在微小扰动时得到的译文质量不会剧烈抖动；
- 基于生成对抗网络的工作使用它来**构建用于训练的对抗样本**，增强模型对对抗样本的区分能力；
- 我们在**目标端**进行加噪操作：显式地引入简单的噪音或隐式地利用模型做出的预测来构建含有错误的输入馈送。

事实上，我们的方法可以增强神经机器翻译模型的鲁棒性。

总的来说，出于我们最初对曝光偏差问题先进行了扩展后再提出改进方法的考量，本项工作提出的方法与差异的形式及最终的度量标准无关，且操作简单易于实现。

3.8　本章小结

在这项工作中，我们提出了两种方法：内部状态融合和逐层输出矫正，以解决融合模型中的曝光偏差问题。第一种方法通过融合曝光偏差产生的内部状态来减少训练和测试之间的差异。同时，我们利用输入馈送为完全正确的目标端的普通解码器产生的内部状态作为辅助监督信号来按照离问题源的远近来约束噪声解码器相同层次的输出。由于我们只对静态的系数进行了实验，我们推测根据模型的能力动态地去调节它们可能会获得更好的结果，这些方法有待进一步探究。此外，由于训练阶段和测试阶段观测的数据始终是不同的，曝光偏差问题始终存在，本方法仅从一种细粒度的隐状态层次进行了探究。

第 4 章　基于 n 元文法匹配精度的曝光偏差性能评估方案

4.1　引言

第 3章中提到，端到端的神经机器翻译模型训练过程和测试阶段提供给模型进行预测下一个 Token 的上下文不一致，这种曝光偏差问题可能会造成测试时的错误累积，导致性能下降。因此很多致力于缓解曝光偏差问题带的工作主要以模型性能（如 BLEU 值）在新的方法下是否（显著）提升来衡量方法的有效性。然而，至少在机器翻译中，曝光偏差问题只相当于模型性能下降的一个充分条件，即曝光偏差问题几乎肯定会影响到模型的性能 Bengio 等 (2015)，并且在大多数情况下会令模型性能下降。新方法的应用使得模型的性能得到了提升仅说明方法在提升机器翻译性能方面有效，但并不能简单地将其归因于由于缓解了曝光偏差问题。

3.1节分析曝光偏差问题时指出，解码时将已经生成的部分序列作为条件，按照每次预测一个 Token 来生成完整的序列的方式，使得曝光偏差问题几乎不可避免，因为在训练过程中，出于提供更强的监督信号来提升模型性能和加速训练的考虑，即使以同样的方式进行预测，我们提供的上下文倾向于包含更多正确的信息[1]。曝光偏差问题出现的必然性及其严重影响模型性能的事实使得提出缓解方法的研究势在必行。与此同时，对于一个问题来说，评估方法是否解决了该问题、如何调试中间结果以及最终如何定性（甚至定量）地衡量解决的程度亦不容忽视：如果无法很好的评估，则对该问题的研究将会有所掣肘。

在上一章我们已经对前人为缓解曝光偏差问题而做的相关工作进行了整体回顾，并且也提出了一种能够缓解曝光偏差问题提升性能的方法。本项工作中，我们关注如何合理评估方法对解决曝光偏差问题是否有效以及如何定性地衡量解决程度，具体地：

- 如果应用了新方法的翻译系统在测试集上的性能相对基线有提升，我们能相信这是由于缓解了曝光偏差问题带来的吗？
- 如果多个翻译系统在测试集上的性能相差不大，我们如何选出**抗曝光偏**

[1]除非我们在训练过程中测试阶段完全一致，即只使用源端序列，这在有监督的神经机器翻译训练方法中显然是不可取的。

差能力（anti-exposure bias capability）[2]更强的系统?

为了回答以上问题，我们通过构建存在错误的上下文后将系统运行在评估模式来获取基于 n 元文法匹配精度的统计量，并对第 3章的方法在不同规模的多个测试集的多个语言对上的模型进行了验证。实验结果表明，相比于基线翻译系统，改进后翻译系统的性能变化及显著性与其抗曝光偏差能力的联系如下：

- 若改进方法对缓解曝光偏差问题有效，则改进后的模型性能越高，抗曝光偏差能力的提升越稳定；
- 翻译系统性能提升的显著性与抗曝光偏差能力的提升的显著与否无必然联系；
- 改进方法对于不同的参考集，得到的显著性结论不一定一致；
- 改进方法对提升各阶 n 元文法的精度的步调不一定一致。

在本章接下来的内容中，我们将首先介绍前人在分析验证方法对缓解曝光偏差问题有效性方面的相关工作。然后指出他们的验证方法中存在的问题，并就此提出合理的改进方法。接着，我们对整体评估方案进行介绍：包括如何对数据集进行处理、具体的操作方法、需要收集的统计量、如何对统计量进行加工转化为最终的单一数值指标，最后如何根据以上步骤得到的指标下结论。我们按照以上方案对多个系统（不同规模、多个数据集、多个语言对）进行了实验，并对结果进行了深入分析。最后，我们对本项工作进行了总结。

4.2 相关工作介绍

现有的这方面的工作通常通过以端到端的机器翻译性能得到了提升来说明缓解了曝光偏差问题，即研究人员通常通过 BLEU 提高这个结果来证明他们的方法缓解了曝光偏差问题，然而这种说明不具有很强的说服力。因为一方面 BLEU 这个指标是为翻译质量评估而设计的，令一方面虽然曝光偏差问题会使机器翻译性能下降，但是并不能肯定应用了新方法得到的提升一定是由于缓解该问题所带来的效益。 Zhang 等 (2019) 在 “Effect on Exposure Bias” 节称他们通过额外的实验验证了性能提升主要是通过解决了曝光偏差问题获得的，具体步骤如下:

- 从 Zh⇒En 翻译任务的训练数据集中随机选择 1K 个句子对；
- 对他们提出的模型和基线模型分别解码 1K 个句子对中的源端以得到预

[2]我们将抗曝光偏差能力定义为：翻译系统在给定的由测试集构建而存在错误的上下文的预测精度。

测的概率分布；

- 统计所有符合如下条件的词的个数 N：目标端中词的概率值在由他们提出的模型生成的概率分布中大于相应的在基线模型生成的概率分布中的；
- 用 C 表示目标端词的总数，计算比率 N/C；

他们将得到的结果，65.06%，作为相应的论据。显然，这个说法是存在争议的，至少从以下两个方面来看他们所验证结论的论据不够充分：

- 他们在训练数据集上进行实验，而训练数据集是提供给模型进行学习的，不适合用于评估；
- 解码时，一般通过 *argmax* 函数选出正确的词，即概率分布中概率值最大那项对应的词，也正因如此预测出正确的词只需要相对的概率即可，与概率值的绝对大小无关[3]。

此外，尽管该数值超过了 0.5，但并没有严格的统计意义，籍此得到的比率不能让我们对曝光偏差的内部细节有更多的理解，基本无法用于调试目的。

总的来说，当前没有还没有一个较好的方案能够用来证明机器翻译性能的提升是由缓解了曝光偏差问题带来的，遑论获知关于曝光偏差问题的内部细节。

4.3　基于 n 元文法匹配精度的性能评估方案

前面提到，如果不能较好证明是由于缓解了曝光偏差问题带来了机器翻译性能的提升，则无从获悉对曝光偏差问题的内部细节，而最终不利于对曝光偏差问题的进一步研究。因此在本节中我们开发了一套评估神经翻译系统[4]在抗曝光偏差方面的能力的完整解决方案：从数据集预处理到如何判定结果。

4.3.1　测试环境准备

由于曝光偏差问题源于训练过程中和解码阶段模型预测下一个词所基于的上下文不一致，我们提出通过运行在一种介于训练和测试之间的环境[5]之上的过

[3] Pereyra 等 (2017) 指出，惩罚高置信度的输出分布可以对神经网络起到正则化的效果，因而能进一步的增强神经网络的泛化性能。

[4]解码时，源端序列及已经部分生成了的目标端序列均可以获得，因此能够作为预测下一个的单词的上下文的一部分，本项工作以当前主流的神经机器翻译系统为例进行详细说明。

[5]即使模型在训练过程中对这两种环境进行了磨合，在模型性能提升不大的情况下，训练过程和解码阶段模型预测下一个词的上下文信息依然是严重不对等的，即由于在测试阶段预测的精确度相对更低将导致上下文中的部分生成序列存在严重的错误累积现象，进而无法准确地得出新的方法是否有助于缓解曝光偏

