文章编号:1000-5862(2019)06-0630-08

民汉稀缺资源神经机器翻译技术研究

赵 阳,周 龙,王 迁,马 聪,刘宇宸,王亦宁,向 露,张家俊,周 玉,宗成庆
(中国科学院自动化研究所,北京 100190)

摘要:该文介绍了中国科学院自动化研究所参加第 15 届全国机器翻译大会(CCMT2019)翻译评测任务总体情况以及采用的技术细节. 在评测中,中国科学院自动化研究所参加了 3 个翻译任务,分别是蒙汉日常用语机器翻译、藏汉政府文献机器翻译以及维汉新闻领域机器翻译;阐述了参评系统采用的模型框架、数据预处理方法以及译码策略;最后给出了不同设置下评测系统在测试数据集上的表现,并进行了对比和分析.

关键词:神经机器翻译;低资源翻译;自注意力机制
中图分类号:TP 302.1 **文献标志码**:A **DOI**:10.16357/j.cnki.issn1000-5862.2019.06.12

0 引言

本文介绍了中国科学院自动化研究所参加第 15 届全国机器翻译大会(CCMT2019)翻译评测任务的情况. 笔者共参与了 CCMT2019 中 3 个有关少数民族语言的翻译项目,分别是蒙汉日常用语机器翻译、藏汉政府文献机器翻译和维汉新闻领域机器翻译,这 3 种翻译方向均属于民汉稀缺资源语言的翻译.

本次评测采用的基线系统为谷歌 Transformer[1] 神经网络机器翻译架构. 为了提高该模型效果,在评测中利用多策略的模型作为“教师”模型,并采用句子级知识蒸馏方法提高基线“学生”模型的翻译性能. 在数据预处理方面,本次评测采用亚词[2] 处理方法切分训练数据,以提高神经机器翻译对于低频词的翻译效果. 为了减少语料噪声对于翻译性能影响,本次评测探索了多种不同语料过滤方法以提高训练语料的质量. 同时为了进一步有效地利用目标端的单语数据,本次评测使用了反向翻译方法来构建伪平行数据,对神经翻译模型的训练集进行了补充. 在最终的译文输出过程中,采用了模型平均和集成解码的策略,并利用重评分策略给出最后的译文. 在实验中,对比了系统在 3 种翻译方向上不同设置下的表现,并对实验结果进行了分析.

1 系统介绍

从整体而言,图 1 给出了本次评测的整体流程图. 下面分别从模型结构、数据处理以及译码策略等方面进行介绍.

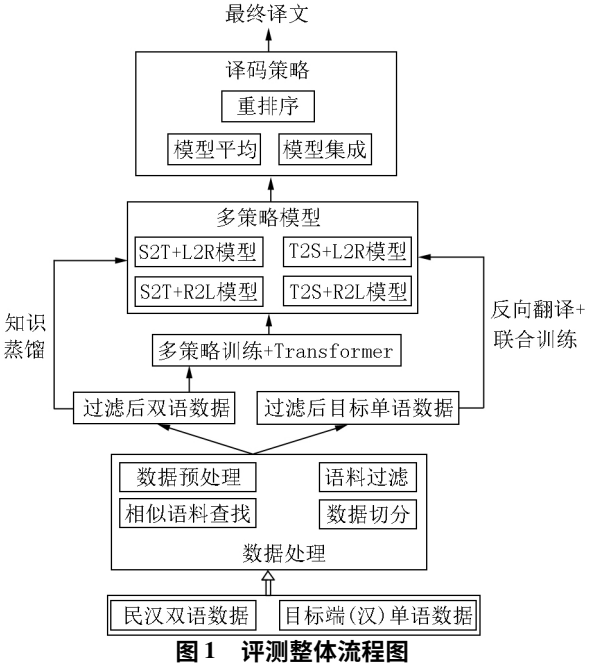

图 1 评测整体流程图

收稿日期:2019-09-07
基金项目:国家重点研发计划(2016QY02D0303),国家自然科学基金(U1836221)和北京市科技计划(Z181100008918017)资助项目.
作者简介:赵 阳(1990-),男,山西运城人,助理研究员,博士,主要从事机器翻译、自然语言处理研究. E-mail: yang.zhao@nlpr.ia.ac.cn

1.1　模型结构

与 2018 年相同[3],本次评测使用的基线系统是基于自注意力机制的 Transformer 模型. 基于自注意力的 Transformer 模型结构如图 2 所示. Transformer 模型同样也包含编码和解码 2 个部分. 与之前方法不同的是,该模型并未采用循环神经网络[4]或者卷积神经网络[5],而是完全基于注意力机制实现. 基于自注意力机制的模型能够在实现算法并行性、加快模型训练速度的同时,进一步缓解长距离依赖,并提高翻译质量.

