vivo AI 研究院 CCMT 2022 评测技术报告

李方圆[1]*，郭攀峰[1]，狄亚超[1]，王承之[1]，丁家杰[1,2]，李春元[1]，滕飞[1]
（1. vivo AI 研究院，浙江 杭州 311121；
2. 苏州大学，江苏 苏州 215006）

摘要：文本详细介绍了 vivo AI 研究院机器翻译团队参加第十八届全国机器翻译大会机器翻译评测(CCMT 2022)的参赛情况和各项任务采用的技术细节。在本次评测中，vivo 共参加了八个评测任务，分别是汉英新闻领域机器翻译（CE）、英汉新闻领域机器翻译（EC）、蒙汉日常用语领域机器翻译（MC）、藏汉政府文献领域机器翻译（TC）和维汉新闻领域机器翻译 5 个双语翻译评测任务，日英汉专利领域多语言翻译评测任务，以及低资源场景下的中泰、泰中语向机器翻译评测任务。报告将主要阐述本次参评系统采用的算法模型框架、数据处理、数据筛选和数据增强用到的方法，并分别给出在不同的方法配置下参评系统的评测结果对比与分析。

关键词 机器翻译；CCMT 2022；翻译评测；vivo
中图分类号：TP391 **文献标志码：**A

1 引言

文本详细介绍了 vivo AI 研究院机器翻译团队参加第十八届全国机器翻译大会机器翻译评测(CCMT 2022)的总体情况。在本次评测中，vivo 共参加了八个评测任务，5 个双语翻译评测任务，包括汉英新闻领域机器翻译（CE）、英汉新闻领域机器翻译（EC）、蒙汉日常用语领域机器翻译（MC）、藏汉政府文献领域机器翻译（TC）和维汉新闻领域机器翻译，1 个多语言机器翻译任务，即专利领域下的日英汉多语言翻译评测任务，以及中泰（CThai）和泰中（ThaiC）机器翻译两个低资源翻译任务。

本次各翻译语向采用 Transformer 神经网络机器翻译架构，在提升模型效果上运用模型平均、模型集成等方法；在数据处理上，结合各语向的语言特点和主办方提供的各数据集质量做了针对性的数据预处理和数据筛选。在受限任务上采用了单语数据回译的方法构建伪语料做数据增强，在非受限任务（中泰低资源任务）上额外搜集了网上的开源数据以及外部中文单语数据进行数据增强。在领域增强上，通过文本相似度方法召回与评测领域相关的语料微调模型。

2 系统介绍

图 2-1 给出了此次参评的技术方案整体流程图，大致可以分为数据过滤、数据预处理、数据增强、翻译模型训练、解码和后处理几个部分。其中数据过滤主要包括简单的规则清洗和使用对偶交叉熵方法的双语语料过滤方法；数据预处理主要包括繁转简、全角转半角、html 标签处理、标点统一、大小写转换、分词、BPE 等；数据增强主要通过文本表征模型来计算相似度，召回领域相关的数据微调模型，以提升模型在评测领域下的效果；翻译模型训练方面，主要采用 Transformer 及其变体模型；解码方面主要采用了模型平均和模型集成方法提升模型的翻译效果；后处理则对译文做去 BPE、去分词、去 tokenize 操作，最终还原符合语法规范的译文形式。流程图上的关键技术多数是在多个语向的评测任务中共用的，后文只在第一次使用的时候做详细介绍。全文实验评测指标均采用 BLEU-4，评测脚本 multi-bleu.perl[1]。

