通过领域适应技术在机器翻译中利用语言相关性

Amit Kumar、Rupjyoti Baruah、Ajay Pratap，*IEEE*会员、Mayank Swarnkar和Anil Kumar Singh

摘要—**机器翻译（MT）面临的一项重大挑战是缺乏大量数据，尤其是平行对齐句子语料库。如果评估标准与资源丰富语言的评估一样严格，那么神经机器翻译（NMT）和统计机器翻译（SMT）都能利用如此大量的数据取得良好效果。然而，对于资源匮乏的语言而言，提升机器翻译输出质量颇具挑战，尤其是在神经机器翻译和统计机器翻译中。为应对机器翻译面临的挑战，我们提出一种新方法：基于采用 Kneser-Ney 平滑技术的 5-gram KenLM语言模型，计算句子的缩放相似度得分，尤其适用于相关语言，并据此从域外语料库中筛选领域内数据，从而提升机器翻译质量。此外，我们还采用多领域、微调和迭代回译等领域适应技术，在印地语-尼泊尔语语言对上对神经机器翻译和统计机器翻译中的新方法进行比较。我们的方法在多领域方法上提升了∼2 个BLEU点，在神经机器翻译微调上提升了 ∼3 个BLEU点，在迭代回译方法上提升了 ∼2个BLEU点。**

索引词—**机器翻译、领域自适应、相关语言、低资源语言。**

I. 引言

机器翻译（*MT*）本质上是一项具有挑战性的任务，因为它必须同时处理多种语言中的各种语言学不规则现象[1]。为了应对语言中的不规则现象，机器翻译需要大量语料库。语料库是一种语言资源，指存储在计算机上的大量非结构化文本集合。近期关于神经机器翻译（*NMT*）的研究表明，在某些高资源语言（*HRLs*）上，其表现已可媲美人工翻译；英语-德语和英语-法语等语言对尤其如此[2]。神经机器翻译模型在高资源语言上展现出令人惊叹的性能。然而，有些语言缺乏充足的资源，被称为低资源语言（*LRLs*）。如今，人们已开发出多种技术，以便在低资源场景下提升低资源语言的表现。但专为低资源语言设计的方法仍存在一些问题。其中一个原因是由于数据的数量和质量较低，评估这类语言比评估资源丰富的语言更加严格。另一个问题是，其结果在实际可用性方面仍远不及高资源语言。为解决这些问题，利用相似语言进行迁移学习（*TL*）是机器翻译领域一种颇具前景的解决方案[3]。将高资源语言的特征迁移至低资源语言的过程称为迁移学习[4]。领域适应（领域自适应）是迁移学习的一种类型，即将针对源分布训练的模型应用于不同但相关的目标领域[5]。有时，“领域”一词用于指一组相似语言中的某种语言，而领域自适应用于将高资源语言的特征迁移至低资源语言。

尽管如此，为低资源语言（LRL）开发的神经机器翻译（NMT）技术仍值得进一步探索，因为这些技术确实展现出了改进效果。在这方面，一个有利事实是，许多此类语言对在语言学层面或多或少存在密切关系，例如马拉地语—印地语和尼泊尔语—印地语[6], [7]。此外，通常情况下，大多数此类语言对至少有一种语言在词汇、形态特征、句法和语义方面与其密切相关。借助领域自适应（DA）等技术，我们可以利用密切相关语言之间的这种相似性，以改进低资源机器翻译。本文正是针对尼泊尔语—印地语语对，在两个翻译方向上开展了这项工作。印地语（*HI*）和尼泊尔语（*NE*）同属印欧语系的共同源头：印度—雅利安语支。印地语采用天城文书写[8]。尼泊尔语同样采用天城文书写，主要通行于尼泊尔以及不丹和印度的部分地区。由于这两种语言的使用者数千年来一直保持接触，因此同源词之间存在交换和借用[9], ，使这两种语言彼此更加相似。除词汇方面的相似性外，这两种语言还具有相同的词序和正字法特征。两种语言在词汇和音系特征上的相似性可通过词语的重叠得到体现（表 VIII）。由于印地语和尼泊尔语都是印度语族语言，因此有必要弥合两者之间的交流鸿沟。开发翻译工具能够消除人际交流中的沟通障碍。

在许多语言对的机器翻译中，领域适应问题已得到充分研究。现有的大多数方法侧重于模型层级的领域自适应，以及 [10], [11], [12] 和 [13]中提出的句子级方法。然而，目前还没有专门针对相似或相关语言的方法。因此，有必要提出一种专为

A. Kumar、R. Baruah、A. Pratap、M. Swarnkar 和 A. K. Singh 就职于印度理工学院（贝拿勒斯印度教大学）瓦拉纳西分校计算机科学与工程系，印度 221005。电子邮箱：{amitkumar.rs.cse17, rupjyotibaruah.rs.cse18, ajay.cse, mayank.cse, aksingh.cse}@iitbhu.ac.in。

相似语言设计的方法，这可以显著提升机器翻译的效果。

为了解决领域适应问题，我们提出了一种面向相似语言、在句子层面进行基于缩放相似度得分的句子选择方法。在句子选择中，我们的假设是，基于 WX 转写方案编码的语料库训练的语言模型 [14] 有助于更好地学习域外句子与域内句子之间的相似性。与其他数据选择方法相比，我们基于采用 Kneser-Ney 平滑技术训练的 5-gram KenLM语言模型，计算句子的缩放相似度得分 （*SSS*） [15]。

本文的贡献总结如下：

1）据我们所知，我们首次将句子的 *SSS* 而非交叉熵用作数据选择方法，从域外语料库中筛选域内句子。2）在训练语言模型时使用 WX 转写方案，有助于更好地学习相似语言的句子表示。3）结合迭代回译（IBT）使用基于缩放相似度得分的数据选择方法，效果优于当前最先进的模型。4）通过为所提出的模型设计可在多项式时间内完成的算法，高效解决领域适应问题。

本文其余部分安排如下：第二节讨论密切相关的研究。第三节介绍所提出的解决方案及其计算复杂度。第四节报告性能研究与结果分析、语料库统计，以及开展实验所需的实验设置。最后，第五节总结本文，并展望未来的工作。

表 I：相关工作的比较

论文	SMT	NMT	SSS	语言对
[10]	✗	✗	✗	英语-法语
[11]	✓	✗	✗	中文-英语
[16]	✓	✗	✗	英语-德语、法语、俄语、西班牙语
[17]	✓	✗	✗	英语-西班牙语
[18]	✓	✗	✗	德语-英语、阿拉伯语-英语
[19]	✓	✗	✗	阿拉伯语-英语、英语-法语
[20]	✓	✗	✗	中文-英语
[21]	✓	✗	✗	英语-德语、英语-土耳其语
[22]	✗	✓	✗	英语-法语、英语-德语
[23]	✗	✓	✗	英语-日语、德语-英语
[13]	✗	✓	✗	英语-德语、英语-意大利语
[24]	✗	✓	✗	德语-英语
[25]	✗	✓	✗	中文-英语、英语-法语、英语-德语
SSS	✓	✓	✓	印地语-尼泊尔语