程来评估模型在抗曝光偏差方面的能力是否有提升。为了更好地评估模型的抗曝光偏差能力，即评测模型在给定的含有相同错误的上下文的预测能力，我们提出构造上下文错误程度可控的输入馈送。我们使用独立于训练数据集的测试数据集作为构建基础。形式化地，对于测试数据集 $\mathbf{D}^*$，我们对测试集中任何一个黄金参考 $\mathbf{y}^*$ 以不同的概率应用词丢弃技术来构建存在错误的上下文的数据集 $\mathbf{D}'$：

$$\mathbf{y}^* = \{y_1^*, y_2^*, y_3^*, y_4^*, \cdots, y_{T_y}^*\}$$

其中，“y_i^*” 表示黄金参考中第 i 个 Token。通过词丢弃，黄金参考 $\mathbf{y}^*$ 相应地被转换为 $\mathbf{y}'$：

$$\mathbf{y}' = \{y_1^*, \square, y_3^*, \square, \cdots, y_{T_y}^*\}$$

“□” 代表该位置的词被随机地替换成了 UNK 符号。注意，被替换的位置是随机的，以上例子仅用作举例。

4.3.2 运行对比系统

我们在 4.3.1节构建的存在错误的测试数据集 $\mathbf{D}'$ 的基础上进一步构建含有错误的上下文。训练过程中，我们对句子对 $(\mathbf{x}, \mathbf{y}^*)$ 进行训练时，预测第 t 个 Token 的上下文为源端序列（句子）$\mathbf{x}$ 和已经部分生成了的序列 $\mathbf{y}_{<\mathbf{t}}^*$，具体地：

$$\mathbf{y}_{<t}^* = \{\#, y_1^*, y_2^*, y_3^*, y_4^*, \cdots, y_{t-1}^*\}$$

其中 $t \in [1, T_y]$，“#” 代表序列起始符 **BOS**。因此在数据集 $\mathbf{D}'$ 上，预测 t 个 Token 的上下文为源端序列 $\mathbf{x}$ 和已经部分生成了的序列 $\mathbf{y}'_{<\mathbf{t}}$：

$$\mathbf{y}'_{<t} = \{\#, y_1^*, \square, y_3^*, \square, \cdots, y_{t-1}^*\}$$

为了保证模型能力不发生变化，我们以评估模式（不进行参数更新）在数据集 $\mathbf{D}'$ 上运行训练程序，此时模型预测下一个 Token 的上下文所含的错误率便可人为控制，即由数据集 $\mathbf{D}'$ 决定。根据我们对早先实验过程和结果的分析，我们在 4.3.1节中构建存在错误的数据集 $\mathbf{D}'$ 时，对词丢弃的不确定性概率 p 分别设置为 $p = 0.0$，$p = 0.05, p = 0.10, p = 0.15$ 和 $p = 0.20$。我们分别对改进模型和基线模型按照相同的程序在数据集 $\mathbf{D}'$ 上进行测试。

差问题的结论。

4.3.3　收集统计量

我们主要收集收敛的基线模型和改进模型按照 4.3.2节的方法预测出的序列并以此计算统计量。以含有 N 个样本的单黄金参考测试集为例，对于第 m 个测试样本 $(\mathbf{x}_m, \mathbf{y}'_m)$，若将基线模型基于该样本的预测结果记作 $\hat{\mathbf{y}}_m$，改进模型基于该样本的预测结果记作 $\tilde{\mathbf{y}}_m$，则我们分别统计它们相对于输入馈送 $\mathbf{y}'_m$ 和黄金参考 $\mathbf{y}^*_m$ 的有效 n 元文法数量和匹配的 n 元文法数量。具体来说，我们将对每一个测试样本统计以下四个元组的有效 n 元文法数量和匹配的 n 元文法数量：

- $(\mathbf{y}', \hat{\mathbf{y}})$：基线模型预测结果相对于输入馈送的；
- $(\mathbf{y}^*, \hat{\mathbf{y}})$：基线模型预测结果相对于黄金参考的；
- $(\mathbf{y}', \tilde{\mathbf{y}})$：改进模型预测结果相对于输入馈送的；
- $(\mathbf{y}^*, \tilde{\mathbf{y}})$：改进模型预测结果相对于黄金参考的。

最后我们在整个测试数据集上计算微平均的 n 元文法匹配精确度（以下称 n 元文法匹配精度）。以元组 $(\mathbf{y}', \tilde{\mathbf{y}})$ 为例，若统计得到的第 m 个样本的有效 n 元文法数量和匹配的 n 元文法数量分别为 $\#total_m^{(\mathbf{y}', \tilde{\mathbf{y}})}$、$\#matched_m^{(\mathbf{y}', \tilde{\mathbf{y}})}$，则该元组在整个测试数据集上的 n 元文法匹配精度的计算方式如下：

$$P^{(\mathbf{y}', \tilde{\mathbf{y}})} = \frac{\sum_{m=1}^{N} \#matched_m^{(\mathbf{y}', \tilde{\mathbf{y}})}}{\sum_{m=1}^{N} \#total_m^{(\mathbf{y}', \tilde{\mathbf{y}})}}$$

我们通过算法 **ValidAndMatchedNGramsStatProg**($\mathbf{seq_1}$, $\mathbf{seq_2}$, n) 来执行以上统计操作，即统计一对序列中有效 n 元文法数量及匹配的 n 元文法数量，详细过程如算法 2所示。针对该算法，我们首先对输入参数作如下几点说明：

1. 参考序列 $\mathbf{seq_1}$ 的第一个 Token 不能是序列起始符号 **BOS**；
2. 参考序列 $\mathbf{seq_1}$ 中必须包含且仅能包含一个序列结束符号 **EOS**；
3. 预测序列 $\mathbf{seq_2}$ 和参考序列 $\mathbf{seq_1}$ 的长度相同；
4. 序列对（$\mathbf{seq_1}$，$\mathbf{seq_2}$）的有效长度为参考序列 $\mathbf{seq_1}$ 中 **EOS** 之前部分（含 **EOS**）[6]的长度，记作 $length$；

然后，我们对算法中加粗的文字进行说明：

- **SubSequence**($\mathbf{seq_1}$, $start$, end) 为截取子序列操作，即截取 $\mathbf{seq_1}$ 的第 $start$ 到 end 个 Token，包含端点位置；

[6]训练过程中，我们对 **EOS** 的预测结果也纳入统计，因为若未终结，则预测序列的长度实际上更长，实际已经偏离了参考序列。

算法 2 ValidAndMatchedNGramsStatProg($\mathbf{seq_1}$, $\mathbf{seq_2}$, n)

Require: 参考序列 $\mathbf{seq_1}$，预测序列 $\mathbf{seq_2}$，待统计的 n 元文法阶数 n

1: $\mathbf{seq_1} = \mathbf{SubSequence}(\mathbf{seq_1}, 1, length)$;

2: $\mathbf{seq_2} = \mathbf{SubSequence}(\mathbf{seq_2}, 1, length)$;

3: $\mathbf{seq_1} = \mathbf{RefineSeq}(\mathbf{seq_1})$;

4: $\mathbf{seq_2} = \mathbf{RefineSeq}(\mathbf{seq_2})$;

5: $\mathbf{counter_1} = \mathbf{NGramCounter}(\mathbf{seq_1}, \mathbf{n})$;

6: $\mathbf{counter_2} = \mathbf{NGramCounter}(\mathbf{seq_2}, \mathbf{n})$;

7: $total = \sum_{key \in \mathbf{Keys(counter_1)}} \mathbf{Value}(\mathbf{counter_1}, key)$;

8: $\mathbf{keys} = \mathbf{Keys}(\mathbf{counter_1})$ & $\mathbf{Keys}(\mathbf{counter_2})$;

9: $matched = \sum_{key \in \mathbf{keys}} \mathbf{Min}\Big(\mathbf{Value}(\mathbf{counter_1}, key), \mathbf{Value}(\mathbf{counter_2}, key)\Big)$;

10: Return $total$, $matched$;

- **RefineSeq** 为对序列进行修正的程序，本项工作将 **RefineSeq** 实现为移除序列中的填充符号；
- $\mathbf{counter_1}$、$\mathbf{counter_2}$ 为映射数据结构的 n 元文法计数器，该数据结构中 n 元文法作为键，相应的计数作为值；
- $\mathbf{NGramCounter}(\mathbf{seq_1}, \mathbf{n})$ 是对序列 $\mathbf{seq_1}$ 的 n 阶 n 元文法进行统计并构建成映射结构数据的程序；
- $\mathbf{Value}(\mathbf{counter_1}, key)$ 为获取映射结构数据 $\mathbf{counter_1}$ 中键 key 对应值的操作；
- $\mathbf{Keys}(\mathbf{counter_1})$ 为返回映射结构数据 $\mathbf{counter_1}$ 中所有键的操作；
- **keys** 为两个 n 元文法计数器 $\mathbf{counter_1}, \mathbf{counter_2}$ 键的交集，即共同的 n 阶 n 元文法；
- “&” 表示求交集操作；
- $\mathbf{Min}(a, b)$ 为返回 a，b 两者较小值的程序。

此外，剩余的斜体变量代表的含义如下：

- $total$：n 元文法计数器 $\mathbf{counter_1}$ 中 n 阶 n 元文法总数，即有效 n 元文法数量；
- $matched$：预测序列匹配到的参考序列的 n 阶 n 元文法数量，即匹配的 n 元文法数量。

由于本节引入了大量的符号，为了更为直观的理解，我们最后进行举例说明。例如，我们将测试样本 $(\mathbf{x}, \mathbf{y}^*)$ 以不确定性 $p = 0.20$ 构造成存在错误的样本 $(\mathbf{x}, \mathbf{y}')$：

$$\mathbf{y}^* = \{y_1^*, y_2^*, y_3^*, y_4^*, y_5^*, y_6^*, y_7^*, y_8^*, y_9^*, y_{10}^*, \varnothing\}$$
$$\mathbf{y}' = \{y_1^*, y_2^*, y_3^*, y_4^*, \square, y_6^*, y_7^*, \square, y_9^*, y_{10}^*, \varnothing\}$$

然后分别按照 4.3.2节方法使基线模型和改进模型在此存在错误的样本上运行，并得到如下结果：

$$\hat{\mathbf{y}} = \{y_1^*, \square, y_3^*, y_4^*, \square, y_6^*, y_7^*, \blacksquare, y_9^*, y_{10}^*, \varnothing\}$$
$$\tilde{\mathbf{y}} = \{y_1^*, y_2^*, y_3^*, y_4^*, \square, y_6^*, \blacksquare, y_8^*, y_9^*, y_{10}^*, \varnothing\}$$