* **通讯作者：**fyli.winnie@gmail.com

[1] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

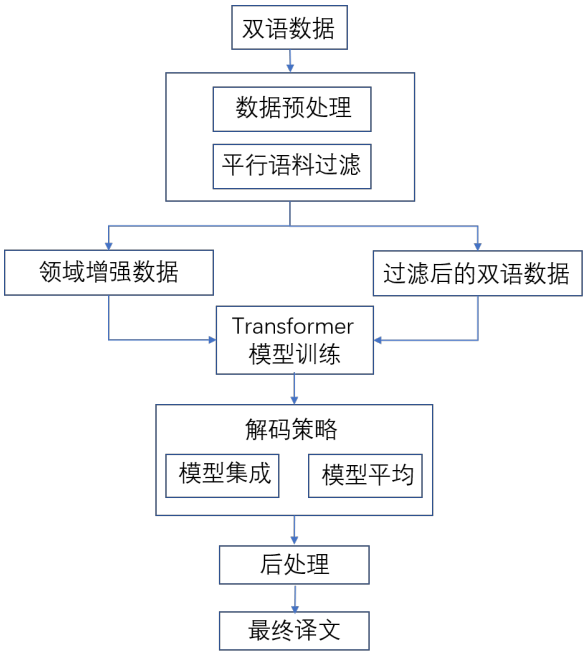

图 2-1 本次评测的整体流程图

Table 2-1 The overall flow chart of this evaluation

3 汉英双语翻译任务

本次汉英双语翻译任务使用的基线系统是基于自注意力机制的 Transformer-base 模型，改进的模型包括 Transformer-big、Transformer-deep (30-6)、Transformer-deep (24-12)这三种结构，并应用于后续的模型集成，具体使用的提升模型效果的策略包括双语数据过滤、领域微调、模型平均和模型集成解码等方法。

3.1 数据预处理

3.1.1 过滤、清洗

1) **规则方法**

长度比过滤方法：根据数据集的具体情况，长度比过滤阈值在 2.5~3 之间调整；

删除过长的语料：删除长度超过 200 个 token 的中文和英文语料；

删除有效字符占比过低的语料：中文语料中汉字、数字、英文、常见标点算有效字符，英文中数字、英文、常见标点算有效字符，删除有效字符占比小于 0.6 的语料；

删除中文里英文占比过高的语料：针对英汉语向训练数据，过滤掉英文占比超过 0.4 的语料，针对短句子做适当的阈值放松；

删除英文里中文占比过高的语料：针对汉英语向训练数据，过滤掉中文占比超过 0.2 的语料；

删除包含过多重复子串的语料：删除句中包含重复子串出现超过 5 次的语料，重复子串排除标点、数字的情况。

对 CCMT 2022 和 WMT 2022 汉英数据集采用规则方法过滤，过滤前后数据条数参考表 3-1。

表 3-1 规则过滤数据情况

Table 3-1 The data filtered by rules

数据集名称	条数	清洗后条数
backtrans-news	19 763 867	17 082 653
UNv1.0	15 886 041	15 540 489
paracrawl	14 170 585	12 375 439
casict2015	2 036 834	2 029 479
neu2017	2 000 000	1 996 719
casict2011	1 936 633	1 930 232
casia2015	1 050 000	1 049 785
datum2015	1 000 004	983 631
wikititles	921 960	796 673
datum2017	1 000 004	724 578
wikimatrix	2 595 119	483 725
news-commentary	313 674	312 934

对于wikimatrix[1]数据，我们额外过滤了语料中边距值（Margin score，该分值在wikimatrix数据集中已提供）低于 1.05 的数据。数据经采样观察，我们发现 paracrawl 和 backtrans-news 两份数据质量较低，将其划分为低质量语料，合并去重后合计 29 458 092 条，剩余数据作为高质量语料，合并去重后合计 25 928 645 条（后简称 2500 万）。

2) **双语过滤方法**

针对 paracrawl 和 backtrans-news 两份数据中英双语句对存在较多不对应的情况，尝试使用 fast_align[2]进行词对齐过滤，但因高质量的双语数据规模仅有 2500 多万，训练出的词对齐模型做双语筛选效果并不理想，后尝试基于对偶条件交叉熵（Dual Conditional Cross-Entropy，DCCE）[3]的方法过滤语料。