注：EN：英语，FR：法语，ZH：中文，DE：德语，RU：俄语，ES：西班牙语，AR：阿拉伯语，TR：土耳其语，JA：日语，IT：意大利语

II. 相关工作

许多研究将领域自适应应用于统计机器翻译（*SMT*）和神经机器翻译模型 [26]。这些研究可分为数据中心方法和模型中心方法。数据中心技术包括基于语言模型（*LM*）从域外平行语料中筛选训练数据，或生成伪平行语料 [10], [11], [16]–[18]。相比之下，模型中心方法侧重于在模型或实例层面注入域内和域外模型 [19]–[21]。接下来，我们将回顾与数据中心方法和模型中心方法密切相关的研究。

A. 数据中心方法

在 [10], 中，作者提出了一种数据选择方法，计算单语语料库中每个句子的语言模型交叉熵差值。Cuong 等人[17] 基于使用期望最大化（*EM*）算法进行迭代加权邀请的方法，推导出一种邀请模型，并为包含多个差异较大领域混合内容的大型平行语料库上的机器翻译提供了一种数据选择方法。在 [18], 中，作者采用基于 EM 的混合建模和数据选择技术，使用两种联合模型进行机器翻译的领域自适应：操作序列模型（一种基于 n 元语法的翻译与重排序模型）和神经网络联合模型（一种连续空间翻译模型）。然而，如果语言模型是在极低资源数据上训练的，这些方法无法取得令人满意的结果。

在 [22], 中，作者探讨了神经机器翻译的句子级领域自适应。他们利用神经机器翻译模型内部的句子嵌入，并使用句子嵌入相似度筛选出与领域内语料库相近的域外句子。他们还提出了动态训练方法和多领域句子加权方法，以平衡训练数据的领域分布，并使训练数据与测试数据的领域分布相匹配。不过，这些方法虽有良好贡献，却没有探讨如何在模型训练中利用语言的正字法特征。

Dou 等人[13] 提出了用于迭代回译的动态数据选择与加权方法。在 [24],中，作者针对领域适应任务提出了分类器增强过滤迭代回译（CFIBT）。他们训练了两个基于卷积神经网络的二元分类器，一个使用源语，另一个使用目标语言，训练数据均由领域内和域外语料库组成。他们使用迭代回译生成合成平行语料库，并采用基于分类器的过滤，移除其中合成句子不属于该领域的句对。这些方法着重于将选择方法扩展至迭代回译，但没有探讨如何在数据选择方法中利用正字法特征和语言间的相关性。

B. 模型中心方法

在 [19], 中，作者在一个广受欢迎的开源统计机器翻译平台上提出了填充式自适应方法，并在语音翻译任务中对其进行了测试。Zhou 等人 [20] 介绍了一种基于句子权重的自适应方法，该方法依据训练集中句子与目标领域文本之间的相似度。在 [21], 中，作者提出了

FDA5 是一个用于特征衰减算法（FDA）的参数化、优化和实现框架。特征衰减算法是一类采用特征衰减的实例选择算法。

Sato 等人 [23] 提出了词汇适应，这是一种简单有效的微调方法，可将给定预训练神经机器翻译模型的嵌入层适配到目标领域。在微调之前，他们的方法会将目标领域单语数据诱导出的通用词嵌入投影到源领域嵌入空间，从而替换神经机器翻译模型的嵌入层。在 [25], 中，作者首先对模型进行剪枝，只保留重要的神经元或参数，使其同时负责通用领域和领域内翻译。随后，他们使用知识蒸馏，以原始完整模型为教师监督训练剪枝后的模型。最后，他们将模型扩展回原始大小，并对新增参数进行微调，以用于领域内翻译。

然而，上述所有模型在训练语言模型时都没有考虑文本的正字法特征。

C. 现有方法的不足

大多数现有方法都需要特定领域的额外单语数据来训练语言模型。语言模型需要充足的训练数据才能更好地学习，而数据是所有低资源机器翻译模型面临的主要挑战。与现有的先进方法不同，我们的方法虽然基于统计语言模型，却不需要大量数据来训练模型。我们通过将文本转写为 WX 转写方案，利用其正字法特征。

表 I 列出了我们提出的方法与其他相关工作的比较。与现有方法相比，我们的模型首创了基于 Kneser-Ney 平滑技术的句子 SSS，用于数据选择。此外，我们的模型基于 5 元 KenLM 训练，并作为过滤器区分域内数据和域外数据。

表 II：命名约定

约定	**领域**
域内	WMT2019
AGRI	农业
ENT	娱乐
BIB	圣经
MD_1	WMT+农业
MD_2	WMT+娱乐
MD_3	WMT+圣经
MD_4	农业+娱乐
MD_5	农业+圣经
MD_6	娱乐+圣经
MD_7	WMT+农业+娱乐
MD_8	WMT+农业+圣经
MD_9	WMT+娱乐+圣经
MD_{10}	农业+娱乐+圣经
MD_{11}	WMT+农业+娱乐+圣经

III. 提出的模型

本节将介绍我们提出的模型，旨在通过利用域外句子与域内句子之间的相似性，解决提升机器翻译质量的领域适应问题。表 II 列出了实验中使用的领域。我们的模型重点采用 SSS 数据选择方法进行领域自适应。所提出模型的架构如图 1 所示。模型以多个领域（农业、娱乐和圣经）的语料库为输入，使用 SSS 筛选域内句子。SSS 是通过使用公式 (1) 在经 WX 转写的领域内语料库上训练 KenLM 计算得出的。SSS 应用于多个领域中每个句子的源语端，并根据所得分数对句子排序。随后，模型选出满足阈值条件的句子，并将选出的句子存储为筛选后数据集。模型将筛选后的数据与领域内语料库拼接，并将拼接后的数据作为平行语料库（源语和目标语）存储，以便进一步训练机器翻译模型。最后，机器翻译模型给出预测的译文。此外，我们还设计了算法，并提出多领域、微调和迭代回译的时间复杂度，以比较各方法的性能；相关内容将在下一节介绍。

A. 多领域方法

本节仅关注基于语料库的方法。算法 1 描述了如何基于不同的语料库组合训练模型。我们将混合领域的训练数据和域内验证数据作为多领域算法 *(MDA)* 的输入。MDA 最终会生成所有 $k \in$ 训练数据 在两个方向上的已训练机器翻译模型列表（$MT_{s\to t,k}$,$MT_{t\to s,k}$）：源语（s）→目标语（t）和目标语（t）→源语（s）。首先，MDA 调用每份训练数据，并对模型 $MT_{s\to t}$ 进行训练，持续 $s \to t$ （第 1-2 行）。训练 $MT_{s\to t}$后，它会沿 $t \to s$ 方向训练模型 $MT_{t\to s}$ （第 3 行）。两个模型会持续训练，直到收敛。然后，MDA 将这两个模型添加到列表中（第 4 行）。调用每份训练数据后，MDA 会返回所有 $k \in$ 训练数据对应的列表（$MT_{s\to t,k}$,$MT_{t\to s,k}$）。