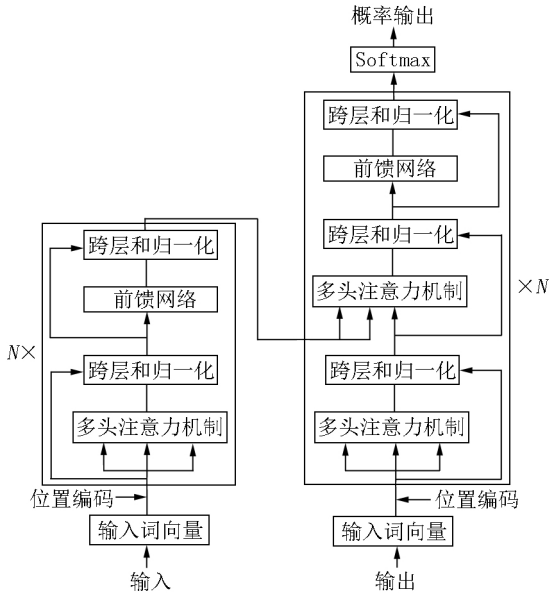

图 2　基于自注意力机制的 Transformer 模型结构

编码器和解码器由 N 个层块堆叠而成,其中编码器的每个层块包含 2 个子模块,分别是多头自注意力模块和前馈神经网络模块,这里多头自注意力模块将隐状态的维度划分为多个部分,每个部分分别使用自注意力函数计算得到,然后将这些输出向量拼接起来. 多头的作用是使模型能够更大程度地关注到不同位置表示子空间的不同特征信息. 多头注意力方法包括 2 个步骤: (i) 点积注意力计算; (ii) 多头注意力计算. 点积注意力的计算方式为

$$A_{ttention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = s_{oftmax}(\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}/\sqrt{d_k})\,\boldsymbol{V},$$

其中 $\boldsymbol{Q}$ 为查询向量,$\boldsymbol{K}$ 为键向量,$\boldsymbol{V}$ 为值向量,d_k 为隐层状态的维度. 在点积注意力的基础上,多头注意力机制的计算方式为

$$M_{ultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = C_{oncat}(h_{ead_1},\cdots,h_{ead_h})\,\boldsymbol{W}^0,$$

其中 $\boldsymbol{W}^0$ 为矩阵参数,每个头的注意力值为

$$h_{ead} = A_{ttention}(\boldsymbol{Q}\boldsymbol{W}_i^{\boldsymbol{Q}},\boldsymbol{K}\boldsymbol{W}_i^{\boldsymbol{K}},\boldsymbol{V}\boldsymbol{W}_i^{\boldsymbol{V}}).$$

解码器每个层块由 3 个子模块构成,除了编码器中的 2 个模块外,在这 2 个模块之间另外加入了一个解码器-编码器的注意力模块,这一模块是用于在解码单词时关注源端信息. 为了避免层数过多导致模型难以收敛的问题,编码器与解码器均使用了残差连接和层级正则技术,使得模型更好地获得输入序列信息的位置信息,在编码器和解码器的输入层中均加入了额外位置编码向量.

在编码器得到隐层状态后,Transformer 模型将该隐层状态输入 Softmax 层并与候选词表进行评分,得到最终的译文结果.

给定训练数据 $D = \{X^{(d)},Y^{(d)}\}_{d=1}^{|D|}$,其中 X 为源语言,Y 为目标语言,$|D|$ 为语料中双语数据的个数. 神经机器翻译模型通过优化以下目标函数来得到翻译模型 θ. $L(\theta,D) = \frac{1}{|D|}\sum_{n=1}^{|D|}\log p(Y^{(n)} \mid X^{(n)};\theta)$,该训练方式被称为最大似然方法.

1.2　知识蒸馏

现有的 Transformer 模型解码方式为从左到右 (L2R) 的解码方法,并且从源端句子解码得到目标端句子(S2T),实验表明这种解码策略会存在输出结果不平衡的问题,如从左向右生成译文会导致右侧译文的准确率较差. 在本次评测中,为了充分利用其他解码模型的优势以及所学习到的翻译知识,采用知识蒸馏的方式来提高原始模型的翻译性能. 知识蒸馏[6]是一种知识迁移的方法,其主要是充分利用“教师”模型的预测分布来指导“学生”模型的参数学习.