其中 “■” 和 “∅” 分别代表填充符 **PAD** 和句子结束符 **EOS**。根据本节提出的方法，我们对以上数据进行了统计，各阶匹配的 n 元文法的统计结果如表 4.1所示，其中有效的各阶 n 元文法均分别为 11, 10, 9, 8：

Prediction	Compared to $\mathbf{y}'$				Compared to $\mathbf{y}^*$			
results $\hat{\mathbf{y}}$ from baseline	10	6	4	2	8	4	1	0
results $\tilde{\mathbf{y}}$ from enhanced	9	7	5	3	9	6	4	2

表 4.1 示例上匹配的 n 元文法统计数据

Table 4.1 Matched n-gram Statistics on the Example

4.3.4　评估指标与结论

在上一节，我们详细描述了如何通过收集模型预测结果来与参考内容（输入馈送或黄金参考）作比较得出统计量。通过统计量，我们可以对每阶 n 元文法计算 4 组 n 元文法匹配精度数据，即 $P^{(\mathbf{y}', \hat{\mathbf{y}})}$，$P^{(\mathbf{y}^*, \hat{\mathbf{y}})}$，$P^{(\mathbf{y}', \tilde{\mathbf{y}})}$ 和 $P^{(\mathbf{y}^*, \tilde{\mathbf{y}})}$。我们进一步对使用同一个参考内容的两个模型的 n 元文法匹配精度做差，即使用改进模型 n 元文法匹配精度减去基线模型的 n 元文法匹配精度，计算公式如下：

$$\Delta P^{\mathbf{y}'} = P^{(\mathbf{y}', \tilde{\mathbf{y}})} - P^{(\mathbf{y}', \hat{\mathbf{y}})}$$
$$\Delta P^{\mathbf{y}^*} = P^{(\mathbf{y}^*, \tilde{\mathbf{y}})} - P^{(\mathbf{y}^*, \hat{\mathbf{y}})}$$

对于精度差，结合数学绘图分析，我们给出如下两个定义：

1. 如果所有的 n 阶 n 元文法的匹配精度差均大于零，则说明改进后的模型在缓解曝光偏差问题方面优于基线模型；

2. 如果精度差随着错误程度的增大而增大，则认为改进后的模型在抗曝光偏差方面的能力显著好于基线模型；

3. 对于 $\Delta P^{\mathbf{y}'}$ 和 $\Delta P^{\mathbf{y}^*}$ 两者，倾向于选择源于更切合实际的后者。

4.3.5 评估方案框架

我们在前面几节中详细介绍了基于 n 元文法匹配精度的评估方案的各个组成部分，在本节中，我们将给出该方案的整体框架。

不失一般性地，对于含有 R 个黄金参考的测试数据集 $\mathbf{D}^*$：

$$\mathbf{D}^* = \{(\mathbf{x}^*_m, \mathbf{y1}^*_m, \mathbf{y2}^*_m, \cdots, \mathbf{yR}^*_m) | m = 1, 2, 3, \cdots, N\}$$

我们按照 4.3.1节的方法对该数据集进行预处理，即使用不同的的不确定性对每个黄金参考应用词丢弃技术构建的存在错误的数据集 $\mathbf{D}'$：

$$\mathbf{D}' = \{\mathbf{D}'_1, \mathbf{D}'_2, \cdots, \mathbf{D}'_C\}$$

其中 C 为应用的不确定性的种类数，如在 4.3.2节中提到，我们对词丢弃的不确定性概率 p 分别设置为 $p = 0.0$，$p = 0.05$, $p = 0.10$, $p = 0.15$ 和 $p = 0.20$，则 $C = 5$。对于每个子数据集 $\mathbf{D}'_k$，其中 $k = 1, 2, \cdots, C$：

$$\mathbf{D}'_k = \{(\mathbf{x}^*_m, \mathbf{y1}'_{km}, \mathbf{y2}'_{km}, \cdots, \mathbf{yr}'_{km}) | m = 1, 2, \cdots, N\}$$

我们分别用收敛的基线模型和改进模型在这个子数据集中每个含有错误的黄金参考上按 4.3.2节的方法以评估模式运行训练程序，并根据预测结果按照算法 2统计有效的 n 元文法数量 $\#total^r_k$ 和匹配的 n 元文法数量 $\#matched^r_k$，其中 $r = 1, 2, \cdots, R$。然后我们以微平均方式计算在不确定性种类 k 下整个数据集上的 n 元文法匹配精度 P_k（4 个元组对应 4 个 P_k）：

$$P_k = \frac{\sum_{r=1}^{R} \#matched^r_k}{\sum_{r=1}^{R} \#total^r_k}$$

最后我们以不确定性种类的值为横坐标、基线模型和改进模型的预测结果相对于输入馈送和黄金参考的精度为纵坐标绘制曲线图，根据 4.3.4的几点定义即可对模型在抗曝光偏差能力的提升得出综合结论。

算法 3 ValidAndMacthedNGramsStatProgForDataset($\mathbf{D}^*$, p, n)

Require: 测试数据集 $\mathbf{D}^*$，用于词丢弃的不确定性值 p，最大 n 元文法阶数 n

$\mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}')} \leftarrow 0,\ \mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}')} \leftarrow 0$ for all $o = 1, 2, \cdots, n$

$\mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)} \leftarrow 0,\ \mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)} \leftarrow 0$ for all $o = 1, 2, \cdots, n$

$\mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}')} \leftarrow 0,\ \mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}')} \leftarrow 0$ for all $o = 1, 2, \cdots, n$

$\mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)} \leftarrow 0,\ \mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)} \leftarrow 0$ for all $o = 1, 2, \cdots, n$

for each sample $\mathbf{s} \in \mathbf{D}^*$ **do**

    $\mathbf{s} \leftarrow (\mathbf{x}^*, \mathbf{y1}^*, \mathbf{y2}^*, \cdots, \mathbf{yR}^*)$

    for each reference $\mathbf{y}^* \in \mathbf{s}$ **do**

        $\mathbf{y}' \leftarrow$ **WordDropout**$(\mathbf{y}^*, \mathbf{p})$

        $\hat{\mathbf{y}} \leftarrow$ **BaselineModel**$(\mathbf{x}^*, \mathbf{y}')$

        $\tilde{\mathbf{y}} \leftarrow$ **EnhancedModel**$(\mathbf{x}^*, \mathbf{y}')$

        for each order $o \in 1, 2, \cdots, n$ **do**

            $t^{(\hat{\mathbf{y}},\mathbf{y}')}, c^{(\hat{\mathbf{y}},\mathbf{y}')} \leftarrow$ **ValidAndMatchedNGramsStatProg**$(\hat{\mathbf{y}}, \mathbf{y}', o)$

            $t^{(\hat{\mathbf{y}},\mathbf{y}^*)}, c^{(\hat{\mathbf{y}},\mathbf{y}^*)} \leftarrow$ **ValidAndMatchedNGramsStatProg**$(\hat{\mathbf{y}}, \mathbf{y}^*, o)$

            $t^{(\tilde{\mathbf{y}},\mathbf{y}')}, c^{(\tilde{\mathbf{y}},\mathbf{y}')} \leftarrow$ **ValidAndMatchedNGramsStatProg**$(\tilde{\mathbf{y}}, \mathbf{y}', o)$

            $t^{(\tilde{\mathbf{y}},\mathbf{y}^*)}, c^{(\tilde{\mathbf{y}},\mathbf{y}^*)} \leftarrow$ **ValidAndMatchedNGramsStatProg**$(\tilde{\mathbf{y}}, \mathbf{y}^*, o)$

            $\mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}')} \leftarrow \mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}')} + t^{(\hat{\mathbf{y}},\mathbf{y}')},\ \mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}')} \leftarrow \mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}')} + c^{(\hat{\mathbf{y}},\mathbf{y}')}$

            $\mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)} \leftarrow \mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)} + t^{(\hat{\mathbf{y}},\mathbf{y}^*)},\ \mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)} \leftarrow \mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)} + c^{(\hat{\mathbf{y}},\mathbf{y}^*)}$

            $\mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}')} \leftarrow \mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}')} + t^{(\tilde{\mathbf{y}},\mathbf{y}')},\ \mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}')} \leftarrow \mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}')} + c^{(\tilde{\mathbf{y}},\mathbf{y}')}$

            $\mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)} \leftarrow \mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)} + t^{(\tilde{\mathbf{y}},\mathbf{y}^*)},\ \mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)} \leftarrow \mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)} + c^{(\tilde{\mathbf{y}},\mathbf{y}^*)}$

        end for

    end for

end for

for each order $o \in 1, 2, \cdots, n$ **do**

    $P_o^{(\hat{\mathbf{y}},\mathbf{y}')} \leftarrow \frac{\mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}')}}{\mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}')}},\ P_o^{(\hat{\mathbf{y}},\mathbf{y}^*)} \leftarrow \frac{\mathbf{C}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)}}{\mathbf{T}_o^{(\hat{\mathbf{y}},\mathbf{y}^*)}},\ P_o^{(\tilde{\mathbf{y}},\mathbf{y}')} \leftarrow \frac{\mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}')}}{\mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}')}},\ P_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)} \leftarrow \frac{\mathbf{C}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)}}{\mathbf{T}_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)}}$

    yield $P_o^{(\hat{\mathbf{y}},\mathbf{y}')}, P_o^{(\hat{\mathbf{y}},\mathbf{y}^*)}, P_o^{(\tilde{\mathbf{y}},\mathbf{y}')}, P_o^{(\tilde{\mathbf{y}},\mathbf{y}^*)}$

end for

基于 n 元文法匹配精度的曝光偏差性能评估方案的中精度差指标计算程式如算法 3所示。算法 3中，**WordDropout**$(\mathbf{y}^*, p)$ 表明词丢弃操作，即对序列 $\mathbf{y}^*$ 中的

