文章编号: 1000-5862(2020) 01-0039-07

多种数据泛化策略融合的神经机器翻译系统

刘俊鹏,宋鼎新,张一鸣,黄德根*
(大连理工大学计算机科学与技术学院,辽宁 大连 116024)

摘要: 在 Transformer 模型的基础上,该文从数据泛化、多样化解码策略和后处理方法 3 个方面进行改进. 多种数据泛化策略融合方法对不同种类的稀疏词语进行识别、泛化和翻译,减少错译现象. 利用检查点平均和模型集成等多样化解码策略进一步提升翻译效果. 在 CCMT 2019 中英新闻领域翻译任务上的实验结果显示,改进后的方法在基线系统上的 BLEU-SBP 值提升了约 1.85%.

关键词: 神经机器翻译; 自注意力机制; 数据泛化; 中英翻译
中图分类号: TP 391 **文献标志码**: A **DOI**: 10.16357/j.cnki.issn1000-5862.2020.01.07

0 引言

基于序列到序列的神经机器翻译[1] 经过不断地发展和完善显示出其强大的翻译性能. 随着长短时记忆单元[2] (long short-term memory) 、门循环单元[3] (gated recurrent unit) 和注意力机制[4] (attention mechanism) 的融入,神经机器翻译的性能获得显著提升,逐渐超越了传统的统计机器翻译方法的性能[5-6] ,成为机器翻译领域的主流方法. 近年来,基于完全自注意力(self-attention) 的神经机器翻译模型 Transformer[7] ,在目前所有机器翻译模型中取得了最好的翻译效果. 由于 Transformer 模型不依赖于循环神经网络和卷积神经网络,具有训练时间更短且并行能力更强的特点,因此在本文的实验中将 Transformer 模型作为基线系统.

本文面向中英新闻领域翻译任务,从数据泛化处理、多样化解码策略及后处理 3 个方面介绍相关方法和技术. 在语料预处理方面,首先对中英双语平行语料进行清洗处理; 然后对中英文语料进行子词处理,以应对集外词问题; 最后,对于语料中的部分实体及特殊表达式进行泛化处理. 在解码策略方面,对束搜索宽度(beam size) 和长度惩罚因子 α 等参数进行调优; 综合运用检查点平均(checkpoint averaging) 、模型集成(model ensemble) 等策略进行多样化解码,并将解码方法使用顺序和数量对实验结果的影响进行研究. 在后处理方面,对泛化部分采取不同的翻译策略进行翻译,并恢复得到最终的翻译结果.

在基线系统基础上,主要从以下 3 个方面进行了改进: (i) 多种数据泛化方法. 通过多样化的方法和规则识别匹配中英双语语料中翻译难度较大的人名、时间表达式、数字表达、网址及特殊表达式,并进行泛化处理. 在泛化过程中,对同一句子中的同类泛化成分进行编号处理以示区分,因此无需对翻译结果中同类泛化结果再进行匹配处理,从而降低了恢复难度. (ii) 多样化解码策略. 通过调整 beam size 和长度惩罚因子 α 对模型参数进行调优,利用不同的方式结合检查点平均和模型集成等技术进行多样化解码. (iii) 后处理技术. 对翻译结果中的泛化部分采用不同的方式进行翻译,并用翻译结果替换其对应的泛化标志符,得到最终的译文.

1 Transformer 模型

1.1 基本模型结构

Transformer 模型的编码器由 N 个同构的网络层堆叠而成. 每一个网络层包含 2 个子网络层: 第 1 层是自注意力机制,第 2 层是一个全连通的前馈神经网络. 在每个子层后,使用残差网络[8] 和层级规

收稿日期: 2019-09-08
基金项目: 国家自然科学基金(61672127) 资助项目.
通信作者: 黄德根(1965-) ,男,福建邵武人,教授,博士生导师,主要从事自然语言处理、神经机器翻译方面的研究. E-mail: huangdg@ dlut. edu. cn

范化[9]连接,以避免由于层数过多而导致模型难以收敛的问题. 编码器中还加入了词的位置编码,使模型能更好地学习序列信息. 通过对 N 个这样的网络层堆叠可以对信息进一步抽象整合,由于残差网络的引入,同构网络中的每个子网络输出以及词向量和位置编码均需要保持同样的维度.