使用经规则过滤清洗好的 2500 万句对高质量语料，训练汉英翻译模型和英汉翻译模型，对于语料库中的每个句子对(x, y)，我们计算双向翻译模型的交叉熵，具体计算方法如下，

$$dcce\ (x,y) = |H_A(y|x) - H_B(x|y)| + \frac{1}{2}(H_A(y|x) + H_B(x|y)) \quad (1)$$

$$adq(x,y) = \exp\ (-dcce\ (x,y)) \quad (2)$$

其中：

$$H_A(y|x) = -\frac{1}{|y|}\sum_{t=1}^{|y|} \log P_A(y_t|y_{<t}, x)$$

$$H_B(x|y) = -\frac{1}{|x|}\sum_{t=1}^{|x|} \log P_B(x_t|x_{<t}, y)$$

A、B 分别表示汉英翻译模型和英汉翻译模型。

公式（1）的第一项主要衡量两个语向的翻译一致性，趋于 0 时，表示双向翻译越一致；

第二项表示两个语向的负对数翻译概率，模型翻译概率越高，整体越趋于 0。通过公式（2）将 dcce 分数转换到(0,1]区间，*adq* 分数越高，数据质量越高，并以此过滤了分值靠后的 40%的数据。

表 3-2 双语过滤结果

Table 3-2 The result of parallel corpus filtering

数据集名称	过滤前条数	过滤后条数
backtrans-news	17 082 653	10 873 240
paracrawl	12 375 439	7 425 141

3.1.2 预处理

中文侧预处理包括：繁体转简体、全角转半角、中文标点统一，分词采用 pyltp 分词模型、BPE 采用 subword-nmt[2]。

英文侧预处理包括：全角转半角、英文标点统一，html 转义符还原，sacremoses[3]分词，BPE 同样采用 subword-nmt。

3.2 领域增强

为了让模型在评测领域上表现更好，我们针对新闻领域做了领域增强。通过在高质量 2500 万语料中检索与 CCMT 2022 在线评测集和离线评测集相似领域的数据微调翻译模型，增强模型在评测领域的表现。具体做法是，通过 SimCSE[4]模型，在 2500 万句对中的中文和英文数据，采用无监督方式分别训练的中、英文表征模型作为基础模型。检索的全量数据为 2500 万高质量句对，检索的种子数据是今年在线和离线合计 2 万条评测数据，先通过对每条数据用基础模型生成向量表征，使用开源向量检索引擎 NGT[5]进行相似度计算，召回相似度 top 1000 的数据作为领域相似语料，并过滤掉召回语料中余弦相似度低于 0.5 的语料。

3.3 模型方法

1) 模型训练

本次评测使用开源的 fairseq 框架训练，每类模型结构均采用 8 块 A100 显卡进行训练，batch 大小设置为 16 384，梯度累积更新系数为 4，学习率为 1e-3，warmup 步数为 4000，Dropout[6]设置为 0.2。在训练层数较深的 Deep 模型时，Encoder 和 Decoder 的层归一化前置。训练语料采用先分词后 BPE 子词切分的方法，其中 BPE 词表大小中英各设置为 5 万。对全量语料统一使用 fairseq 处理成二进制文件用于训练，其中模型词表规模中文为 6 万，英文为 5 万。

2) 领域微调

针对每个模型，我们首先在全部数据上训练 36 个 epoch，然后基于今年的离线在线评测集、历年的测试开发集作为种子召回 2500 万高质量语料中的相似领域数据微调模型，继续训练 18 个 epoch，学习率为 3.2e-4；最后再针对评测测试集原文，利用 Transductive ensemble learning[7]方法，基于多模型构造伪数据进一步训练 6 个 epoch 微调模型，学习率为 2.7e-4。

---

[2] https://github.com/rsennrich/subword-nmt
[3] https://github.com/alvations/sacremoses

3) **模型集成**

模型集成可以整合多个模型的概率分布，协同决策，提高整体的翻译质量。在汉英双语评测中，我们使用了模型平均和解码器集成两种策略。其中模型平均是对模型的 N 个检查点参数矩阵进行平均，合并成一个模型去生成译文。解码器集成则是在解码过程中，整合不同模型的概率分布预测译文词语。

4) **模型推理**

模型推理部分使用 fairseq 和 CTranslate2[4]进行解码，beam size 设置为 5，其中针对模型漏译导致译文长度比偏低的问题，两个语向的长度惩罚系数均设置为 1.2，针对模型重复翻译的问题设置重复惩罚系数为 1.05，最终使用解码器集成的方式输出推理结果。