针对 NMT 的特点,本次评测采用了句子级的知识蒸馏方法,具体的知识蒸馏过程包括 3 个步骤: (i) 多策略模型学习; (ii) 蒸馏译文生成; (iii) 混合训练. 下面分别进行介绍.

(i) 多策略模型学习(教师模型). 在本次评测中,首先通过以下 4 种策略训练 4 个教师模型: (a) S2T + L2R 模型,即解码方向为源端到目标端且从左到右生成译文的神经机器翻译模型; (b) S2T + R2L 模型,即解码方向为源端到目标端且从右到左生成译文的神经机器翻译模型; (c) T2S + L2R 模型,即解码方向为目标端到源端且从左到右生成译文的神经机器翻译模型; (d) T2S + R2L 模型,即解码方向为目标端到源端且从右到左生成译文的神经机器翻译模型.

(ii) 蒸馏译文生成. 在得到上述 4 个教师翻译

模型后,利用教师模型对训练数据分别进行解码,得到各自的解码译文,并与各自的输入句子构成知识蒸馏的双语句对.

(iii) 混合训练. 蒸馏的最后一步是将蒸馏译文与原始译文进行混合训练,这样在混合的双语数据中,除了包含原始的训练数据外,还包含各自教师模型的预测结果. 最后利用混合的训练数据,并采用最大似然估计的方法训练得到学生模型.

2 数据处理

2.1 语料预处理和切分

本次评测使用了 CCMT2019 提供的从少数民族到汉语翻译的平行语料,其中蒙汉平行语料为日常用语领域,藏汉平行语料为政府文献领域,而维汉平行语料为新闻领域. 3 种语言对的特点既相似又不同,为此对 3 种语言对的处理采用 2 阶段处理法: 通用预处理阶段和特定预处理阶段,以得到更好的语料预处理结果.

在通用预处理阶段中,本次评测进行了如下的一系列预处理操作: 重复句的删除、数字及标点符号的全半角处理、转义字符处理、分词操作、大小写转换、长句切分,其中源端语言(蒙语、藏语、维语)和目标端语言(汉语)的分词均采用实验室开发的词法工具 Urheen(https://www.nlpr.ia.ac.cn/cip/software.html)来进行实现.

机器翻译用语训练的平行语料质量对最终训练得到的机器翻译模型性能有非常大的影响. 根据分析,评测中提供的未经处理的翻译语料存在较大的噪声,这将严重影响机器翻译的性能. 且训练语料同验证集存在一定的领域不适应性,这将导致训练测试不一致问题的发生,也会造成机器翻译性能的下降. 基于上述存在的问题,进行了特定预处理阶段的操作,即对于藏汉翻译平行语料进行了过滤来缓解低质量的语料对翻译结果造成的影响. 对原始数据中的平均句长、源端-目标端的长度比例进行统计分析,统计信息如表 3 所示.

表 3 藏汉数据统计

句长统计	训练集		验证集	
	源端	目标端	源端	目标端
平均句长	12.75	10.05	7.31	8.84
句长比例	1.340		1.360	

根据统计结果,对数据进行长度和长度比过滤. 对于长度过滤,设置最短长度为 1,最大长度为 50. 对于句长比过滤,设置最小长度比例为 0.31,最大长度比例为 5.16.

再根据词对齐对语料进一步过滤,删除源端和目标端不对应的句对. 这里使用 fastalign 词对齐工具[7],并统计每个句对中对齐词的比例,按照对齐比例阈值 0.4 进行语料过滤. 过滤结果如表 4 所示.

表 4 藏汉过滤情况 个

过滤方式	平行句对数目
原始	156 580
句长过滤	155 804
长度比例过滤	155 654
词对齐过滤	155 544

此外,对于藏汉的数据,通过分析后发现验证集中的句长分布集中在较少的词语. 因此,本文对藏汉的双语平行训练语料进行了长度切分,以保证训练数据的长度基本与开发集长度保持相符.

对于蒙汉和维汉 2 个任务,同样进行了句长和比例统计,但未对语料进行过滤. 统计结果如表 5 和表 6 所示.

表 5 蒙汉数据统计

句长统计	训练集		验证集	
	源端	目标端	源端	目标端
平均句长	27.63	18.79	44.24	29.27
句长比例	1.480		1.450	

表 6 维汉数据统计

句长统计	训练集		验证集	
	源端	目标端	源端	目标端
平均句长	19.40	20.47	25.41	27.19
句长比例	0.978		0.976	

对于蒙汉翻译平行语料,发现存在源端、目标端语言混杂的情况,即在源端语言的训练文件中存在大量的目标端语言句子,反之亦然. 针对该情况,利用语种检测删除了这些语种混杂的情况,由原始的 261 457 个平行句对得到处理后的255 822个,共删除 5 635 个句对.