算法 1： 多领域方法

输入：$Training_Data \in$ 列表（域内、AGRI、娱乐、BIB, MD_1、MD_2、MD_3、MD_4、MD_5、MD_6、MD_7、MD_8、MD_9、MD_{10}、MD_{11}），$Valid_Data \in$ 域内，l = 列表 ()

输出：列表（$MT_{s\to t,\ k}$、$MT_{t\to s,\ k}$），$\forall k \in$训练_数据

1 **对于** $\forall k \in$训练_数据 **执行**

2 $MT_{s\to t} \leftarrow$ 使用 $Training_Data, Valid_Data$训练机器翻译对于 s → t;

3 $MT_{t\to s} \leftarrow$ 使用 $Training_Data, Valid_Data$训练机器翻译对于 t → s;

4 l.append($MT_{s\to t}$, $MT_{t\to s}$);

B. 域内多领域微调

算法 2 描述了微调方法*(FTA)*的训练过程。我们将域外（OD_p）和域内（ID_p）的平行训练数据，以及域内验证数据作为 FTA 的输入。最终，FTA 得到训练后的神经机器翻译模型列表（$NMT_{s\to t,\ b}$,

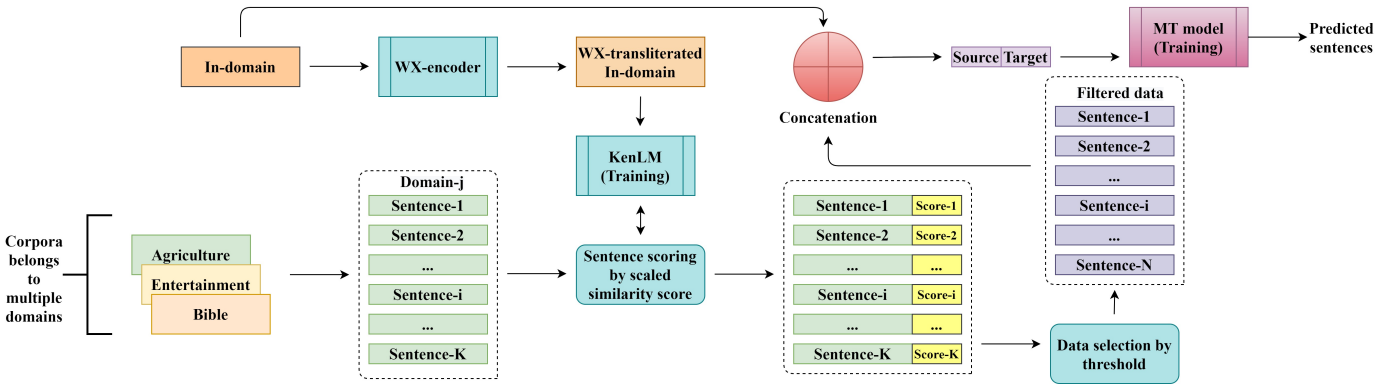


图 1：所提出的架构。

$NMT_{t\to s}$，$_b$）$\forall b \in OD_p$，其包含两个方向的模型：$s \to t$ 和 $t \to s$。首先，FTA 依次调用 OD_p 中的每个元素，并针对方向 $s \to t$ 训练模型 $NMT_{s\to t}$（第 1–2 行）。训练完 $NMT_{s\to t}$后，它会沿 $t \to s$ 方向训练模型 $NMT_{t\to s}$（第 3 行）。两个模型都会持续训练，直至收敛。随后，它使用上述模型 $NMT_{s\to t}$，$_b$获得的模型权重，针对方向 $s \to t$ 在 ID_p 上训练 $NMT_{s\to t}$（第 4 行）。类似地，它使用上述模型 $NMT_{t\to s}$，$_b$获得的模型权重，针对方向 $t \to s$在 ID_p 上训练$NMT_{t\to s}$（第 5 行）。然后，FTA 将两个模型都添加到列表中（第 6 行）。调用完 OD_p中的每个元素后，FTA 返回该列表（$NMT_{s\to t}$，$_b$，$NMT_{t\to s}$，$_b$）$\forall b \in OD_p$。

算法 2： 微调方法

输入：$OD_p \in$ 列表（AGRI、娱乐、BIB、MD_4, MD_5, MD_6, MD_{10}），$ID_p \in$ 域内，$Valid_Data \in$ 域内，l = 列表()

输出：列表($NMT_{s\to t,b}$, $NMT_{t\to s,b}$)，$\forall b \in OD_p$

对于 $\forall b \in OD_p$ **do**
- $NMT_{s\to t} \leftarrow$ 使用 OD训练神经机器翻译模型$_p$ 作为训练_数据，$Valid_Data$ 对于 s → t;
- $NMT_{t\to s} \leftarrow$ 使用 OD训练神经机器翻译模型$_p$ 作为训练_数据，$Valid_Data$ 针对 t → s;
- $NMT_{s\to t} \leftarrow$ 使用模型权重 ID_p 训练神经机器翻译模型来自 $NMT_{s\to t,\ b}$ 针对 s → t;
- $NMT_{t\to s} \leftarrow$ 使用模型权重 ID_p 训练神经机器翻译模型对于 t → s，从 $NMT_{t\to s,\ b}$;
- l.append($NMT_{s\to t}$, $NMT_{t\to s}$);

算法3： 用于 SSS 选择方法的领域自适应

输入：MD_{10}，$ID_p \in$ 域内，$Valid_Data \in$ 域内

输出：$MT_{s\to t}$，$MT_{t\to s}$

为 s → t 训练 KenLM（s $\in ID_p$）;
为 t → s 训练 KenLM（t $\in ID_p$）;
对于 $\forall j \in MD_{10}$ **do**
- S(j) ← $SSS(j)$ 对于 s → t;
- S'(j) ← $SSS(j)$ 对于 t → s;
- **如果** $S(j) \geq threshold1$ **则**
  - $ID_p' \leftarrow$j;
- **如果** $S'(j) \geq threshold2$ **则**
  - $ID_p'' \leftarrow$j;

$ID_p' \leftarrow ID_p + ID_p'$;
$MT_{s\to t} \leftarrow$ 使用 ID_p' 为 s →t 训练机器翻译;
$ID_p'' \leftarrow ID_p + ID_p''$;
$MT_{t\to s} \leftarrow$ 使用 ID_p'' 为 t → s 训练机器翻译;