3.4 实验结果

表 3-3 和 3-4 分别为汉英和英汉语向上，翻译模型在 newstest2020 评测集上的结果，Baseline 采用 Transformer-base 模型。可以看出，较深的 Transformer-deep 模型和较宽的 Transformer-big 模型，均比 base 模型有明显的提升，验证了增大 Encoder 和 Decoder 层数和模型的维度能够提升模型在翻译任务的表现。加入召回的领域数据进行微调，以及利用 Transductive ensemble learning 的方法来构造领域伪数据微调，能够提升模型在特定领域的表现，验证了微调方法本身及相似度引擎的有效性。最后利用经过领域微调的多个不同结构的模型进行解码器集成，能够进一步提升模型整体的性能，最终的汉英语向翻译结果相比 Baseline 提升 6.5 个点，英汉模型较 Baseline 提升 7.8 个点。

表 3-3 汉英语向在 newstest2020 上的结果

Table 3-3 Chinese-English BLEU evaluation results on the newstest2020 test set

实验	BLEU-4
Baseline	30.23
Transformer-deep 30-6	32.74
Transformer-deep 24-12	32.91
Transformer-big	32.58
+ 召回领域数据微调	34.66
+ Transductive ensemble	35.14
+ 多模型集成	**36.75**

表 3-4 英汉语向在 newstest2020 上的结果

Table 3-4 English-Chinese BLEU evaluation results on the newstest2020 test set

实验	BLEU-4
Baseline	37.42
Transformer-deep 30-6	40.18
Transformer-deep 24-12	40.09
Transformer-big	40.37
+ 召回领域数据微调	42.61
+ Transductive ensemble	44.26
+ 多模型集成	**45.20**

[4] https://github.com/OpenNMT/CTranslate2

4 藏汉、维汉、蒙汉翻译

该任务中，由于可使用的平行语料资源较少，我们首先使用有限的平行语料训练反向的汉藏、汉维、汉蒙翻译模型；再通过回译方法，将提供的汉语单语数据经反向模型翻译得到伪双语增强数据；最后，将伪数据与双语数据合并，训练得到最终的藏汉、维汉、蒙汉模型。

4.1 数据预处理

4.1.1 藏语-汉语、维语-汉语

1) **数据处理**：包括标点符号归一化、半角全角字符统一、html 转义符还原、汉语繁转简、特殊字符和不可见字符过滤、结尾标点不匹配修复等；
2) **数据过滤**：包括长度比过滤、有效字符（去除标点符号和其他语种字符）占比过滤、句中 token 长度上下限过滤、重复子串过滤等方法；
3) **分词**：中文采用 jieba[5]分词，藏语采用 TIP-LAS[8]分词，维语采用 polyglot[6]分词；
4) BPE：采用 subword-nmt 训练 BPE 码表，并分别应用于双侧文本。BPE 码表大小，反向翻译模型设置为 3 万，正向翻译模型为 5 万。

4.1.2 蒙语-汉语

1) **数据处理**：现行蒙古文 Unicode 编码中使用控制字符表达字的不同变形，如果不对其进行预处理，Moses 分词时会将带有控制字符的词分成很多个词。此情况一方面会造成数据稀疏并影响翻译过程中的词对齐效果，另一方面会使蒙古文句子长度增加，影响翻译质量和评测[9]。因此，我们去掉了出现次数较多的控制字符，Unicode 编码包括“\u180e”、“\u202f”、“\u200c”和“\u200d”。其他数据处理方法同 4.1.1；
2) **数据过滤**：同 4.1.1；
3) **分词**：中文采用 jieba 分词，蒙语采用 moses 分词；
4) BPE：采用 subword-nmt 训练 BPE 码表，并分别应用于双侧文本。BPE 码表大小，汉蒙模型设置为 1 万，蒙汉模型为 1.5 万。