通过训练语料通用预处理和特定预处理 2 阶段,得到的训练语料平行句对规模如表 7 所示.

表 7 训练数据总数 个

训练项目	藏汉	蒙汉	维汉
句对数目	155 544	255 823	170 061

2.2 相似语料过滤

为了进一步提高训练数据、验证数据和测试数据的一致性,本文分别根据验证数据和测试数据的

源端,从训练数据中过滤出相似的语料,对后期模型进行优化. 具体地,对于测试数据中的每个句子,首先根据 Dice 距离,分别从训练数据中粗选一定数量的相似语料. 之后从粗选的结果中,再根据源端句子的编辑距离,选取一定数量的相似语料. 对于验证数据,Dice 距离选择的数目设为 500 个,编辑距离选择的数目设为 10 个,并且在计算距离的过程中忽略前 500 个高频词. 对于测试数据,Dice 距离选择的数目设为 20 个(蒙汉)和 100 个(藏汉、维汉),编辑距离选择的数目设为 1 个,并且在计算距离的过程中忽略前 500 个高频词. 数据选择结果如表 8 所示.

表 8　相似语料选择　　个

训练项目	藏汉	蒙汉	维汉
验证数据	10 000	10 000	10 000
测试数据	1 000	1 001	1 000

2.3　单语数据利用

在 CCMT2019 的评测中为少数民族到汉语的翻译提供了统一的汉语单语增强语料. 由于蒙语、藏语、维语到汉语的翻译语料分别为日常用语领域、政府文献领域和新闻领域,所以用同样的单语语料对 3 种语言对的机器翻译增强并不能达到最好的语料增强效果. 于是,本文根据 3 种语言对分别训练了 3 个汉语语言模型来对单语语料进行过滤和筛选. 针对语言模型,这里采用了 srilm 语言模型来分别对训练集中的 3 个汉语单语语料训练了语言模型[8],并利用训练得到的语料模型在单语数据上进行评分,根据评分的结果排序,为每一种语言对的机器翻译筛选单语增强语料.

具体地,对于蒙汉的翻译任务,本文筛选了 35 万小数据集,200 万大数据集; 对于藏汉的翻译任务,筛选了 25 万小数据集,100 万大数据集; 对于维汉的翻译任务,筛选了 35 万小数据集,200 万大数据集. 对于筛选得到的单语数据,本文采取了回翻策略构造伪平行数据来增强机器翻译结果. 根据评测提供的少数民族到汉语的平行语料构建汉语到少数民族语言的翻译模型,继而通过该模型将语言模型评分后抽取的汉语单语语料翻译到对应的少数民族语言.

通过实验发现,仅仅通过回翻得到的伪平行语料并不能有效地提高少数民族语言到汉语的机器翻译质量. 基于该分析,本文对回翻的过程采取了文本加噪[9]的方式提高回翻语料的质量和回翻伪平行语料使用过程中的鲁棒性. 在本文的加噪操作中,共使用了 3 种加噪策略:

(i) 删除策略,以概率 $p=0.1$ 删除句子中的部分词语;

(ii) 替换策略,以概率 $p=0.1$ 用特殊词语替换原句中的词语;

(iii) 顺序替换策略,以概率 $p=0.1$ 将原句中的 2 个词语进行顺序替换.

在加噪实验中,本文尝试了在汉语单语语料中添加噪声以及回翻后的少数民族语言中添加噪声. 由于汉语单语语料回翻后得到的伪平行数据规模远远大于评测提供的双语数据,所以本文对评测的双语资源进行上采样来提升高质量双语平行语料在机器翻译训练过程中的作用. 最终将加噪完成后的回翻伪平行数据与上采样后的评测提供的高质量双语平行语料混合进行共同训练,利用汉语单语资源的机器翻译模型,并后续与其他实验设置的机器翻译模型进行模型融合.

3　译码策略

3.1　模型平均与模型集成

为了减少模型参数不稳定性,提升模型鲁棒性,本次评测使用了模型平均技术. 模型平均是指将同一模型在训练不同时刻中保存的参数进行平均得到更加鲁棒的模型参数. 这里保存的参数通常是在模型基本收敛时对应的最后 N 个时刻的参数. 图 3 给出了模型平均的示意图.