C. 领域过滤方法

我们使用 SSS 来获取语料库（C）中 i^{th} 句子（S_i）的分数，计算方法如下：

$$SSS(S_i) = \frac{L_i - min(L_S)}{max(L_S) - min(L_S)} \quad (1)$$

其中，L_i 是 KenLM 为 S_i计算出的分数，min（L_S）和 max（L_S）分别是 KenLM 为 C中的句子计算出的最低分数和最高分数。

算法3使用 SSS，通过在 WX 转写后的源语句子和领域内语料库句子的目标侧文本上训练 KenLM，将领域内数据与域外语料库区分开来。我们使用 MD_{10} 和 ID_p 作为训练数据，并将领域内验证集作为输入传入基于缩放相似度分数的选择方法的领域自适应（DASSA）。DASSA 最终会得到两个方向上的训练后机器翻译模型（$MT_{s\to t}$、$MT_{t\to s}$），即 $s \to t$ 和 $t \to s$。首先，DASSA 在 s 和 t 上训练 KenLM（第 1–2 行）。训练完 KenLM 后，DASSA 遍历 MD_{10} 中的每个句子，并分别为 s 和 t 计算 SSS（第 3–5 行）。然后，它检查 s 和 t 的 SSS 是否均大于或等于某个阈值；如果 SSS 满足阈值条件，DASSA 就将该句子添加到经过筛选的领域内语料库（ID'_P 或 ID''_P）中（第 6–9 行）。DASSA 将筛选后的领域内语料库添加到 ID_p，并训练模型 $MT_{s\to t}$ 和$MT_{t\to s}$（第 10–13 行）。最后，它返回 $MT_{s\to t}$ 和$MT_{t\to s}$）。

D. 使用 *SSS* 选择方法的迭代回译

我们在 IBT 方法中使用 SSS 方法，筛选出与领域内数据相近的单语语料库。算法 4 描述了基于缩放相似度得分的迭代回译方法（SSSIBA）。我们将源语和目标语言单语语料库（M_s 和 M_t）、ID_p 语料库作为训练数据，并将领域内验证数据作为 SSSIBA 的输入。最后，SSSIBA 得到经过训练的神经机器翻译模型（$NMT_{s\to t}$、$NMT_{t\to s}$），分别用于两个方向的翻译：$s \to t$ 和 $t \to s$。首先，SSSIBA 在 ID_p 的源语端和目标语端训练 KenLM（第 1–2 行）。然后，它使用领域内平行语料库训练 $NMT_{s\to t}$ 和 $NMT_{t\to s}$（第 4–5 行）。

（第 4–5 行）。训练完成后，SSSIBA 分别使用 $NMT_{t\to s}$ 和 $NMT_{s\to t}$翻译M_t和 M_s（第 4–5 行）。然后，SSSIBA 对翻译后的单语数据（M_s' 和 M_t'）进行筛选，并创建合成平行语料库（SN_s' 和 SN_t'）（第 6–15 行）。接着，将合成平行语料库与 ID_p 拼接，生成 ID_p' 和 ID_p''（第 16–17 行）。最后，SSSIBA 分别使用 ID_p'和 ID_p'' 训练 $NMT_{s\to t}$ 和 $NMT_{t\to s}$（第 18–19 行）。整个过程重复进行，直到收敛。

算法 4： 采用 SSS 选择方法的迭代回译

输入：$M_t \in$ 单语语料(t)，$M_s \in$ 单语语料(s)，$ID_p \in$ 列表(域内)，$Valid_Data \in$ 域内
输出：列表($NMT_{s\to t}$, $NMT_{t\to s}$)

针对 s $\to$ t 训练 KenLM(s $\in ID_p$);
针对 t $\to$ s 训练 KenLM(t $\in ID_p$);
当 $NMT_{s\to t}$ 且 $NMT_{t\to s}$ 未收敛时 **执行**
  $M_s' \leftarrow$ 将 M_t 翻译 $NMT_{t\to s}$;
  $M_t' \leftarrow$ 翻译 M_s 使用 $NMT_{s\to t}$;
  对于 $\forall j \in M_s'$ **执行**
    S(j) $\leftarrow SSS(j)$ 对于 s $\to$ t;
    如果 $S(j) \geq threshold1$ **那么**
      $F_s \leftarrow$j;
  对于 $\forall j \in M_t'$ **执行**
    对于 t $\to$ s，S(j) $\leftarrow SSS(j)$;
    如果 $S(j) \geq threshold2$ **则**
      $F_t \leftarrow$j;
  $SN_s' \leftarrow (F_s, M_t)$;
  $SN_t' \leftarrow (M_s, F_t)$;
  $ID_p' \leftarrow ID_p + SN_s'$;
  $ID_p'' \leftarrow ID_p + SN_t'$;
  $NMT_{s\to t} \leftarrow$ 使用 ID_p' 训练神经机器翻译模型，s为 $\to$ t;
  $NMT_{t\to s} \leftarrow$ 使用 ID_p'' 训练神经机器翻译模型，t为 $\to$ s ;

E. 所提算法分析

本节讨论所提出模型的收敛性和时间复杂度。所提出模型的时间复杂度取决于其调用的机器翻译模块的复杂度。KenLM 的时间复杂度为$O(\log\log\ K)$，其中 K是有序数组 [15]中的键值对数量。Transformer 神经机器翻译的时间复杂度为 $O(W^2D + WD^2)$，其中 W是词向量的数量，D 是每个词向量的维度 [27]。统计机器翻译的时间复杂度取决于 KenLM、Moses 解码器和重排序模型的复杂度。采用直方图剪枝进行解码的时间复杂度为 $O(TL^2)$，其中 T 是假设的最大堆栈大小，L 是句子 [28]的长度。重排序模型的时间复杂度为 $O(TL)$ [28]。因此，统计机器翻译的时间复杂度为 $O(T^2L^3\log\log\ K)$。

各算法的时间复杂度将在下一节中介绍：

1) MDA：使用统计机器翻译和神经机器翻译的 MDA 时间复杂度分别为 $O(GT^2L^3\log\log\ K)$ 和 $O(GW^2D + GWD^2)$，其中 G 是用于训练的领域数量（算法 1 的第 1–3 行）。

2) FTA：FTA 的时间复杂度为 $O(HW^2D + HWD^2)$，其中 H 是用于训练的域外数量（算法 2 的第 1–5 行）。

3) DASSA：算法 3 第 1–2 行的时间复杂度为 $O(\log\log\ K)$，第 3–9 行为 $O(N)$，第 10–13 行对于 SMT 为 $O(T^2L^3\log\log\ K)$，对于 NMT 为 $O(W^2D + WD^2)$。因此，使用 SMT 和 NMT 的 DASSA 的时间复杂度分别为 $O(N + \log\log\ K + T^2L^3\log\log\ K))$ 和 $O(N + \log\log\ K + W^2D + WD^2)$，其中 N 是 MD_{10} 领域中的句子数量。