4.2 模型训练

4.2.1 藏语-汉语

藏汉翻译任务的模型训练过程如下：

1) **汉藏反向模型训练：**使用过滤后的藏汉平行语料，基于 Transformer 训练汉藏翻译模型。其中，Encoder 层数 10， Decoder 层数 6，Dropout 为 0.2，采用标签平滑交叉熵（Label-Smoothed Cross Entropy）损失函数[10]与 Adam 算法进行优化，并在 checkpoint 中保存最后的 5 个模型参数用于模型平均；
2) **伪双语数据生成：**基于评测方提供的中文单语语料，输入汉藏翻译模型生成藏语译文，组成伪双语数据。为了提升最终模型的翻译鲁棒性，对于每条输入的汉语句子，我们分别使用 Beam Search 和 top-k 随机采样两种方法，生成两种不同的译文，从而构造更多的伪双语数据。同时，所有的伪双语数据也采用 4.1.1 中的方法做清洗、过滤等处理；
3) **藏汉正向模型训练：**使用预处理后的伪双语数据和评测方提供的平行语料（上采样处理），共同训练藏汉翻译模型。其中，Encoder 为 30 层，其他配置及模型平均方法均与

[5] https://github.com/fxsjy/jieba
[6] https://github.com/aboSamoor/polyglot

反向模型相同。所有的伪双语数据开头都添加了回译标签“<bt>”[11]。在解码阶段，我们提高了模型重复输出相同 token 的惩罚系数，防止译文中出现过多重复的子串；

4) **藏汉正向模型微调：**训练 20 个 epoch 后，采用 3.2 中领域增强方法基于离线评测集检索领域相似数据对模型进行微调，学习率为 3.2e-4，提升模型在评测领域的表现。

4.2.2 维语-汉语

维汉翻译任务的模型训练步骤与藏汉翻译基本一致，区别仅在第四步中，由于维汉平行语料的规模相比藏汉语料更小，故未进行领域筛选，在训练 10 个 epoch 后使用离线开发集完成模型微调，并设置 4 倍上采样。

4.2.3 蒙语-汉语

蒙汉翻译任务的模型训练步骤与藏汉翻译类似，区别在于 Dropout 设为 0.3，并采取早停策略防止过拟合，同时在训练过程中保存 BLEU 值最高的 5 个 checkpoint 以及迭代结束前的最后 5 个 checkpoint 用于模型平均，得到 avg_last 和 avg_best 模型，然后将 avg_last、avg_best 与 BLEU 最高的模型进行模型集成，得到最终结果。

4.3 实验结果

4.3.1 藏语-汉语和维语-汉语

表 4-1 为本次提交的翻译系统在藏汉、维汉翻译任务中的评测结果。所有评测均使用主办方提供的数据集，未采用其他额外数据和预训练模型等。对数据做了清洗过滤、预处理和去重。其中，Baseline 仅使用双语平行语料进行训练；在 Baseline 基础上加入回译数据增强，以及进行了领域数据微调。

表 4-1 藏汉、维汉实验结果

Table 4-1 Experimental results of Tibetan-Chinese and Uyghur-Chinese

评测任务	Baseline	+加入回译数据	+领域数据微调
藏汉	45.22	51.96	53.48
维汉	26.31	36.39	37.34

由表 4-1 可知，基于回译数据增强策略能大幅缓解低资源问题，使藏汉、维汉两个语向的译文质量都得到了显著提升。在此基础上，微调训练使最终的模型更接近真实语料的分布，也使输出译文的 BLEU 值得到进一步提升。

4.3.2 蒙语-汉语

表 4-2 蒙-汉模型对比

Table 4-2 Experimental results of Mongolian-Chinese models

实验	BLEU-4
Baseline	56.57
+ Dropout 0.3	56.81
+ Label-smoothing	59.44
+ 加入回译数据	59.74
+ Encoder 层数 30	61.29

+ 模型平均	61.41
+ 模型集成（主系统）	**61.60**
BPE 词表扩大为 5 万（对比系统）	61.13

由表 4-3 可知， Label-smoothing 策略和加深 Encoder 层数能有效提高翻译效果，BLEU 分别提高了 2.6 和 1.6，合适的 Dropout 参数也能提高模型效果。同时，模型平均和模型集成策略也能有效提高翻译质量。相对而言，加入大量的回译数据对于模型的增益并不理想，原因可能是低资源下翻译模型质量不佳，得到的回译数据质量较低。此外，扩大 BPE 词表到 5 万，模型并没有什么增益。最终将模型集成后的结果作为主系统、BPE 词表 5 万下的结果作为对比系统。