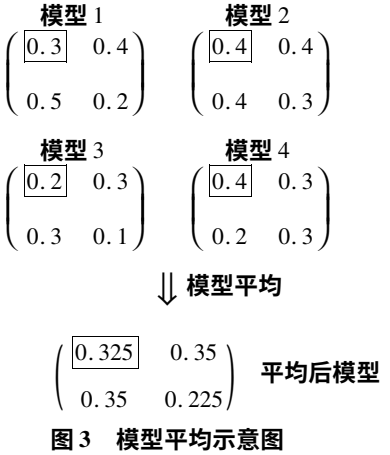

图 3　模型平均示意图

模型集成作为一种增强模型鲁棒性的方法可以进一步提升模型性能. 该方法将多个训练完成的模型集成到一个统一的大模型中,在神经机器翻译模型中预测当前时刻目标端语言单词时,使用多个模

型分别预测当前单词的概率分布,进而将多个模型的输出概率分布进行加权平均,以联合预测最终输出,如图 4 所示. 用于做集成译码的模型可以是同一模型在训练的不同时刻保存的模型,也可以是模型结构相同但是参数初始化方式不同的模型,或者是模型结构和初始化方式均不同的模型. 一般而言,最后一种方式更具有差异性,能够带来更大的质量提升[10-11].

同时,考虑到引入更多的差异性,增加模型的数目同样能带来一定程度上的性能提升. 本文主要在不同训练数据上得到的和不同结构的多个模型上进行了集成操作.

本文设计了一种集成方式,将不同结构的基线模型、增加单语数据的模型、增加对抗噪声样本的模型、对训练数据进行上采样的模型以及在开发集数据上调优的模型等多个模型进行集成操作,从而得到最终的翻译结果.

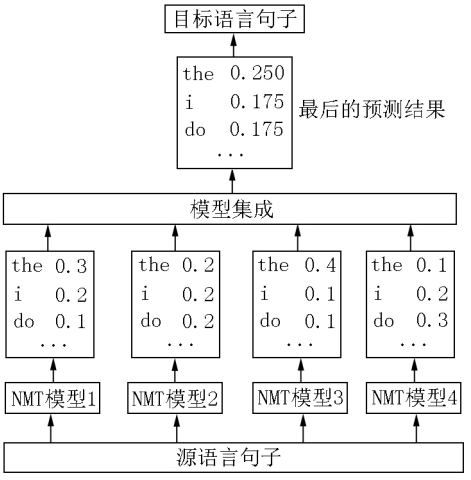

图 4 模型集成示意图

3.2 重排序

神经机器翻译模型通常采用从左到右的解码方式,容易受到曝光误差的影响,面临着不平衡输出以及错误累积的问题,导致译文质量随着长度增加而下降[12-13]. 译文在生成的过程中,若前几个时刻产生错误,则后续很难再产生正确的结果. 这一问题在一定程度上可以通过增大柱搜索的搜索空间来缓解. 然而增大搜索空间会显著降低了解码效率,并且若仅根据柱搜索的预测概率选择预测概率最大的结果作为最终输出,则增大搜索空间带来的收益并不明显,有时甚至还会带来一些性能损耗[14].

为此,本文使用了重排序的方法对柱搜索中的候选结果进行选择. 先训练了多个评分模型作为对候选译文进行质量评估,其中包括 N 个民语到汉语的正向模型、N 个汉语到民语的反向模型、目标端从左到右的模型、目标语言的神经语言模型、目标语言 n-gram 语言模型、候选译文与源端输入的长度比、正反向翻译对齐概率与翻译覆盖率,以及候选译文之间的贝叶斯风险概率等[15]. 再利用贝叶斯风险概率计算不同翻译候选译文之间的相似程度,该得分越低意味着该译文与候选空间中绝大多数译文更为相似,进而规避了罕见词或者错误词语的产生. 然后利用这些特征,借助 Z-MERT(http://cs.jhu.edu/~ozaidan/zmert/) 开源工具包在开发集上训练得到各个特征的最优权重值. Z-MERT 采用了最小错误率训练的方法,每次选择一个特征并固定其他权重计算候选集的错误率,进而对该权重值进行更新,在训练过程中不断迭代优化,最终得到全部特征的最优权重值. 最后,根据学习到的特征权重,对测试集中的候选译文进行评分并排序[16],选择得分最高的译文作为最终的输出译文.