元素以概率 p 替换为 **UNK**，但对结尾的 **EOS** 不进行操作，**BaselineModel**$(\mathbf{x}^*, \mathbf{y}')$ 和 **EnhancedModel**$(\mathbf{x}^*, \mathbf{y}')$ 分别表示对基线模型和改进模型在测试样本 $(\mathbf{x}^*, \mathbf{y}')$ 上以评估模式运行训练程序进行预测的程序。

4.4 实验结果与分析

在本节接下来的内容中，我们将使用本项工作提出的方法对第 3章的工作进行评估，实验结果表明，第 3章提出的方法能够有效缓解曝光偏差问题，提升模型的抗曝光偏差能力。

4.4.1 实验数据

我们使用第 3章收敛的基线模型和改进模型进行实验，它们来自多种规模的多个数据集的多个语言对。具体来说，实验数据信息如表所示：

#	Dataset & Description	System & Description	BLEU
1	WMT16 En⇒Ro: 608,319 newstest2016: 1,999	**Baseline** (Share & Decoder WT)	33.20
		Enhanced (Share & Decoder WT)	**33.22**
2		**Baseline** (Share)	32.21
		Enhanced (Share)	**33.24**$^{\Uparrow}$
3	WMT16 Ro⇒En: 608,319 newstest2016: 1,999	**Baseline** (Share & Decoder WT)	32.82
		Enhanced (Share & Decoder WT)	**33.22**$^{\uparrow}$
4		**Baseline** (Share)	31.88
		Enhanced (Share)	**32.72**$^{\Uparrow}$
5	LDC Zh⇒En: 1,252,977 MT03-08: 919/1,788/1,082/1,664/1,357	**Baseline**	42.69
		Enhanced	**44.23**$^{\Uparrow}$
6	WMT16 En⇒De: 4,500,966 newstest2013: 3,003	**Baseline** (Share & Decoder WT)	27.26
		Enhanced (Share & Decoder WT & *Fusion only*)	**27.91**$^{\uparrow}$

表 4.2 评估用实验模型及配置信息

Table 4.2 Model to be Tested and Configurations for Evaluation

表 4.2中，“**Dataset & Description**” 列描述了训练集的名称及句子对数和测试集及测试样本数。“**System & Description**” 列描述了系统名称及运行设置，其中 “**Baseline**”、“**Enhanced**” 为基线模型、改进模型。“Share” 表示模型使用联合词表，即源端和目标端共享词汇表，“Decoder WT” 代表解码器使用权重绑定技术 (Press 和 Wolf, 2017)，而 “*Fusion only*” 表示改进模型仅使用了其中的内部状

态融合技术 3.3。“**BLEU**” 列给出了模型在测试集上的平均 BLEU 值。

4.4.2　实验结果及分析

我们对表 4.2的 6 组对比模型实施了评估方案，图 4.1、图 4.2、图 4.3、图 4.4、图 4.5和图 4.6给出了相应的评测结果。

在这些图中，每个图均有 1 元语法、2 元语法、3 元语法和 4 元语法的曲线子图组成。每个子图中，横轴代表构建存在错误的输入馈送时对黄金参考施加词丢弃时的概率，纵轴表示模型基于给存在错误的输入馈送时在整个测试集范围内的预测精确度。实线代表改进模型的结果，而虚线表示基线模型的结果。灰线的曲线是由模型预测结果对输入馈送计算得来的，而橙线的曲线是对黄金参考计算的结果。具体而言，对于子图中的 4 条曲线，它们与 4.3.3节中提到的 4 个元组的对应关系如下：

- 灰色圆形虚线：$(\mathbf{y}', \hat{\mathbf{y}})$；
- 橙色正方形虚线：$(\mathbf{y}^*, \hat{\mathbf{y}})$；
- 灰色三角形实线 $(\mathbf{y}', \tilde{\mathbf{y}})$；
- 橙色菱形实线：$(\mathbf{y}^*, \tilde{\mathbf{y}})$。

在表 4.2对应的 6 组图中，除了 BLEU 值提升最大的第 5 组数据外，我们观察到实线的总在虚线下，这是由于我们构建存在错误的输入馈送时引入的都是 **UNK**，因此改进模型以此为参照能够预测得更加准确，对翻译意义也不大，因为最终用于评测的是黄金参考，因此 3节中的定义 3设定是有意义的。表 4.2中，就数值而言，提升最大和最小的分别是第 5 组（+1.54）和第 1 组（+0.02）数据，且前者提升的统计显著性很强，而后者从统计显著性意义来讲不算真正意义上的性能提升。这两组数据对应的评估结果曲线图 4.1、图 4.5中，以输入馈送和黄金参考作为参照的结果均表明了改进模型在抗曝光偏差能力的提升是显著的[7]。结合图 4.1和图 4.2、图 4.3和图 4.3这两组实验结果来看，翻译系统性能提升的显著性与抗曝光偏差能力的提升的显著与否无必然联系。图 4.6的结果显示，仅应用 3.3节的方法使得模型抗曝光偏差能力下降，但随着错误程度的增大，这种差距在逐渐减小。从 6 组图中的趋势来看，总体而言，第 3章的方法对提升模型抗曝光偏差能力有效。

[7]虽然图 4.1中以输入馈送作为参考的结果在存在的错误较轻时的 n 元文法匹配精度上要低于基线模型，但随着错误程度的增大，改进模型的结果明显要好，并且定义 3指出这个结果是次要的。

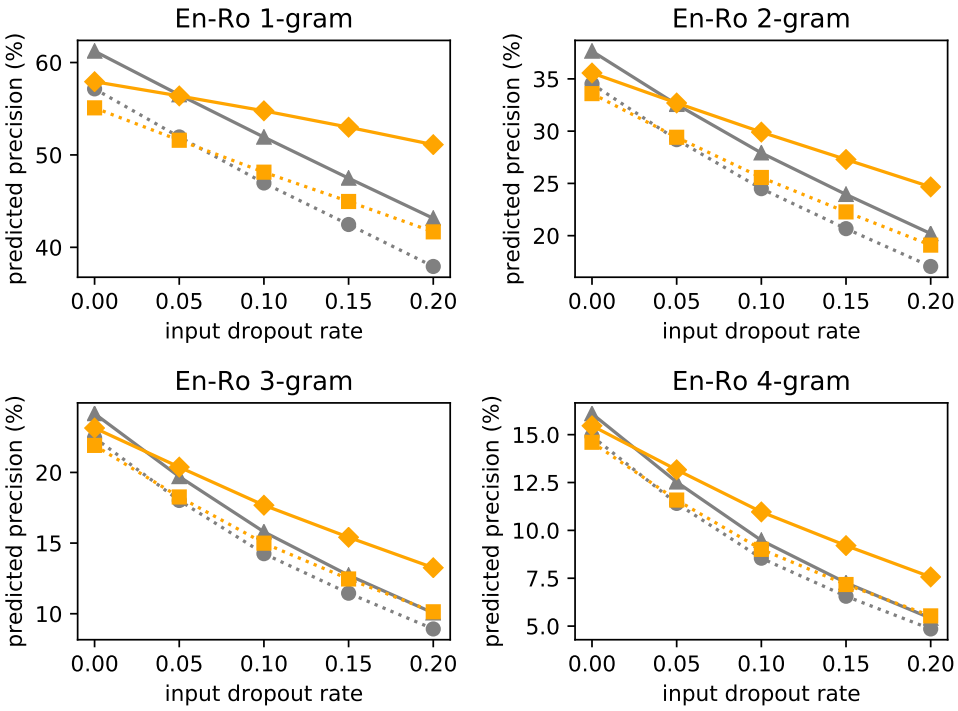

图 4.1 表4.2中第 1 组评估结果曲线: En⇒Ro (33.20 v.s 33.22)

Figure 4.1 Curves of Evaluation Results for the 1st of in Table 4.2: En⇒Ro (33.20 v.s 33.22)