5 中泰低资源语言翻译

低资源语言翻译评测任务包含泰中翻译和中泰翻译两个子任务。因是非受限评测，本文额外引入了大规模英泰平行语料库 scb-mt-en-th-2022[7]的泰文单语语料及网络爬取的中文单语语料进行数据增强[12]，同样采用 Transformer 模型进行训练，最后采用模型平均和集成等策略来进一步提升泰中和中泰翻译模型在综合领域的翻译表现。

5.1 数据

5.1.1 数据集说明

在低资源评测任务中，平行语料严重匮乏，为缓解低资源所导致的翻译模型性能不佳的问题，本文除了使用 CCMT 2022 提供的中泰平行语料外，还使用了 VISTEC 提供的大规模英泰平行语料库 scb-mt-en-th-2020 的泰文单语语料，以及网络爬取的中文单语语料作为外部数据来进行数据增强，相关数据的详细情况如表 5-1 所示。

表 5-1 低资源语言翻译评测任务所使用的数据

Table 5-1 The data used in low-resource language translation evaluation task

数据集	数据类型	数据量	筛选后数据量
CCMT 2020	中泰/泰中双语	200 000	199 944
scb-mt-en-th-2020	泰文单语	1 001 752	993 121
网络爬取中文单语数据	中文单语	157 657 977	9 682 451

5.1.2 语料过滤、清洗

由于训练语料的数据质量参差不齐，在本次评测中，采用如下几种筛选策略。

1) 删除长度大于 200 个 token 的语料；
2) 删除长度比大于 3 的语料；
3) 删除原文与译文公共子串占比大于 0.6 的平行语料；
4) 使用语种识别模型，删除语种错误的语料；
5) 使用相似度引擎对单语数据评分，过滤与 CCMT 2022 平行语料相似度低的语料。

5.1.3 数据增强

在低资源情况下，双语平行语料匮乏是影响翻译质量的瓶颈[13]。本文选用谷歌翻译[8]对

[7] https://huggingface.co/datasets/scb_mt_enth_2020
[8] https://translate.google.cn/

大规模英泰平行语料库 scb-mt-en-th-2020 的泰文单语语料以及网络爬取的中文单语语料进行回译，并将其应用于中泰和泰中语向的翻译模型。回译生成的伪平行语料如表 5-2 所示。

表 5-2 回译的伪平行数据

Table 5-2 Pseudo-parallel data generated by back translation

数据集	数据类型	数据量
scb-mt-en-th-2020 回译数据	中泰/泰中双语	993 121
网络爬取中文单语回译数据	中泰/泰中双语	9 682 451

5.1.4 数据预处理

在训练翻译模型之前，需要对训练数据进行预处理，进一步提高语料质量，并且将语料处理成翻译模型标准输入的格式。在本次评测中，本文采用的数据处理方法与 4.1.1 相同，中文使用 jieba 分词，泰文使用 polyglot[9]分词，subword-nmt 训练 BPE，词表大小 1.5 万。

5.1.5 构建数据集

本文将经过预处理后的数据按 19:1 划分训练集和验证集。由于本次低资源语言翻译评测任务并没有提供开发集或测试集，所以本文将 CCMT 2022 提供的中泰和泰中语向离线评测集通过谷歌翻译、阿里翻译[10]和火山翻译[11]等引擎构建多个机器翻译参考译文，进行多参考答案的 BLEU 评测。最终构建的训练集、验证集和测试集如表 5-3。

表 5-3 训练集、验证集和测试集

Table 5-3 Training, validation, and test data sets

数据集类型	数据量
训练集	10 331 901
验证集	543 615
测试集	10 000

5.2 模型方法

泰中中泰翻译任务均采用 Transformer 模型训练，Encoder 为 30 层，Decoder 为 6 层，Dropout 设为 0.2，采用了 Adam 算法进行优化，并且采取早停策略防止过拟合。同时，由于数据集中含有大量地道性和流畅性较差的伪平行语料，本文采用 label-smoothing 策略来降低模型对标签的信任程度，从而提高模型的鲁棒性。另外，为了进一步提升模型的泛化能力，本文采用模型平均策略将训练过程中表现最好的 5 个模型的参数进行平均获得 top_avg 模型，并将训练过程中最后 5 个模型的参数进行平均获得 last_avg 模型，最后将表现最好的 3 个模型与 top_avg 模型和 last_avg 模型进行模型集成，获得最终的翻译模型。