解码器同样包含 N 个堆叠的同构网络层,每个网络层包含 3 个子网络层: 第 1 层是与编码器相似的自注意力机制,不同之处在于由于解码器在解码时只能看到已生成词的信息,因此模型使用掩码技术以屏蔽未生成词的信息; 第 2 层是多头注意力网络,该网络将源语言句子的隐层状态同目标语言的隐层状态进行建模生成源语言句子的上下文向量; 第 3 层是一个全连通的前馈神经网络. 与编码器类似,解码器的每个子层后也使用了残差网络和层级规范化连接.

1.2 注意力机制

Transformer 的注意力机制由缩放点积注意力(scaled dot-product attention) 和多头注意力(multi-head attention) 2 部分组成.

1.2.1 *缩放点积注意力* 该注意力网络计算一个隐层状态 $\boldsymbol{Q}$ 和某个隐层状态 $\boldsymbol{K}$ 的相似度,通过 softmax 归一化生成权重,使用该权重计算隐层状态 $\boldsymbol{V}$ 的加权值并输出. 在使用 softmax 进行归一化时,引入缩放因子 $\sqrt{d_k}$ 以避免在反向传播时产生一个特别大的梯度值而导致训练过程不稳定,计算公式为

$$A_{ttention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = s_{oftmax}(\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}/\sqrt{d_k})\,\boldsymbol{V}. \quad (1)$$

1.2.2 *多头注意力* 与传统的注意力机制只使用一个注意力网络不同,多头注意力网络将多个注意力网络进行拼接. 给定一组 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$,首先使用不同的线性映射将 $\boldsymbol{Q}$、$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 映射到不同的子空间,然后使用不同的注意力网络计算得到不同空间的上下文向量,并将这些上下文向量拼接后得到最后的输出,计算公式为

$$M_{ultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = C_{oncat}(\boldsymbol{h}_{ead_1},\boldsymbol{h}_{ead_2},\cdots,\boldsymbol{h}_{ead_h})\,\boldsymbol{W}^{O}, \quad (2)$$

$$\boldsymbol{h}_{ead_i} = A_{ttention}(\boldsymbol{Q}\boldsymbol{W}_i^{\boldsymbol{Q}},\boldsymbol{K}\boldsymbol{W}_i^{\boldsymbol{K}},\boldsymbol{V}\boldsymbol{W}_i^{\boldsymbol{V}}), \quad (3)$$

其中 $\boldsymbol{W}_i^{\boldsymbol{Q}}$、$\boldsymbol{W}_i^{\boldsymbol{K}}$、$\boldsymbol{W}_i^{\boldsymbol{V}}$、$\boldsymbol{W}^{O}$ 均为参数矩阵.

1.3 *位置编码* 由于基于完全注意力网络的编码器和解码器都没有考虑位置信息,而位置信息对于语言的理解和生成十分重要,因此 Transformer 模型在最底层编码器和解码器的输入向量中加入位置编码. 位置编码可以采用固定位置编码、相对位置编码或者学习到的位置编码. 其中,固定的位置编码基于三角函数,具体公式为

$$P_E(p_{os},2i) = \sin(p_{os}/(10\,000^{2i/d_{model}})), \quad (4)$$

$$P_E(p_{os},2i+1) = \cos(p_{os}/(10\,000^{2i/d_{model}})), \quad (5)$$

其中 p_{os} 为词语的位置序号,i 为位置编码的维度,d_{model} 为位置编码的长度. $P_E(p_{os},2i)$ 定义了位置序号为 p_{os} 的位置编码的第 $2i$ 维的值. 同样地,$P_E(p_{os},2i+1)$ 则定义了位置编码的第 $2i+1$ 维的值. 这样,对于 2 个相距固定间隔 k 的词语,其位置编码 $P_{E_{pos+k}}$ 可以经由三角函数变换由 $P_{E_{pos}}$ 表示.