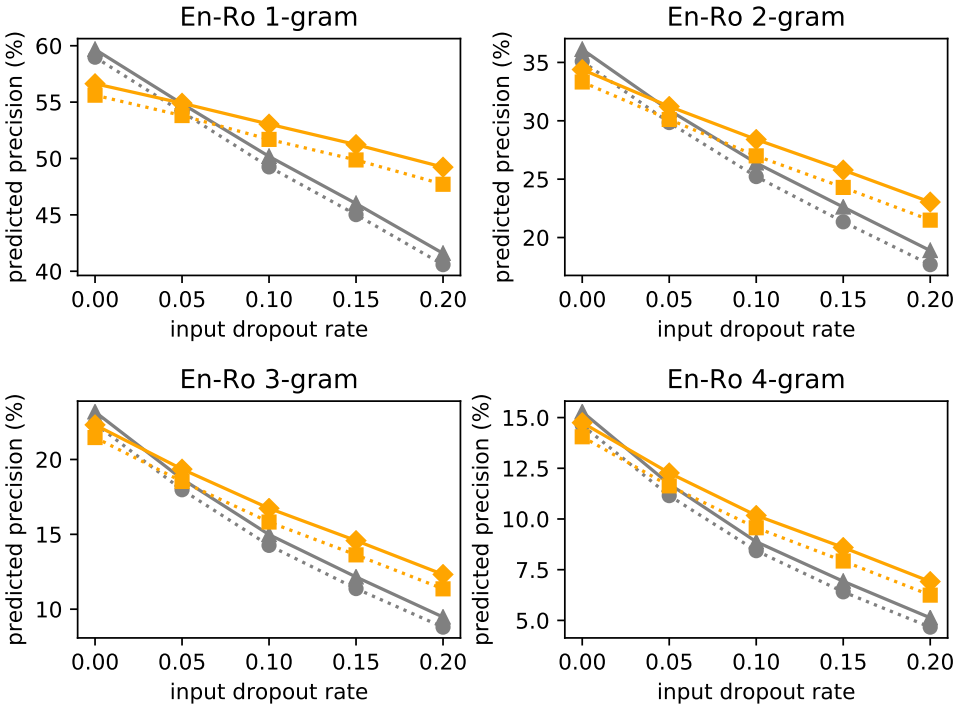

图 4.2 表4.2中第 2 组评估结果曲线: En⇒Ro (32.21 v.s 33.24⇑)

Figure 4.2 Curves of Evaluation Results for the 2nd of in Table 4.2: En⇒Ro (32.21 v.s 33.24⇑)

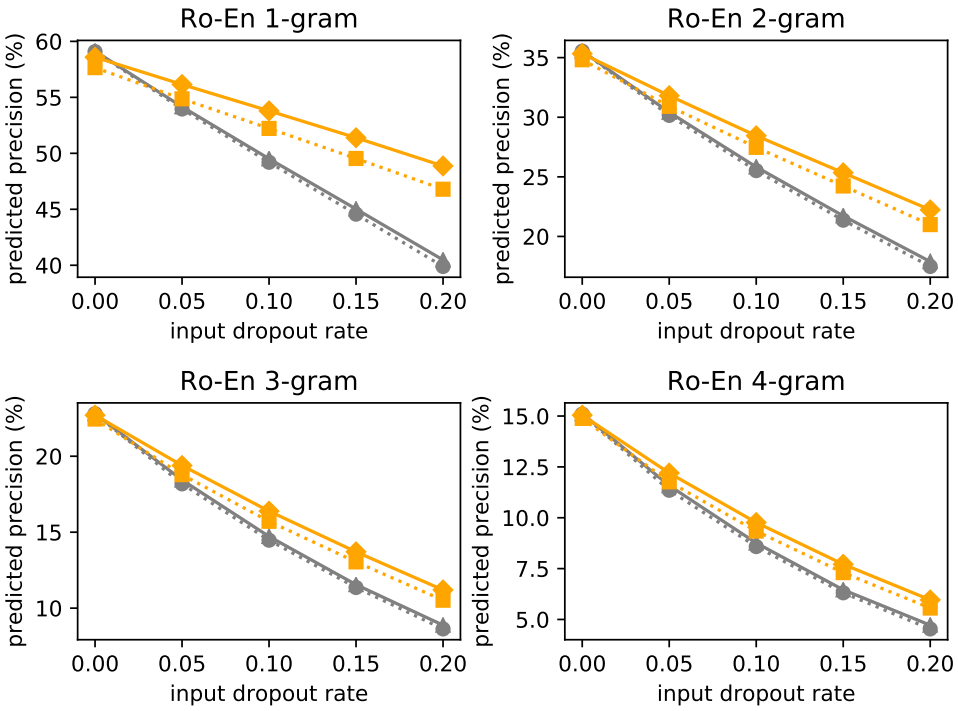

图 4.3 表4.2中第 3 组评估结果曲线: Ro⇒En (32.82 v.s 33.22$^{\uparrow}$)

Figure 4.3 Curves of Evaluation Results for the 3rd of in Table 4.2: Ro⇒En (32.82 v.s 33.22$^{\uparrow}$)

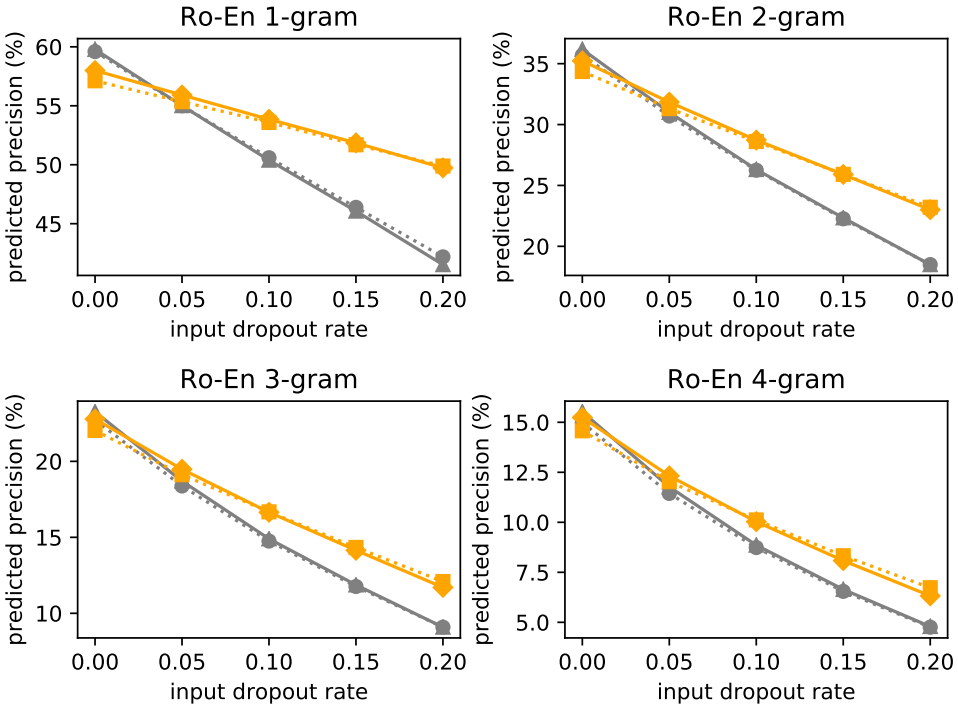

图 4.4 表4.2中第 4 组评估结果曲线: Ro⇒En (31.88 v.s 32.72$^{\Uparrow}$)

Figure 4.4 Curves of Evaluation Results for the 4th of in Table 4.2: Ro⇒En (31.88 v.s 32.72$^{\Uparrow}$)

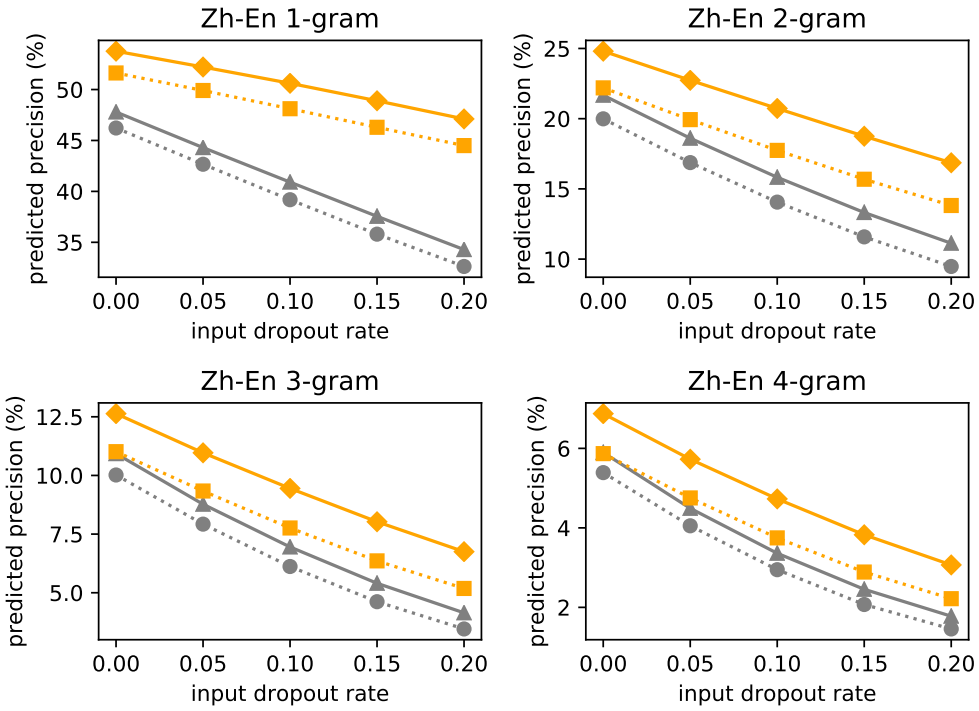

图 4.5 表4.2中第 5 组评估结果曲线: Zh⇒En (42.69 v.s 44.23⇑)

Figure 4.5 Curves of Evaluation Results for the 5th of in Table 4.2: Zh⇒En (42.69 v.s 44.23⇑)