4) SSSIBA：算法 4 第 1-2 行的时间复杂度为 $O(\log\log\ K)$，第 4-5 行和第 18-19 行的时间复杂度为 $O(W^2D + WD^2)$，第 6-13 行的时间复杂度为 $O(M)$。因此，SSSIBA 的时间复杂度为 $O(\log\log\ K + W^2ID + WID^2 + IM)$，其中 M 是单语语料库中的句子数量，I 是使模型收敛所需的迭代次数。

5) 模型收敛：当模型连续 10 次迭代没有任何改进时，我们认为机器翻译模型已收敛。

IV. 性能研究

本节将讨论执行实验所需的语料库统计信息和实验设置。

A. 语料库描述

用于域内印地语-尼泊尔语数据训练、验证和测试的平行语料库，采集自机器翻译研讨会（WMT 2019）组织的一项相似语言共享任务 [7]。此外，农业和娱乐领域的数据来自 TDIL[29]，，圣经领域的数据和单语数据则收集自 OPUS [30]。表 III 详细列出了属于各个领域的语料库统计信息。

表 I：语料库描述

领域	训练句子
域内	65505
AGRI	14000
ENT	27000
BIB	30486
MD_1	79505
MD_2	92505
MD_3	95991
MD_4	41000
MD_5	44486
MD_6	57486
MD_7	106505
MD_8	109991
MD_9	122991
MD_{10}	71486
MD_{11}	136991
单语（印地语）	90000
单语（尼泊尔语）	90000

B. 实验设置

本节将讨论训练模型所需的不同框架和参数。

1）神经机器翻译：实验中，我们使用$Fairseq$ [31]——一个用于训练模型的序列建模工具包，并在$Nvidia\ V100\ GPU$上执行实验。我们使用***Transformer*** 架构训练神经机器翻译模型 [27]。我们使用 $SentencePiece$ 库进行预处理

[32], 并让模型学习联合词汇表。解码器 和编码器层 均设置为 5。编码器、解码器和输出层共用嵌入，这意味着我们的模型使用共享词典和嵌入空间，这对相似语言和领域自适应应该有所帮助。前馈网络中编码器 和解码器嵌入维度 设为 2048。编码器和解码器的嵌入维度 为 512。解码器和编码器的注意力头 数设为 2。模型采用随机失活、标签平滑 和权重衰减进行正则化，相应的超参数分别设为 0.4、0.2 和 0.0001。模型使用$Adam$进行优化，采用 $\beta 1 = 0.9$ 和 $\beta 2 = 0.98$，并将耐心值设为 10。

2）统计机器翻译（*SMT*）：我们使用开源工具包 $Moses$训练统计机器翻译（SMT）[33]。为了从印地语-尼泊尔语平行语料库中创建短语/词语对齐，我们使用$GIZA++$ [34]。训练时使用 5-gram KenLM语言模型[15]。我们使用最小错误率训练在验证集上调整参数，并使用测试集进行测试 [35]。

3）过滤模型：我们使用 KenLM 在领域内语料库上训练$5\text{-}gram$ 语言模型。我们将阈值设为 0.8，以便从域外数据中过滤出领域内数据。

C. 结果分析

为了评估模型在不同领域上的表现，我们使用了 $BLEU$ [36]、 [37]、 $chrF2$ [38]和 TER [39]，如表 V 和表 VII 所示。我们展示了在不同设置下使用统计机器翻译和神经机器翻译的方法。BLEU分数是研究人员普遍接受的一项标准指标，用于衡量预测翻译输出与人工翻译的参考句子（黄金标签） NLP [36]之间的准确度。可以观察到，BLEU分数越高，翻译输出的质量越好。BLEU 分数的公式如下 [36]:

$$BLEU = min\left(1, \frac{output_length}{reference_length}\right)\left(\prod_{i=1}^{4} precision_i\right) \tag{2}$$

其中输出_长度和参考_长度分别是预测句子和参考句子的长度。

由于 MD_{10} 由三个域外语料库组成：农业、娱乐和圣经。我们将过滤方法应用于 MD_{10} ，以便将结果与其他现有方法进行比较。

1）在语言模型中应用 *WX* 转写法：我们计算了使用 WX 转写和不使用 WX 转写时，基于领域内语料库源语端和目标端训练的 KenLM语言模型的困惑度。困惑度得分列于表 IV。我们发现，使用 WX 转写法转写后的训练数据训练的语言模型，其困惑度低于使用未经转写的训练数据训练的语言模型。困惑度较低的语言模型总是更好。这意味着，经 WX 转写的语言模型优于基于未经 WX 转写的语料库训练的语言模型。

因此，我们可以说，基于经 WX 转写训练的语言模型的数据选择方法优于现有方法。

表 IV：KenLM 的困惑度

语料库	困惑度
源语侧（HI）	477.324
目标端（NE）	550.333
源语端（HI）*	8.162
目标端（NE）*	6.756

*数据已转写为 WX 转写法

2）使用 *SSS* 方法对 *SMT* 和 *NMT* 的影响：表 V 给出了应用SSS进行领域过滤，并分别与 SMT 和 NMT 进行比较的结果。模型使用数量不同的语料库进行训练，这些语料库由多个领域的数据合并而成。领域过滤应用于该 MD_{10}领域的纯域外语料库（语料库分别属于农业、娱乐和圣经领域）。我们的方法在 SMT 上比 MDA 的表现高出约 $\sim$2.2，且在 NMT 上平均高出 $\sim$2。与 Moore 等人在 MD10 上提出的 [10] 数据选择方法相比，我们在 SMT 上提升了 $\sim$1 ，在 NMT 上提升了 $\sim$2 。取得这种提升的原因是，SSS 数据选择方法依赖于经 WX 转写训练的语言模型。WX 转写缩小了域外语料库与领域内语料库之间的语言距离（表 IV）。另一个原因是，使用 SSS 对分数进行缩放，缩小了句子最低分与最高分之间的巨大差距（分数范围为 0 到 1）。因此，该方法倾向于选择更多接近领域内语料库的句子。

3）使用相关语言进行微调的效果：表六展示了在域内数据上，对使用合并后的域外语料库训练的模型进行微调所得的结果。与未经微调的模型相比，微调使神经机器翻译的译文质量略有提升。这种结果的原因在于，域外数据由多个领域的数据构成。由于不同领域的数据在表示上存在较大差异，多领域数据限制了模型进一步提升性能的能力。我们提出的数据选择方法降低了域外语料库的数据复杂度，并在微调时带来 $\sim$3 的提升；在 MD_{10} 上应用 Moore 等人的 [10] 数据选择方法时，则带来 $\sim$2 的提升。取得如此显著提升的原因与上一节所述相同。SSS 缩放和 WX 转写有助于模型在选择时挑选出更多接近域内语料库的句子。