5.3 实验结果

本节介绍了不同方法下模型的结果，按字符评测，实验结果如表 5-4 所示，可以看出，label-smoothing 策略、加深 Encoder 层数均能带来 2~3 个点提升。回译数据增强策略能大幅缓解低资源问题，显著提升了模型的翻译效果，BLEU 提升 6~7 个点。除此之外，模型平均和模型集成带来的提升效果接近，最终选择模型集成后的模型作为主系统，模型平均后的模

[9] https://github.com/aboSamoor/polyglot
[10] https://translate.alibaba.com/
[11] https://translate.volcengine.com/translate

型作为对比系统。

表 5-4 泰中-中泰实验结果

Table 5-4 Thai-Chinese and Chinese-Thai experimental results

参数	中泰模型	泰中模型
基线模型	52.34	36.20
+ Label-smoothing	54.04	38.02
+ Encoder 设置 30 层	57.59	40.46
+ 数据增强	64.54	46.12
+ 模型平均	66.31	47.84
+ 模型集成	**66.65**	**48.22**

6 日汉英多语种翻译

本任务的目标是使用汉英语料和汉日语料，构建日英翻译模型。由于日英平行语料未直接提供，我们采用回译的方法，首先基于已有平行语料，分别训练汉日、汉英语向翻译模型；然后，将汉英平行语料的汉语侧通过汉日模型翻译成日语，并与汉英平行语料的英语侧组成反向回译数据，将汉日平行语料的汉语侧通过汉英模型翻译成英语，并与汉日平行语料的日语侧组成正向回译数据；最后，合并所有的回译数据作为日英伪双语训练集，训练得到最终的日英翻译模型。

6.1 数据预处理

1) 对平行语料中的日语、汉语、英语侧数据分别进行清洗，包括汉繁转简，标点统一、全角转半角、html 转义符处理、特殊字符和不可见字符修正、结尾标点不匹配修复等。
2) 对汉英、汉日平行语料分别进行双语过滤，过滤方法同 3.1.1 节。
3) 中文采用 jieba 分词，日语采用 mecab[12]分词，英文采用 moses tokenize。
4) BPE 采用 subword-nmt，并分别应用于日语、汉语、英语侧文本。本任务中，BPE 码表大小均设置为 5 万，且汉英、汉日、日英等各个语向的码表相互独立。

6.2 模型训练

多语种翻译任务的模型训练过程如下：

1) **汉日、汉英语向模型训练。**利用评测方提供的平行语料，基于 Transformer 架构训练基础翻译模型。其中，Encoder 层数为 30，Decoder 为 6 层，Dropout 设置为 0.2，采用标签平滑交叉熵作为损失函数，并使用 Adam 算法进行优化。
2) **日英正向、反向回译数据生成。**将平行语料中的汉语侧通过翻译模型生成日语或英语译文，从而和平行语料的另一侧组成日英伪双语数据。其中，日语侧语料和英语译文组成正向回译数据，英语侧语料和日语译文组成反向回译数据。正向回译和反向回译分别采用 Beam search 和 top-k 随机采样两种方法生成译文。所有的回译数据也采用 6.1 中的方法，进行了清洗、过滤等处理。
3) **日英语向模型训练。**使用所有的回译数据进行训练，模型参数配置与基础翻译模型相同。在解码阶段，我们提高了模型重复输出相同 token 的惩罚系数，防止译文中出现过多重复的子串。为了提升模型的鲁棒性，我们还在回译数据的基础上构造了增强语料，将日语侧句子中无实际意义的助词以一定概率随机遮盖，并在模型微调阶段加入训练。

[12] http://taku910.github.io/mecab/

4) **日英语向模型微调。**采用 3.2 中领域增强方法，从评测方提供的平行语料中召回与开发集语料领域相似的数据，对模型进行微调，使模型更好地学习真实的领域数据分布。

6.3 实验结果

下表为本次提交的翻译系统在多语种翻译任务中的评测结果。所有评测均使用主办方提供的对应语向开发集数据，并进行了合并去重、清洗过滤等预处理。其中，Baseline 表示模型使用伪双语回译数据进行训练，然后在 Baseline 基础上使用了领域数据进行微调。各模型的 BLEU 值结果如表 6-1 所示：