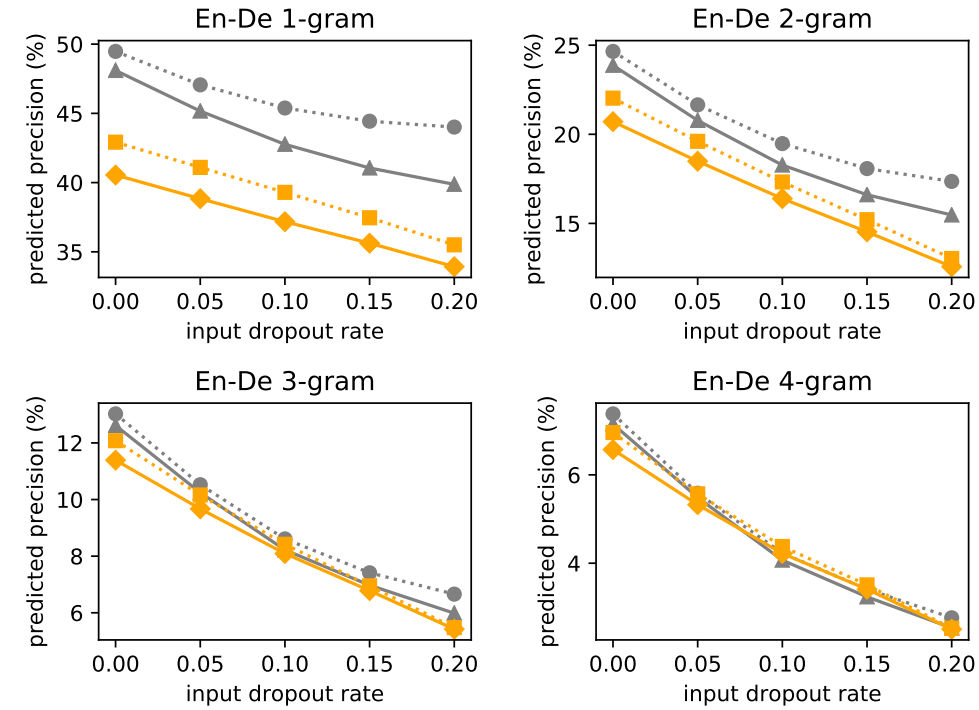

图 4.6 表4.2中第 6 组评估结果曲线: En⇒De (27.26 v.s 27.91↑)

Figure 4.6 Curves of Evaluation Results for the 6th of in Table 4.2: Zh⇒En (27.26 v.s 27.91↑)

4.5　本章小结

在本章工作中，出于对曝光偏差问题出现的必然性且影响模型性能，而评估及调试对于一个问题来说又很重要的考虑，我们提出了基于 n 元文法匹配精确度的曝光偏差性能评估方案，整个过程流程如下:

- 预处理：对整个测试数据集的每个黄金参考应用不同确定性的词丢弃构建存在不同程度错误的数据集；
- 运行：同时在改进后的模型和基线模型上以评估模式运行训练程序（不进行参数更新），获得它们基于存在相同程度错误的上下文的预测结果；
- 统计：按照算法 2统计上述预测结果与相应参照标准（输入馈送或黄金参考）的各阶有效 n 元文法和匹配的 n 元文法计数，$n = 1, 2, 3, 4$；
- 计算：计算数据集级别上两个模型的在各阶的 n 元文法匹配精度；
- 结论：通过做差或者做差后观察以不确定性为横轴、精度差为纵轴的曲线，根据 4.3.4节的定义得出结论。

我们提出了一个相对更为合理的评估方案，以期更好地衡量新的方法在缓解曝光偏差问题方面的改进及得到更多的内部细节。在该方案下，我们验证了前一个工作的方法能够增强模型在抗曝光偏差方面的能力。尽管如此，由于没有对结论的定义没有理论支持，导致可能缺乏统计层面上的意义，这点有待进一步改进。

第 5 章　总结与展望

5.1　总结

语言是人类沟通的桥梁，在全球化的浪潮中，日益增长的跨语言交流需要使得我们对高效率、高质量的机器翻译的需求更为迫切。而机器翻译是自然语言处理领域的应用级任务，在早先的翻译方法中，是集分词、命名实体识别及词法句法分析等技术于一体的系统。虽然当前主流的神经机器翻译方法占据了空前的优势，但机器翻译涉及自然语言方方面面的事实以及神经网络黑箱模式的特点使得重新从问题源头思考改进的方向和设计出能够挖掘内在工作原理的研究永葆青春。

因此本文研究中，针对翻译是在语义指导下进行的不同自然语言间的转换的基本原理，提出在模型训练过程中施加语义对齐的约束并在解码时融入语义要素来提升模型的性能，建立了语义对齐框架后，我们的方法在解决了语义坍缩等训练问题后显著地提升了模型的性能。另一方面，出于对尽可能多地利用监督信号的考量，加上机器翻译的训练必然存在监督信号，因此模型在训练过程和解码阶段接触的信息必然是不一致的。着眼于这种差异，则原始的曝光偏差问题仅是这种差异的很小一类，因此我们提出了构建更为通用的程序在神经网络的基本组成部分—隐状态层面对差异进行磨合并进一步利用了前一个过程引入的弱引导作为辅助监督信号并在原始的问题上进行了实验验证。对于一个问题来讲，研究如何测量、评估它和研究如何对其改进几乎同等重要，鉴于曝光偏差问题的评估手段尚缺或者不合理，我们在详细分析了训练和解码过程之后提出了基于 n 元文法匹配精度的评估方案。

详细来说，本文针对神经机器翻译中的语义保持性问题和曝光偏差问题进行了以下 3 项研究：

- 我们提出了在神经机器翻译中显式的引入语义对齐约束。为了实现这个目标，我们首先建立了一种能够解释一对多翻译现象的句子语义空间概念模型 S3CM，基于该模型我们设计了对齐的衡量度量。为了实现显式的语义抽取过程，我们使用基于 n 元文法的语义抽取器。为了将语义信息融入到解码过程中，我们提出通过可广播的语义集成网络来对形状不兼容的原始语义表示和抽象的

Token 表示进行融合。此外，为了让对齐有意义，我们利用深度前馈神经网络的通用函数近似性质设法将目标端语义映射到源端语义空间。深入分析了训练过程中出现的语义坍缩现象产生的原因后，我们在对学成语义的信息量进行操作后实现了模型性能的提升。最后我们验证了性能提升是由于引入语义带来的。

- 我们提出了在细粒度的隐状态级别对产生于曝光偏差问题源的内部状态进行训练期间的融合的方法，为此我们引入了概念上的噪声解码器。完全平行的解码器对引入了可用来构建辅助监督信号的可能，借此进一步增强了模型的性能。该方法在能够一致地收敛模型的机器翻译性能，并且不破坏训练的并行性，亦可看作是一种内部进化的重新训练方法。
- 我们提出了基于 n 元文法匹配精度的曝光偏差问题评估方案。该方案建立在让模型运行在错误程度介于训练过程和测试阶段之间并且可控的基于测试集构建的数据集上，对多种 n 元文法的精度进行综合考量，来得出新方法对缓解曝光偏差问题是否有效以及提升模型抗曝光偏差能力程度的定性结论。

然而，本研究工作亦有不足之处，包括但不仅限于以下几点：

1. 分布式语义的高维性及受可视化手段的制约导致语义对齐研究工作中提出的句子语义空间概念模型的正确与否无法判定；

2. 评估方案中得到的结果没有进行理论上的推导，统计量未在具有统计意义的数值下定义，仅能下定性结论。

5.2 展望

在上一节中，我们对整体的研究工作进行了回顾，并指出了存在的不足。由于研究本身就是一个不断地总结前人经验进行进一步探究的过程，因此过程中有些阶段的成果在下一阶段看来是甚至是错误的，如“天圆地方”、“地心说”、“日心说”等说法的演进，由于问题在被不断的发掘和重新定义，不必过于执着研究结果是否绝对正确，当且仅当我们用了当时科学的实验方法进行了探究。尽管本文对神经机器翻译中的语义对齐问题进行了谈讨，但也只是浅尝辄止。语义对齐问题始终存在，曝光偏差问题亦是如此。由于语义无法被很好地形式化表述，语义对齐框架下的真实对齐质量的测量也尤为必要。

参考文献

宗成庆. 统计自然语言处理[M]. 2008: 1-475.

ALINEJAD A, SIAHBANI M, SARKAR A. Prediction improves simultaneous neural machine translation[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 3022-3027. https://www.aclweb.org/anthology/D18-1337. DOI: 10.18653/v1/D18-1337.

BA J L, KIROS J R, HINTON G E. Layer normalization[J]. 2016.

BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[C/OL]//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.

BAU A, BELINKOV Y, SAJJAD H, et al. Identifying and controlling important neurons in neural machine translation[C/OL]//7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. 2019. https://openreview.net/forum?id=H1z-PsR5KX.

BENGIO S, VINYALS O, JAITLY N, et al. Scheduled sampling for sequence prediction with recurrent neural networks[M/OL]//CORTES C, LAWRENCE N D, LEE D D, et al. Advances in Neural Information Processing Systems 28. Curran Associates, Inc., 2015: 1171-1179. http://papers.nips.cc/paper/5956-scheduled-sampling-for-sequence-prediction-with-recurrent-neural-networks.pdf.

BENGIO Y, DUCHARME R, VINCENT P. A neural probabilistic language model [M/OL]//LEEN T K, DIETTERICH T G, TRESP V. Advances in Neural Information Processing Systems 13. MIT Press, 2001: 932-938. http://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.pdf.

BENGIO Y, LOURADOUR J, COLLOBERT R, et al. Curriculum learning[C/OL]//ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning. New York, NY, USA: ACM, 2009: 41-48. http://doi.acm.org/10.1145/1553374.1553380.