4）使用回译的效果：我们将SSS 数据选择方法与神经机器翻译中的迭代回译相结合。我们在尼泊尔语-印地语语言对上，将其与一次迭代回译（BT）以及 Dou 等人 [13]使用词频-逆文档频率提出的动态数据选择方法进行比较。表 VII 列出了我们整合的迭代回译方法与回译及动态数据选择方法的对比。我们整合的迭代回译方法优于回译，提升了 $\sim$2 ，并提升了 [13]，幅度为 $\sim$1

表 V：数据选择结果

模型	尼泊尔语→印地语			印地语→尼泊尔语		
	BLEU	chrF2	TER	BLEU	chrF2	TER
MDA（统计机器翻译）	43.91	0.53	0.601	47.11	0.57	0.562
[10]（统计机器翻译）	45.88	0.54	0.579	48.91	0.57	0.545
DASSA（统计机器翻译）	47.01	0.54	0.578	49.31	0.58	0.540
MDA（神经机器翻译）	30.51±0.04	0.49±0.00	0.645±0.000	32.57±0.64	0.52±0.00	0.624±0.000
[10] (神经机器翻译)	30.80±0.01	0.49±0.00	0.640±0.00	31.66±0.13	0.51±0.00	0.627±0.00
DASSA（神经机器翻译）	32.13±0.55	0.51±0.00	0.642±0.000	35.41±0.23	0.53±0.00	0.620±0.000

注意：神经机器翻译模型使用大小为 4000、5000 和 6000 的字节对编码进行训练。

表六：微调结果

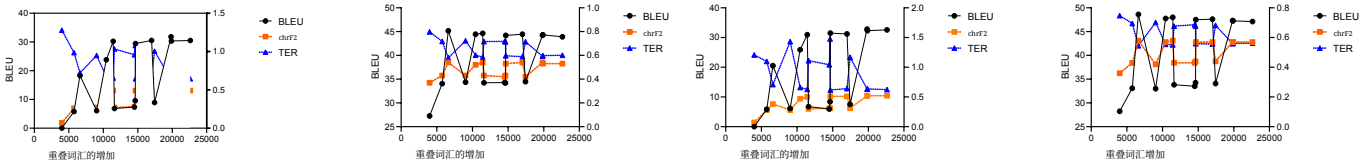
领域	尼泊尔语→印地语			印地语→尼泊尔语		
	BLEU	chrF2	TER	BLEU	chrF2	TER
AGRI*	15.69±0.02	0.34±0.00	0.791±0.000	16.60±0.21	0.35±0.00	0.789±0.000
ENT*	19.29±0.05	0.35±0.00	0.803±0.000	21.81±0.04	0.36±0.00	0.797±0.000
BIB*	7.75±0.00	0.28±0.00	0.796±0.000	10.50±0.02	0.29±0.00	0.790±0.000
MD4*	17.48±0.31	0.36±0.00	0.770±0.000	17.56±0.44	0.36±0.00	0.756±0.000
MD5*	19.21±0.03	0.37±0.00	0.759±0.000	21.70±0.01	0.37±0.00	0.769±0.000
MD6*	19.93±0.34	0.38±0.00	0.752±0.000	20.70±0.23	0.39±0.00	0.740±0.000
MD10*	16.91±0.04	0.36±0.00	0.758±0.000	18.32±0.42	0.37±0.00	0.742±0.000
MD10 [10]	17.88±0.03	0.37±0.00	0.756±0.000	19.12±0.67	0.38±0.00	0.739±0.000
MD10**	19.21±0.13	0.38±0.00	0.745±0.000	21.31±0.05	0.40±0.00	0.731±0.000

注：神经机器翻译模型是在字节对编码大小为 4000、5000 和 6000 的设置下训练的。*：使用 FTA方法。**在 MD10 领域应用 SSS方法，然后执行 FTA。

表 VII：结合数据选择的回译

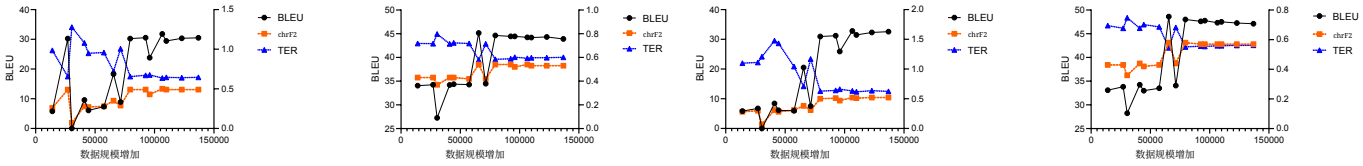
领域	尼泊尔语→印地语			印地语→尼泊尔语		
	BLEU	chrF2	TER	BLEU	chrF2	TER
BT*	21.17±0.04	0.36±0.00	0.699±0.00	23.67±0.14	0.41±0.00	0.645±0.000
Dou 等人 [13]	21.88	0.37	0.670	24.75	0.42	0.641
SSSIBA	22.91±0.15	0.39±0.00	0.589±0.00	25.88±0.27	0.44±0.00	0.561±0.00

注：神经机器翻译模型采用大小为 4000、5000 和 6000 的字节对编码进行训练。*在神经机器翻译中使用单语数据进行回译。



(a) 尼泊尔语→印地语神经机器翻译模型。 (b) 尼泊尔语→印地语统计机器翻译模型。 (c) 印地语→尼泊尔语神经机器翻译模型。 (d) 印地语→尼泊尔语统计机器翻译模型。

图2：重叠词汇与不同评估指标（BLEU、chrF2 和 TER）的关系。



(a) NE→HI 神经机器翻译模型。 (b) NE→HI 统计机器翻译模型。 (c) HI→NE 神经机器翻译模型。 (d) HI→NE 统计机器翻译模型。

图3：数据规模与不同评估指标（BLEU、chrF2 和 TER）的关系。

BLEU 分。将 SSS 集成到迭代回译后，翻译效果再次提升，这表明通过 SSS 和 WX 转写对分数进行缩放，可以筛选出更多接近领域内语料库的句子，从而提升模型性能。

5）语言间的相关性：为了定量衡量相似性，我们使用了 *char-BLEU* [40], *chrF2*和 *TER*，除了用于评估之外，这些评估指标还可用于评估相似度，前提是数据采用通用编码格式表示。因此，我们将所有训练数据编码为 WX encoding[14]。然后，将所有这些指标应用于所有领域的训练语料库。*char-BLEU* 得分越高，且

$chrF2$，表示语言之间越相似。另一方面，TER 得分越低，表明语言之间越相似。我们还使用重叠词汇来衡量每份训练数据中印地语和尼泊尔语之间的相似度。重叠词汇越多的领域，表示该领域的尼泊尔语和印地语语料库关联性或相似度越高。这样做是为了更深入地了解实验结果及其与相似度的关系。表 VIII 列出了各项指标的得分，用以表示语言对之间的相似度。其目的是报告并观察相似度估计值与评估结果之间的关系。圣经领域就是一个明显的例子。该领域的相似度最低，机器翻译的表现也最差。同样，在娱乐领域，相似度和机器翻译表现都很高。