4 实验

4.1 参数设置

本次评测系统在开源项目 tensor2tensor(https://www.github.com/tensorflow/tensor2tensor) 上加以修改. 参数设置如下: 每个模型使用 4 块 GPU 核进行训练,每个 batch 大约含有 4 096 个中文 token 和英文 token,模型训练 20 万 steps,每 30 min 保存一次模型用于之后的模型平均. 不同初始化保存 20 个模型进行模型平均. 词向量的维度为 1 024,隐层状态维度为 4 096,编码器与解码器均为 6 层,多头自注意力机制使用 16 个头. 本次评测采用了 dropout 机制,dropout 设为 0.3.

训练语料在蒙汉翻译、维汉翻译以及藏汉翻译中均采用 BPE 切分,其中源语言蒙语、维语以及藏语的词表大小均为 30 000,目标语言的词表也设定为 30 000. 2 者的词表不共享但是词向量共享. 使用 Adam 梯度优化算法来训练得到最终的模型参数,其中 $\beta_1 = 0.90$,$\beta_2 = 0.98$ 以及 $\varepsilon = 1 \times 10^{-9}$. 初始学习率为 0.1,warmup 步数设定为 8 000.

4.2 实验结果

表 9 ~ 表 11 分别在蒙汉翻译、维汉翻译以及藏汉翻译测试集上给出了本次提交的 3 个系统的评测结果,该结果采用的评测指标均是大小写敏感的,其中主系统是本次评测主系统的实验结果,使用了 6

个模型集成,beam size 大小为 50,经过重评分的方法得到的最终结果. 对比系统 1 则是使用了重评分方法中最小贝叶斯风险解码策略解码得到的. 对比系统 2 则是未利用单语数据的系统所取得的实验结果.

根据表 9 ~ 表 11 的实验结果,将主系统和对比系统 1 进行对比,可以看出不同的重评分策略对结果的影响有所不同,最小贝叶斯风险解码比使用多特征融合的方法有更大地提升. 将主系统和对比系统 2 进行对比,单语数据可以有效提升翻译系统的翻译性能. 接下来通过对开发集的实验结果进行分析来测试不同模块的作用.

4.2.1　基本分析　首先对比了不同单词切分粒度对实验结果的影响,目前切分粒度主要包括以下 3 种粒度: 以亚词(BPE) 为翻译单元, 以字(character) 为翻译单元和以 WordPiece 为翻译单元. 3 种切分粒度在藏汉开发集上的结果如表 12 所示.

表 9　在蒙汉翻译测试集上的实验结果

模型	BLEU5-SBP	BLEU5	BLEU6	NIST6	NIST7	GTM	mWER	TER
主系统	0.588 2	0.617 9	0.596 7	10.397 9	10.405 2	0.790 2	0.322 4	0.296 8
对比 1	0.462 0	0.484 1	0.453 5	9.300 9	9.308 0	0.731 6	0.401 7	0.370 9
对比 2	0.550 7	0.576 8	0.554 4	9.959 6	9.966 4	0.762 3	0.360 8	0.329 4

表 10　在维汉翻译测试集上的实验结果

模型	BLEU5-SBP	BLEU5	BLEU6	NIST6	NIST7	GTM	mWER	TER
主系统	0.603 8	0.622 1	0.594 3	10.607 0	10.639 0	0.814 6	0.294 0	0.264 5
对比 1	0.598 3	0.617 0	0.588 3	10.567 5	10.599 0	0.812 3	0.297 2	0.267 9
对比 2	0.551 6	0.568 6	0.540 8	10.036 3	10.066 0	0.779 2	0.346 7	0.313 5

表 11　在藏汉翻译测试集上的实验结果

模型	BLEU5-SBP	BLEU5	BLEU6	NIST6	NIST7	GTM	mWER	TER
主系统	0.469 4	0.485 0	0.441 3	10.393 4	10.406 3	0.779 8	0.366 4	0.326 3
对比 1	0.467 2	0.482 9	0.439 2	10.374 0	10.386 9	0.779 1	0.367 4	0.327 8
对比 2	0.445 9	0.464 8	0.420 7	10.098 4	10.110 4	0.764 4	0.394 0	0.350 7

表 12　不同切分粒度对实验结果的影响

系统	BLEU5	BLEU5-SBP
Base(BPE-BPE)	42.70	39.38
Base(BPE-Cha)	41.43	38.52
Base(WordPiece)	43.44	39.35

从表 12 的实验结果可以发现 3 种切分方法的性能基本相当,其中源语言和目标语言分别采用以亚词(BPE) 为翻译单元在 BLEU5-SBP 取得了最好的翻译结果(39.38) ,而以 WordPiece 为翻译单元的模型在 BLEU5 上表现最好. 考虑到 BPE 方法较为简单实用,因此在后续的实验采用 BPE 的切分方法来对源语言和目标语言进行切分.