CHEN K, WANG R, UTIYAMA M, et al. Syntax-directed attention for neural machine translation [EB/OL]. 2018. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16060.

CHEN M X, FIRAT O, BAPNA A, et al. The best of both worlds: Combining recent advances in neural machine translation[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for

Computational Linguistics, 2018b: 76-86. https://www.aclweb.org/anthology/P18-1008. DOI: 10.18653/v1/P18-1008.

CHENG Y, TU Z, MENG F, et al. Towards robust neural machine translation[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 1756-1766. https://www.aclweb.org/anthology/P18-1163. DOI: 10.18653/v1/P18-1163.

CHENZE SHAO Y F F M, Jinchao Zhang, ZHOU J. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation[J]. arXiv preprint arXiv:1911.09320, 2019.

CHO K, ESIPOVA M. Can neural machine translation do simultaneous translation?[J/OL]. CoRR, 2016, abs/1606.02012. http://arxiv.org/abs/1606.02012.

CHO K, VAN MERRIËNBOER B, BAHDANAU D, et al. On the properties of neural machine translation: Encoder–decoder approaches[C/OL]//Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. Doha, Qatar: Association for Computational Linguistics, 2014a: 103-111. https://www.aclweb.org/anthology/W14-4012. DOI: 10.3115/v1/W14-4012.

CHO K, VAN MERRIËNBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation[C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014b: 1724-1734. https://www.aclweb.org/anthology/D14-1179. DOI: 10.3115/v1/D14-1179.

DALVI F, DURRANI N, SAJJAD H, et al. Incremental decoding and training methods for simultaneous translation in neural machine translation[C/OL]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 493-499. https://www.aclweb.org/anthology/N18-2079. DOI: 10.18653/v1/N18-2079.

DING Y, LIU Y, LUAN H, et al. Visualizing and understanding neural machine translation[C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017: 1150-1159. https://www.aclweb.org/anthology/P17-1106. DOI: 10.18653/v1/P17-1106.

DINU G, MATHUR P, FEDERICO M, et al. Training neural machine translation to apply terminology constraints[C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 3063-3068. https://www.aclweb.org/anthology/P19-1294. DOI: 10.18653/v1/P19-1294.

DOMHAN T. How much attention do you need? a granular analysis of neural machine transla-

tion architectures[C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 1799-1808. https://www.aclweb.org/anthology/P18-1167. DOI: 10.18653/v1/P18-1167.

DUONG L, ANASTASOPOULOS A, CHIANG D, et al. An attentional model for speech translation without transcription[C/OL]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, 2016: 949-959. https://www.aclweb.org/anthology/N16-1109. DOI: 10.18653/v1/N16-1109.

FENG Y, XIE W, GU S, et al. Modeling fluency and faithfulness for diverse neural machine translation[J]. arXiv preprint arXiv:1912.00178, 2019.

GEHRING J, AULI M, GRANGIER D, et al. Convolutional sequence to sequence learning[C/OL]//Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. 2017: 1243-1252. http://proceedings.mlr.press/v70/gehring17a.html.

GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. MIT Press, 2016.

GOYAL K, DYER C, BERG-KIRKPATRICK T. Differentiable scheduled sampling for credit assignment[C/OL]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vancouver, Canada: Association for Computational Linguistics, 2017: 366-371. https://www.aclweb.org/anthology/P17-2058. DOI: 10.18653/v1/P17-2058.

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. 2018a. https://openreview.net/forum?id=B1l8BtlCb.

GU J, HASSAN H, DEVLIN J, et al. Universal neural machine translation for extremely low resource languages[C/OL]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018b: 344-354. https://www.aclweb.org/anthology/N18-1032. DOI: 10.18653/v1/N18-1032.

GUO J, TAN X, XU L, et al. Fine-tuning by curriculum learning for non-autoregressive neural machine translation[J]. arXiv preprint arXiv:1911.08717, 2019.

HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[J]. arXiv preprint arXiv:1512.03385, 2015.

HINTON G E, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J/OL]. CoRR, 2015, abs/1503.02531. http://arxiv.org/abs/1503.02531.

KINGMA D P, BA J. Adam: A method for stochastic optimization[C/OL]//3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. http://arxiv.org/abs/1412.6980.

KOEHN P. Statistical significance tests for machine translation evaluation[C/OL]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain: Association for Computational Linguistics, 2004: 388-395. https://www.aclweb.org/anthology/W04-3250.

LEE J, MANSIMOV E, CHO K. Deterministic non-autoregressive neural sequence modeling by iterative refinement[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 1173-1182. https://www.aclweb.org/anthology/D18-1149. DOI: 10.18653/v1/D18-1149.

LI Z, WANG R, CHEN K, et al. Explicit sentence compression for neural machine translation[C]// the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020). 2020.

LIU X, WONG D F, LIU Y, et al. Shared-private bilingual word embeddings for neural machine translation[C/OL]//Proceedings of the 57th Conference of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 3613-3622. https://www.aclweb.org/anthology/P19-1352.

LUONG T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation[C/OL]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 1412-1421. https://www.aclweb.org/anthology/D15-1166. DOI: 10.18653/v1/D15-1166.

MARUF S, HAFFARI G. Document context neural machine translation with memory networks [C/OL]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018: 1275-1284. https://www.aclweb.org/anthology/P18-1118. DOI: 10.18653/v1/P18-1118.

MENG F, ZHANG J. Dtmt: A novel deep transition architecture for neural machine translation [J/OL]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(01):224-231. https://aaai.org/ojs/index.php/AAAI/article/view/3789. DOI: 10.1609/aaai.v33i01.3301224.

MICULICICH L, RAM D, PAPPAS N, et al. Document-level neural machine translation with hierarchical attention networks[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 2947-2954. https://www.aclweb.org/anthology/D18-1325. DOI: 10.18653/v1/D18-1325.

MIHAYLOVA T, MARTINS A F T. Scheduled sampling for transformers[C/OL]//Proceedings of

the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. Florence, Italy: Association for Computational Linguistics, 2019: 351-356. https://www.aclweb.org/anthology/P19-2049. DOI: 10.18653/v1/P19-2049.

MURTHY R, KUNCHUKUTTAN A, BHATTACHARYYA P. Addressing word-order divergence in multilingual neural machine translation for extremely low resource languages[C/OL]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 3868-3873. https://www.aclweb.org/anthology/N19-1387. DOI: 10.18653/v1/N19-1387.

OTT M, EDUNOV S, GRANGIER D, et al. Scaling neural machine translation[C]//Proceedings of the Third Conference on Machine Translation (WMT). 2018.

OTT M, EDUNOV S, BAEVSKI A, et al. fairseq: A fast, extensible toolkit for sequence modeling [C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 48-53. https://www.aclweb.org/anthology/N19-4009.

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 Computational Linguistics. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002: 311-318. https://www.aclweb.org/anthology/P02-1040. DOI: 10.3115/1073083.1073135.

PEREYRA G, TUCKER G, CHOROWSKI J, et al. Regularizing neural networks by penalizing confident output distributions[J/OL]. CoRR, 2017, abs/1701.06548. http://arxiv.org/abs/1701.06548.

POURDAMGHANI N, GHAZVININEJAD M, KNIGHT K. Using word vectors to improve word alignments for low resource machine translation[C/OL]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 524-528. https://www.aclweb.org/anthology/N18-2083. DOI: 10.18653/v1/N18-2083.

PRESS O, WOLF L. Using the output embedding to improve language models[C/OL]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Valencia, Spain: Association for Computational Linguistics, 2017: 157-163. https://www.aclweb.org/anthology/E17-2025.

RANZATO M, CHOPRA S, AULI M, et al. Sequence level training with recurrent neural networks [J]. CoRR, 2015, abs/1511.06732.

SALESKY E, SPERBER M, WAIBEL A. Fluent translations from disfluent speech in end-to-end

speech translation[C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 2786-2792. https://www.aclweb.org/anthology/N19-1285. DOI: 10.18653/v1/N19-1285.

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 (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 1715-1725. https://www.aclweb.org/anthology/P16-1162. DOI: 10.18653/v1/P16-1162.

SHAH H, BARBER D. Generative neural machine translation[M/OL]//BENGIO S, WALLACH H, LAROCHELLE H, et al. Advances in Neural Information Processing Systems 31. Curran Associates, Inc., 2018: 1346-1355. http://papers.nips.cc/paper/7409-generative-neural-machine-translation.pdf.

SHAO C, CHEN X, FENG Y. Greedy search with probabilistic n-gram matching for neural machine translation[C/OL]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 4778-4784. https://www.aclweb.org/anthology/D18-1510. DOI: 10.18653/v1/D18-1510.

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 (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 1683-1692. https://www.aclweb.org/anthology/P16-1159. DOI: 10.18653/v1/P16-1159.

SHU R, LEE J, NAKAYAMA H, et al. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior[J]. AAAI, 2020.

SONG L, GILDEA D, ZHANG Y, et al. Semantic neural machine translation using AMR[J/OL]. Transactions of the Association for Computational Linguistics, 2019, 7:19-31. https://www.aclweb.org/anthology/Q19-1002. DOI: 10.1162/tacl_a_00252.

SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: A simple way to prevent neural networks from overfitting[J/OL]. Journal of Machine Learning Research, 2014, 15:1929-1958. http://jmlr.org/papers/v15/srivastava14a.html.

SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks [M/OL]//GHAHRAMANI Z, WELLING M, CORTES C, et al. Advances in Neural Information Processing Systems 27. Curran Associates, Inc., 2014: 3104-3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

TU Z, LU Z, LIU Y, et al. Modeling coverage for neural machine translation[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long

Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 76-85. https://www.aclweb.org/anthology/P16-1008. DOI: 10.18653/v1/P16-1008.

VAIBHAV V, SINGH S, STEWART C, et al. Improving robustness of machine translation with synthetic noise[C/OL]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 1916-1920. https://www.aclweb.org/anthology/N19-1190. DOI: 10.18653/v1/N19-1190.

VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[M/OL]// GUYON I, LUXBURG U V, BENGIO S, et al. Advances in Neural Information Processing Systems 30. Curran Associates, Inc., 2017: 5998-6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

VASWANI A, BENGIO S, BREVDO E, et al. Tensor2tensor for neural machine translation[J/OL]. CoRR, 2018, abs/1803.07416. http://arxiv.org/abs/1803.07416.

VENKATRAMAN A, HEBERT M, BAGNELL J A. Improving multi-step prediction of learned time series models[C/OL]//AAAI'15: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI Press, 2015: 3024-3030. http://dl.acm.org/citation.cfm?id=2888116.2888137.

VOITA E, SENNRICH R, TITOV I. Context-aware monolingual repair for neural machine translation[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 877-886. https://www.aclweb.org/anthology/D19-1081. DOI: 10.18653/v1/D19-1081.

WANG C, ZHANG J, CHEN H. Semi-autoregressive neural machine translation[C/OL]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 479-488. https://www.aclweb.org/anthology/D18-1044. DOI: 10.18653/v1/D18-1044.

WANG L, TU Z, WAY A, et al. Exploiting cross-sentence context for neural machine translation [C/OL]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, 2017: 2826-2831. https://www.aclweb.org/anthology/D17-1301. DOI: 10.18653/v1/D17-1301.

WESTON J, CHOPRA S, BORDES A. Memory networks[C/OL]//3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. http://arxiv.org/abs/1410.3916.

WIETING J, BERG-KIRKPATRICK T, GIMPEL K, et al. Beyond BLEU:training neural machine translation with semantic similarity[C/OL]//Proceedings of the 57th Annual Meeting of the Asso-

ciation for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 4344-4355. https://www.aclweb.org/anthology/P19-1427. DOI: 10.18653/v1/P19-1427.

WU Y, SCHUSTER M, CHEN Z, et al. Google's neural machine translation system: Bridging the gap between human and machine translation[J/OL]. CoRR, 2016, abs/1609.08144. http://arxiv.org/abs/1609.08144.

YANG M, WANG R, CHEN K, et al. Sentence-level agreement for neural machine translation [C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 3076-3082. https://www.aclweb.org/anthology/P19-1296. DOI: 10.18653/v1/P19-1296.

ZHANG B, XIONG D, SU J, et al. Variational neural machine translation[C/OL]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics, 2016: 521-530. https://www.aclweb.org/anthology/D16-1050. DOI: 10.18653/v1/D16-1050.

ZHANG W, FENG Y, MENG F, et al. Bridging the gap between training and inference for neural machine translation[C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 4334-4343. https://www.aclweb.org/anthology/P19-1426. DOI: 10.18653/v1/P19-1426.

ZHENG B, ZHENG R, MA M, et al. Simpler and faster learning of adaptive policies for simultaneous translation[C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019a: 1349-1354. https://www.aclweb.org/anthology/D19-1137. DOI: 10.18653/v1/D19-1137.

ZHENG R, MA M, ZHENG B, et al. Speculative beam search for simultaneous translation [C/OL]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019b: 1395-1402. https://www.aclweb.org/anthology/D19-1144. DOI: 10.18653/v1/D19-1144.

ZHENG Z, ZHOU H, HUANG S, et al. Modeling past and future for neural machine translation [J/OL]. Transactions of the Association for Computational Linguistics, 2018, 6:145-157. https://www.aclweb.org/anthology/Q18-1011. DOI: 10.1162/tacl_a_00011.

ZHU J, XIA Y, WU L, et al. Incorporating bert into neural machine translation[C/OL]//International Conference on Learning Representations. 2020. https://openreview.net/forum?id=Hyl7ygStwB.

致　谢

光阴荏苒，蓦然回首，三年的硕士研究生生涯仿佛在弹指间远去。这段珍贵的学习经历使我受益颇丰。在雁栖湖的集中教学阶段，远郊优美的校园环境令人心旷神怡，我有幸在这结识了一群志趣相投的小伙伴。回到计算所后的两年时光，在研究组的科研氛围里，我正式踏上了向往的科研之路。值此毕业之际，我要对所有曾经在心理上支持过及在行动上帮助过我的老师、同学和家人，对培养我的中国科学院计算技术研究所、中国科学院大学表示由衷的感谢！

首先我要衷心地感谢录取我进入计算所的自然语言处理组团队，是您们让我有机会接触并能够深入了解通往人工智能的必要知识—自然语言处理。衷心地感谢刘群老师能够录取我。刘老师在自然语言处理领域耕耘几十载，拥有渊博的学识、丰富的经验以及对科研敏锐的洞察力，他坐得住冷板凳的专注精神始终是支撑我进行科研工作的重要动力来源。非常感谢冯洋老师为实验室营造了浓厚的学术氛围并增强了已有的实验环境，她制定的科研规章制度井井有条且与时俱进。冯老师是我的导师，她对科研充满热情，学术成果颇多。在科研道路上，冯老师始终起着督促我的作用，她对讨论科研思路的态度始终激励着我，使我纵使屡败，旋即屡战。作为组里唯一的科研指导老师，冯老师以一己之力便挑起了对全组上下十多名硕博生进行科研指导的重担，可谓艰辛！感谢实验室秘书刘琳老师为研究组提供的便利，帮我处理了很多工作上的事务。

作为一个早先在组里实习过的学生，我亦感激曾经的各位老师和师兄师姐们对我在学习和生活中的帮助和指导，是您们让我最早感受到实验室的风采。感谢姜文斌老师和赵秋野老师的知遇之恩，您们是最早带我入门自然语言处理的老师。姜老师在工作中充满活力，为人幽默风趣，为组里营造了轻松愉悦的氛围。赵老师具有优秀的管理和协调能力、直爽的做事风格让我受益匪浅。感谢赵红梅老师和郑达奇、孟凡东、刘洋（女）、王明轩、陈宏申、顾茂杰、刘毅、张金超、张源、马青松、胡镓伟、李响等师兄、师姐们对我生活和工作上的关心，您们对工作、生活的积极态度让我得益良多。

感谢张文师兄、薛海洋师兄和李京谕师姐，您们给予了我很大的帮助。张文师兄技术扎实、经验丰富且乐于助人，每每与师兄交流，均有新的收获。薛海洋

师兄亲切和善、幽默风趣，实习期间同住一屋的经历让我和他完全没有跨级的隔阂，他为组里高效地管理着服务器资源。李京谕师姐很爱笑、爱专研，虽交集不多，但其追求完美的心态是我所欣赏的。

感谢丁春发师兄、欧蛟师姐和刘舒曼师姐，在您们身上我学习到了很多优秀的品质；感谢与我同时进入实验室杨郑鑫、谷舒豪、申磊，和你们一起学习和工作是我的荣幸；感谢邵晨泽、单勇、李绩成、郭登级、李泽康等师弟们，您们都是实验室未来的希望。愿大家学业顺利，获得满意的成果。

再次感谢对我工作十分关心的各位师兄师姐们，郑达奇、王明轩、张金超、陈宏申、李响等师兄和马青松师姐。尤其是找工作期间张金超、陈宏申、李响和王明轩等师兄给出的热心帮助。感谢在组里的实习生赵紫毫、董宁等在科研项目中对我工作的支持。感谢高中同窗赖琦、刘雅露等为我的科研工作提供参考意见。

感谢家人在我读研期间对我的默默关心、支持和陪伴，是你们激励我不断进步，面对困境。即使今后我将面对新的挑战，你们永远是我身后最坚实的盾牌。

感谢的话语溢于言表。毕业在即，站在人生的新起点，带着学到的知识和经验，我定能稳稳地踏出每一步，不负我爱的人对我的厚望!

最后感谢在百忙之中评阅论文并提出宝贵意见的各位老师。

作者简历及攻读学位期间发表的学术论文与研究成果

基本情况

姓名: 王树根　性别: 男　出生日期: 1994.02.28　籍贯: 江西吉安

教育情况

2017.9 - 2020.7　中国科学院计算技术研究所硕士生

2013.9 - 2017.7　电子科技大学信息与软件工程学院本科生

攻读硕士期间参加的科研项目

2019.4 - 2019.6　CCMT 语音翻译评测　第二名

2017.9 - 2018.12　外交外事翻译系统　所级横向项目

攻读硕士学位期间的获奖情况

2018–2019 学年中国科学院大学三好学生

攻读硕士学位期间发表的文章

[1] Jie Zhu, Shugen Wang,Yanru Wu, Meijing Guan, and Yang Feng，A New Algorithm for Component Decomposition and Type Recognition of Tibetan Syllable，The 8th CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC 2019)，October 9-14, 2019，0-1，Dunhuang, Gansu Province, China，2019

联系方式

地址: 北京市海淀区科学院南路 6 号 中国科学院计算技术研究所

邮编: 100190

邮箱: wangshugen@ict.ac.cn