表 VIII：尼泊尔语与印地语之间的相似度

领域	char-BLEU	chrF2	TER	OW*
域内	0.5255	0.47	0.857	6619
AGRI	0.5088	0.44	0.977	5757
ENT	0.5092	0.44	0.965	11622
BIB	0.2374	0.20	1.191	3998
MD_1	0.5183	0.46	0.915	11433
MD_2	0.5163	0.45	0.926	17002
MD_3	0.3281	0.28	1.107	10431
MD_4	0.5091	0.44	0.969	14632
MD_5	0.3119	0.26	1.140	9040
MD_6	0.3443	0.28	1.108	14516
MD_7	0.5144	0.45	0.939	19833
MD_9	0.3843	0.32	1.064	19877
MD_8	0.3661	0.30	1.083	14672
MD_{10}	0.3768	0.31	1.087	17440
MD_{11}	0.4053	0.34	1.052	22620

*重叠词的数量。

我们还借助图 2 所示的折线图，比较了重叠词汇对神经机器翻译和统计机器翻译的影响。我们可以看到印地语和尼泊尔语训练句子之间的重叠词汇。从图 2a、2b、2c 和 2d 可以看出，词汇重叠程度较高对神经机器翻译和统计机器翻译都有益处，但在某些数据与领域内语料库不相近的领域中例外。例如，“MD_{11}”领域包含 22620 个重叠词，与其他领域的语料库相比，该领域提升了神经机器翻译的 BLEU 分数。然而，尽管训练领域与测试领域的数据存在较大差异，我们在 MD10 上采用数据选择方法后，仍实现了 ~ 2 个 BLEU 分数的提升。可以说，共享词汇对神经机器翻译有积极影响，但即使共享词汇不利于机器翻译，我们的方法仍能带来显著提升。

6）增加数据规模的影响：我们借助图3所示的折线图，展示了增加训练数据规模对相关语言的神经机器翻译和统计机器翻译的影响。通过查看图3a和图3c，我们发现，与统计机器翻译相比，增加训练数据对神经机器翻译更有益。此外，用于评估模型的所有指标彼此之间都呈现相关性。从图3b和图3d可以看出，统计机器翻译在域内模型上的结果已经趋于饱和。

V. 结论

我们提出了一种基于 SSS 的数据选择方法，其表现优于现有的最先进方法。我们采用 Transformer 架构训练神经机器翻译模型，并采用 Moses 和 KenLM语言模型进行统计机器翻译，覆盖多个领域；随后在不同的领域适应方法设置下比较结果。我们的方法在 MDA 上提升了 ~ 2 BLEU分数，在 FTA 上提升了 ~ 3 个 BLEU 点，在迭代回译方法上提升了 ~ 2 个 BLEU 点。我们还发现，对于相似的语言对，在源语数据有限的情况下，统计机器翻译的翻译效果优于神经机器翻译。统计机器翻译表现出色的原因之一，是这些语言之间存在正字法相似性。

未来，我们将从动态视角扩展数据选择方法，并将其与多种相似语言的跨语言迁移学习相结合。

参考文献

[1] W. N. Locke 和 A. D. Booth，《语言机器翻译：十四篇论文》。纽约：麻省理工学院技术出版社和 Wiley，1955年。[2] I. Sutskever、O. Vinyals 和 Q. V. Le，“使用神经网络进行序列到序列学习”，载于《神经信息处理系统进展》（Z. Ghahramani、M. Welling、C. Cortes、N. Lawrence 和 K. Q. Weinberger 编），第27卷，Curran Associates, Inc.，2014年。[3] N. Wu、E. DeMattos、K. H. So、P.-z. Chen 和 Ç. Çöltekin，“相似语言的语言区分与迁移学习：特征组合与适应实验”，载于《第六届相似语言、变体与方言自然语言处理研讨会论文集》，（密歇根州安娜堡），第54–63页，计算语言学协会，2019年6月。[4] S. Ruder、M. E. Peters、S. Swayamdipta 和 T. Wolf，“自然语言处理中的迁移学习”，载于《2019年北美计算语言学协会会议论文集：教程》，（明尼苏达州明尼阿波利斯），第15–18页，计算语言学协会，2019年6月。[5] C. Chu 和 R. Wang，“神经机器翻译领域适应综述”，载于《第27届国际计算语言学会议论文集》，（美国新墨西哥州圣菲），第1304–1319页，计算语言学协会，2018年8月。[6] L. Barrault、M. Biesialska、O. Bojar、M. R. Costa-jussà、C. Federmann、Y. Graham、R. Grundkiewicz、B. Haddow、M. Huck、E. Joanis、T. Kocmi、P. Koehn、C.-k. Lo、N. Ljubešić、C. Monz、M. Morishita、M. Nagata、T. Nakazawa、S. Pal、M. Post 和 M. Zampieri，“2020年机器翻译会议（WMT20）成果”，载于《第五届机器翻译会议论文集》，（在线），第1–55页，计算语言学协会，2020年11月。[7] L. Barrault、O. Bojar、M. R. Costa-jussà、C. Federmann、M. Fishel、Y. Graham、B. Haddow、M. Huck、P. Koehn、S. Malmasi、C. Monz、M. Müller、S. Pal、M. Post 和 M. Zampieri，“2019年机器翻译会议（WMT19）成果”，载于《第四届机器翻译会议论文集（第2卷：共享任务论文，第1天）》，（意大利佛罗伦萨），第1–61页，计算语言学协会，2019年8月。[8] “印度宪法第17部分：官方语言。”
https://rajbhasha.nic.in/en/constitutional-provisions。访问日期：2021年6月5日。[在线]。[9] A. K. Singh 和 H. Surana，“为构建词汇资源而开展的南亚语言同源词研究”，载于《印度语言计算与处理词汇资源创建全国研讨会论文集》，（印度孟买），Citeseer，2007年。