然后,本次评测对比了模型参数、GPU 数量和相对位置建模对实验结果的影响(见表 13) . 具体来说,在藏汉开发集上,使用 3 个 GPU 所训练得到的模型取得了 41.06 BLEU 值的翻译质量,而采用单个 GPU 训练得到的模型取得了 40.68 BLEU 值,3 个 GPU 的模型的翻译质量比使用 1 个 GPU 的模型高 0.38 BLEU 值. 因此,后面的评测中采用 3 个 GPU 来训练模型.

通过对比相对位置模型与绝对位置模型的实验结果,可以看到相对位置模型比绝对位置模型高出 0.85 BLEU 值. 因此,后面的实验中采用相对位置模型.

本次评测得出了 Big 模型设置(实验设置章节所设置的模型参数) 优于 Base 模型(另一种较为常用的实验参数设置[9]) . 因此,在后续评测中,使用了 Big 模型设置作为基线模型.

本次评测也对比了知识蒸馏、单语数据以及不同译码策略对最终的实验结果的影响,实验结果如表 14 所示,其中基线系统是仅使用平行语料训练得到的单模型的结果, “+伪平行”是使用平行语料加上伪平行语料后的结果, “+模型平均”是指经过模型平均后的结果, “+模型集成”是经过模型集成后的译码结果, “+重排序”则是在集成译码基础上采用重评分策略后的翻译结果.

表 13　模型参数、GPU 数量和相对位置建模对实验结果的影响

模型	BLEU
Base 参数 + 1 GPU + 绝对位置	40.68
Base 参数 + 1 GPU + 相对位置	41.53
Base 参数 + 3 GPU + 绝对位置	41.06
Big 参数 + 1 GPU + 相对位置	42.49

表 14 蒙汉翻译开发集知识蒸馏、单语数据以及不同译码策略的实验结果

模型	BLEU
基线系统(base)	53.60
基线系统(big)	55.27
+知识蒸馏	56.86
+伪平行	61.78
+模型平均	62.70
+模型集成	66.70
+重排序	67.77

从表 14 的实验结果可以看出,伪平行数据使用、模型平均、集成译码和重评分策略对翻译质量的提升均有一定的帮助. 加入伪平行数据的方法对结果有显著提升,这说明了该方法的有效性.

对于低资源语料,知识蒸馏能达到增强语料的作用. 但是,双语数据上的知识蒸馏表现不稳定,如在蒙汉和维汉上有显著提升,而在藏汉上却基本持平.

4.2.2 单语数据的分析 本次评测也分析了不同的单语数据策略对实验结果的影响,并分析了不同的加噪声方法对实验结果的影响. 具体实验结果如表 15 所示.

表 15 蒙汉翻译开发集不同单语数据策略和不同加噪声方法的实验结果

模型	BLEU
基线系统	56.54
+大规模单语	57.90
+小规模单语	59.35
+小规模单语+前噪声	60.04
+小规模单语+后噪声	61.78
+小规模单语+后噪声+过采样	62.70

这里“+大规模单语”为筛选出来的大规模 300 万单语数据回翻后构成的平行语料;“+小规模单语”为对回翻后的语料进一步过滤得到的 40 万伪平行数据;“+前噪声”表示先对汉语单语数据进行加噪,然后再翻译;“+后噪声”表示先对汉语单语数据进行翻译,然后再加噪;“+过采样”表示对双语平行数据进行过采样.

从表 15 可以得出以下结论:(i) 不是所有的单语数据都对翻译质量有帮助,小规模单语取得了较好的实验结果;(ii) 单语数据回翻完成后再过滤有利于质量提升;(iii) 单语数据加噪再回翻在实际中有一定帮助.

5 总结

本文介绍了中科院自动化研究所在本次CCMT-2019 从少数民族语言到汉语的机器翻译任务上使用的主要技术和方法. 总体来说,本次评测在模型上使用了基于自注意力机制的 Transformer 的架构,并利用知识蒸馏方法提高模型效果. 在数据预处理方面,探索了多种语料过滤方法,使用反向翻译方法构建伪数据. 在译文输出过程中,采用了模型平均和集成解码的策略,并利用重评分给出最后的译文. 实验结果表明,这些方法能够有效提高翻译的质量.