表 6-1 日英模型实验结果

Table 6-1 Experimental result of Japanese-English model

模型	BLEU-4
Baseline	44.06
+领域数据微调	44.91

7 总结

本文详细介绍了 vivo AI 研究院机器翻译团队参加 CCMT 2022 的总体情况。vivo 共参加了汉英新闻领域机器翻译、英汉新闻领域机器翻译、蒙汉日常用语领域机器翻译、藏汉政府文献领域机器翻译、维汉新闻领域机器翻译、专利领域下的日英汉多语言翻译，以及中泰和泰中机器翻译，合计八个翻译评测任务。各任务采用 Transformer 机器翻译架构，做了语料清洗、数据预处理、数据增强、领域微调和模型平均、模型集成等方法提升模型的整体表现。由于时间有限，还有很多方法未来得及尝试和做对比实验，希望在今后持续精进技术，对机器翻译行业有所贡献。

8 参考文献

[1] Schwenk H, Chaudhary V, Sun S, et al. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia[J]. arXiv preprint arXiv:1907.05791, 2019.

[2] Dyer C, Chahuneau V, Smith N A. A simple, fast, and effective reparameterization of ibm model 2[C]//Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013: 644-648

[3] Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

[4] Gao T, Yao X, Chen D. Simcse: Simple contrastive learning of sentence embeddings[J]. arXiv preprint arXiv:2104.08821, 202

[5] Iwasaki M, Miyazaki D. Optimization of indexing based on k-nearest neighbor graph for proximity search in high-dimensional data[J]. arXiv preprint arXiv:1810.07355, 2018.

[6] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958

[7] Wang, Y., Wu, L., Xia, Y., Qin, T., Zhai, C., & Liu, T.-Y. (2020). Transductive Ensemble Learning for Neural Machine Translation. Proceedings of the AAAI Conference on Artificial Intelligence, 34(04), 6291-6298.

[8] 李亚超，加羊吉，江静，等. 融合无监督特征的藏文分词方法研究[J]. 中文信息学报, 2017, 31(02):71-75.
[9] 李金廷,侯宏旭,武静,王洪彬,樊文婷. 语料预处理对蒙古文-汉文统计机器翻译的影响[J]. 计算机科学, 2017, 44(10): 259-264
[10] Müller R, Kornblith S, Hinton G E. When does label smoothing help?[J]. Advances in neural information processing systems, 2019, 32.
[11] Caswell I, Chelba C, Grangier D. Tagged back-translation[J]. arXiv preprint arXiv:1906.06442, 2019.
[12] Lowphansirikul L， Polpanumas C， Rutherford A T，et al. scb-mt-en-th-2020: A Large English-Thai Parallel Corpus[J]. 2020.
[13] Haddow B， Bawden R， Barone A，et al. Survey of Low-Resource Machine Translation[J]. 2021.

Evaluation Technical Report for CCMT by vivo AI Lab

Fangyuan Li[1]*, Panfeng Guo[1], Yachao Di[1], Chengzhi Wang[1], Jiajie Ding[1,2],
Chunyuan Li[1], Fei Teng[1]
（1. vivo AI Lab, Hangzhou 311121, China；
2. Soochow University，Suzhou 215006, China）

Abstract: This paper describes the participation of vivo AI Lab in the 18th China Conference on Machine Translation (CCMT 2022) and the technical details of evaluation tasks. vivo has participated in eight evaluation tasks in total: Chinese-English & English-Chinese machine translation in news respectively, Mongolian-Chinese machine translation in daily language, Tibetan-Chinese machine translation in government literature, and Uyghur-Chinese machine translation in news, multilingual machine translation in patents involving Japanese, Chinese, and English, and Chinese-Thai & Thai-Chinese machine translation in low-resource scenario. This report describes the algorithm model framework, and data processing, data screening, and data enhancement methods adopted in the evaluation, and conducts a comparative analysis on evaluation results of the evaluation system under different settings.

Keywords: Machine Translation; CCMT 2022; Translation Evaluation; vivo