[10] R. C. Moore 和 W. Lewis，“语言模型训练数据的智能选择”，载于《ACL 2010会议短论文集》，（瑞典乌普萨拉），第220–224页，计算语言学协会，2010年7月。[11] A. Axelrod、X. He 和 J. Gao，“通过伪域内数据选择实现领域适应”，载于《2011年自然语言处理经验方法会议论文集》，（英国苏格兰爱丁堡），第355–362页，计算语言学协会，2011年7月。[12] R. Wang、M. Utiyama、A. Finch、L. Liu、K. Chen 和 E. Sumita，“神经机器翻译领域适应的句子选择与加权”，《IEEE/ACM音频、语音与语言处理汇刊》，第26卷，第10期，第1727–1741页，2018年。[13] Z.-Y. Dou、A. Anastasopoulos 和 G. Neubig，“迭代回译的动态数据选择与加权”，载于《2020年自然语言处理经验方法会议论文集（EMNLP）》，（在线），第5894–5904页，计算语言学协会，2020年11月。[14] S. Diwakar、P. Goyal 和 R. Gupta，“使用 WX 转写法进行印度语言间转写”，载于《2010年自然语言处理会议论文集》，第147–150页，萨尔大学出版社，2010年。[15] K. Heafield，“KenLM：更快、更小的语言模型查询”，载于《第六届统计机器翻译研讨会论文集》，（苏格兰爱丁堡），第187–197页，计算语言学协会，2011年7月。[16] K. Duh、G. Neubig、K. Sudoh 和 H. Tsukada，“使用神经语言模型进行适应数据选择：机器翻译实验”，载于《计算语言学协会第51届年会论文集（第2卷：短论文）》，（保加利亚索非亚），第678–683页，计算语言学协会，2013年8月。[17] H. Cuong 和 K. Sima’an，“混合领域干草堆中的潜在领域翻译模型”，载于《COLING 2014——第25届国际计算语言学会议论文集：技术论文》，（爱尔兰都柏林），第1928–1939页，都柏林城市大学和计算语言学协会，2014年8月。[18] N. D. H. S. S. Joty 和 A. A. S. Vogel，“使用联合模型进行统计机器翻译领域适应”，《第十五届机器翻译峰会论文集》，第117页，2015年。[19] A. Bisazza、N. Ruiz 和 M. Federico，“基于短语的统计机器翻译适应中的填充法与插值法”，载于《2011年国际口语翻译研讨会（IWSLT）》，第136–143页，2011年。[20] X. Zhou、H. Cao 和 T. Zhao，“使用句子权重进行统计机器翻译领域适应”，载于《基于自然标注大数据的中文计算语言学与自然语言处理》（M. Sun、Z. Liu、M. Zhang 和 Y. Liu 编），（Cham），第153–163页，Springer国际出版社，2015年。[21] E. Biçici 和 D. Yuret，“利用特征衰减算法优化统计机器翻译的实例选择”，《IEEE/ACM音频、语音与语言处理汇刊》，第23卷，第2期，第339–350页，2015年。[22] R. Wang、M. Utiyama、A. Finch、L. Liu、K. Chen 和 E. Sumita，“神经机器翻译领域适应的句子选择与加权”，《IEEE/ACM音频、语音与语言处理汇刊》，第26卷，第10期，第1727–1741页，2018年。[23] S. Sato、J. Sakuma、N. Yoshinaga、M. Toyoda 和 M. Kitsuregawa，“神经机器翻译领域适应的词汇适应”，载于《计算语言学协会论文集：EMNLP 2020成果》，（在线），第4269–4279页，计算语言学协会，2020年11月。[24] S. Kumari、N. Jaiswal、M. Patidar、M. Patwardhan、S. Karande、P. Agarwal 和 L. Vig，“通过过滤式迭代回译实现神经机器翻译领域适应”，载于《第二届自然语言处理领域适应研讨会论文集》，（乌克兰基辅），第263–271页，计算语言学协会，2021年4月。[25] S. Gu、Y. Feng 和 W. Xie，“用于神经机器翻译领域适应的先剪枝后扩展模型”，载于《2021年北美计算语言学协会会议论文集：人类语言技术》，（在线），第3942–3952页，计算语言学协会

，2021年6月。

[26] P. Koehn、F. J. Och 和 D. Marcu，“统计短语翻译”，载于*2003*年北美计算语言学协会分会人类语言技术会议论文集，第127–133页，2003年。[27] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、L. u. Kaiser 和 I. Polosukhin，“注意力就是一切”，载于神经信息处理系统进展（I. Guyon、U. V. Luxburg、S. Bengio、H. Wallach、R. Fergus、S. Vishwanathan 和 R. Garnett 编），第30卷，Curran Associates, Inc.，2017年。[28] P. Koehn，《统计机器翻译》。剑桥大学出版社，2009年。[29] “印地语-尼泊尔语农业与娱乐文本语料库 ILCI-II。”*https://www.tdil-dc.in/index.php?lang* = 20, 英文。2021访问日期：6月。 [在线]。[30] J. Tiedemann，“OPUS中的平行数据、工具和接口”，载于第八届国际语言资源与评测会议（*LREC’12*）论文集，（土耳其伊斯坦布尔），第2214–2218页，欧洲语言资源协会（ELRA），2012年5月。[31] M. Ott、S. Edunov、A. Baevski、A. Fan、S. Gross、N. Ng、D. Grangier 和 M. Auli，“fairseq：一种快速、可扩展的序列建模工具包”，载于*2019*年北美计算语言学协会分会会议论文集（演示），（明尼苏达州明尼阿波利斯），第48–53页，计算语言学协会，2019年6月。[32] T. Kudo 和 J. Richardson，“SentencePiece：一种简单且与语言无关的神经文本处理子词分词器和去分词器”，载于*2018*年自然语言处理实证方法会议：系统演示，（比利时布鲁塞尔），第66–71页，计算语言学协会，2018年11月。[33] P. Koehn、H. Hoang、A. Birch、C. Callison-Burch、M. Federico、N. Bertoldi、B. Cowan、W. Shen、C. Moran、R. Zens、C. Dyer、O. Bojar、A. Constantin 和 E. Herbst，“Moses：统计机器翻译开源工具包”，载于计算语言学协会第*45*届年会伴卷：演示与海报论文集，（捷克共和国布拉格），第177–180页，计算语言学协会，2007年6月。[34] F. J. Och 和 H. Ney，“各种统计对齐模型的系统比较”，《计算语言学》，第29卷，第1期，第19–51页，2003年。[35] F. J. Och，“统计机器翻译中的最小错误率训练”，载于计算语言学协会第*41*届年会论文集，（日本札幌），第160–167页，计算语言学协会，2003年7月。[36] K. Papineni、S. Roukos、T. Ward 和 W.-J. Zhu，“BLEU：一种机器翻译自动评测方法”，载于计算语言学协会第*40*届年会论文集，（美国宾夕法尼亚州费城），第311–318页，计算语言学协会，2002年7月。[37] M. Post，“呼吁清晰报告BLEU分数”，载于第三届机器翻译会议：研究论文，（比利时布鲁塞尔），第186–191页，计算语言学协会，2018年10月。[38] M. Popović，“chrF：用于机器翻译自动评测的字符n元语法F分数”，载于第十届统计机器翻译研讨会论文集，（葡萄牙里斯本），第392–395页，计算语言学协会，2015年9月。[39] M. Snover、B. Dorr、R. Schwartz、L. Micciulla 和 J. Makhoul，“针对性人工标注下的翻译编辑率研究”，载于美洲机器翻译协会第七届会议：技术论文，（美国马萨诸塞州剑桥），第223–231页，美洲机器翻译协会，2006年8月8日至12日。[40] E. Denoual 和 Y. Lepage，“字符级BLEU：迈向无词语分隔符语言的机器翻译自动评测”，载于会议论文集伴卷（含海报、演示和教程摘要），2005年。