受限于时间和计算资源,本次评测中还有许多方法没有尝试,也在评测过程中发现了一些不足和问题,采用的翻译模型和系统仍存在很大提升空间. 在今后的研究中期望能够学习各方先进技术,为提升我国的机器翻译水平贡献绵薄之力.

6 参考文献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [EB/OL]. [2019-04-17]. https://arxiv.org/abs/1706.03762.

[2] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subwordunits [EB/OL]. [2019-04-17]. https://arxiv.org/abs/1508.07909.

[3] 刘宇宸,闫璟辉,张家俊,等. 第 14 届机器翻译研讨会中科院自动化所技术报告 [EB/OL]. [2019-04-17]. https://max.book118.com/html/2019/0123/6152242142002003.shtm.

[4] Cho K, Gulcehre B V M C, Bahdanau D, et al. Learning phrase representationsusing rnn encoder-decoder for statistical machine translation [EB/OL]. [2019-04-17]. https://arxiv.org/abs/1406.1078.

[5] Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning [EB/OL]. [2019-04-17]. https://arxiv.org/pdf/1705.03122.pdf.

[6] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network [EB/OL]. [2019-04-17]. https://arxiv.org/abs/1503.02531.

[7] Dyer C, Chahuneau V, Smith N A. A simple, fast, and effective Reparameterization of IBM Model 2 [EB/OL]. [2019-04-17]. https://www.aclweb.org/anthology/N13-1073.

[8] Stolcke A. SRILM: an extensible language modeling toolkit [EB/OL]. [2019-04-17]. https://www.bibsonomy.org/bibtex/3c28bc012b07b969bbd7df72f7e19762.

[9] Edunov S, Ott M, Auli M, et al. Understandingback-translation at scale [EB/OL]. [2019-04-17]. https://arxiv.org/abs/1808.09381.

[10] Sennrich R, Birch A, Currey A, et al. The university of edinburgh´sneural MT systems for WMT17 [EB/OL].

[2019-04-17]. https://arxiv.org/pdf/1708.00726.pdf.

[11] Liu Yuchen, Zhou Long, Wang Yining, et al. A comparable study on model averaging, ensembling and reranking in nmt [EB/OL] [2019-04-17]. http://link.springer.com/chapter/10.1007/978-3-319-99501-4_26.

[12] Liu Lemao, Utiyama M, Finch A, et al. Agreement on target-bidirectional neural machine translation [EB/OL]. [2019-04-17]. https://www.researchgate.net/publication/305334571_Agreement_on_Target-bidirectional_Neural_Machine_Translation.

[13] Zhao Yang, Zhang Jiajun, He Zhongjun, et al. Addressing troublesome words in neural machine translation [EB/OL]. [2019-04-12]. // https://www.researchgate.net/publication/334116501_Addressing_Troublesome_Words_in_Neural_Machine_Translation.

[14] Zhou Long, Zhang Jiajun, Zong Chengqing. Synchronous bidirectional neural machine translation [J]. Transactions of the Association for Computational Linguistics, 2019, 7: 91-105.

[15] Shu R, Nakayama H. Later-stage minimum Bayes-risk decoding for neural machine translation [EB/OL]. [2019-04-17]. https://arxiv.org/pdf/1704.03169.pdf.

[16] Omar F Zaidan. Z-MERT: a fully configurable open source tool for minimum error rate training of machine translation systems [J]. The Prague Bulletin of Mathematical Linguistics, 2009, 91: 79-88.

The Study on Ethnic-to-Chinese Scare-Resource Neural Machine Translation

ZHAO Yang, ZHOU Long, WANG Qian, MA Cong, LIU Yuchen, WANG Yining,
XIANG Lu, ZHANG Jiajun, ZHOU Yu, ZONG Chengqing
(Institute of Automation, Chinese Academy of Science, Beijing 100190, China)

Abstract: The overview and the technical details adopted by Institute of Automation Chinese Academy of Science (CASIA) to participate in the 15th China Conference on Machine Translation (CCMT 2019) evaluation tasks are described in the paper. In the conference, CASIA participates in three translation tasks, including Mongolian-Chinese daily language machine translation, Tibetan-Chinese government literature machine translation, and Uyghur-Chinese news machine translation. The content of the report describes the model framework, datasets pre-processing methods and decoding strategies. Lastly, the report gives the performance of the system on the evaluation dataset under different settings and conducts a comparative analysis.

Key words: neural machine translation; low-resource machine translation; self-attention mechanism

(责任编辑:冉小晓)