2 语料处理

2.1 语料预处理

训练语料使用由 CCMT 2019 提供的中英新闻领域双语平行语料. 由于训练语料的质量对于最终翻译模型的结果有较大影响,因此在训练前对训练语料进行了预处理,主要处理内容包括: (i) 对语料中含有乱码的句子进行过滤; (ii) 对语料中的转义字符进行转换处理; (iii) 将语料中的全角字符转成半角字符; (iv) 对训练语料源语言/目标语言分词结果进行长度比过滤; (v) 使用 GIZA++ 工具(http://code.google.com/p/giza-pp/downloads/detail?name=giza-pp-v1.0.7.tar.gz) 对训练语料进行词对齐,删除双语句子中词对齐比例过低的句子; (vi) 过滤语料中重复的句子.

2.2 分词与 BPE 子词处理

中英文分词均采用在东北大学 NiuTrans[10] 中提供的分词工具. 为了减少词汇表大小,同时应对集外词问题,使用 R. Sennrich 等[11] 提出的 BPE 算法(https://github.com/resennrich/subword-nmt) 将词语切分成粒度更小的子词,并用标志符“@@”连接 2 个连续的子词,以便在翻译完成后进行恢复.

2.3 语料泛化处理

“识别-匹配-替换”的方式对训练前的语料进行泛化处理. 在新闻领域的语料中常常包含着大量的命名实体(人名、地名和组织机构名),这些命名实体出现的次数较多,但重复率不高,特别是人名. 在本次评测中只处理了命名实体中的人名部分. 此外,对时间表达式、数字、网址以及特殊表达等也进行了泛化处理,具体方式如下: 首先,对出现在训练语料

中的人名、时间表达式、数字、网址以及特殊表达等进行识别匹配,然后将上述识别匹配的双语对齐对分别替换为“$name”、“$date”/“$time”、“$number”、“$web”和“$special”标志符. 由于在同一句子中通常存在着多个同类泛化成分,在替换时若不对其进行区分,则在翻译完成后进行恢复时需要再次对齐和匹配,增加了恢复的复杂度. 为了降低恢复难度,在替换时对同一句子中的同一类型的标志符进行编号处理,即“$name_$i$”、“$date_i”/“$time_$i$”、“$number_i”、“$web_$i$”和“$special_i”($i=0,1,\cdots,n$),以便于区分和建立对应关系,示例如表 1 所示.

表 1　同类标志符替换示例

示例	原文	译文
1	$name_0 总统称赞 $name_2 为他战胜民主党人 $name_1 设计师.	president $name_0 paid tribute to $name_2 as the architect of his victory over democrat $name_1.
2	$date_0 初期实际利率是负数,但在 $date_1 初却急剧上升.	real interest rates were negative during the early part of the $date_0, but they rose sharply in the early $date_1.
3	不幸的是,轻轨起点不在市中心,但乘坐磁悬浮去机场全程仅耗时 $number_0 min,时速可高达每小时 $number_1 km.	unfortunately the magnetic levitation railway doesn't start from the city center but the journey time is a mere $number_0 minutes, with speeds reaching up to $number_1 km·h^{-1}.

2.3.1　人名处理　由于词表规模的限制,神经网络翻译模型往往无法较好地处理人名的翻译问题,容易出现漏翻译或错翻译的现象,而人名的翻译质量往往对最终译文的质量有着极大的影响. 在以往的方法中,人名常常和地名、组织机构名等作为命名实体被一同识别. 这些方法[12-43]通过设计大量特征和迭代训练来识别双语语料中的命名实体. 由于在本次评测任务中仅对命名实体中的人名部分进行识别,因此对人名进行了更具针对性的处理. 利用中英人名互译词典和音译特征相结合的方法对于中英双语语料中的人名进行识别和匹配处理,如语法 1 描述所示.

语法 1　中英平行语料中双语人名的识别和匹配

Input: S: source sentence; T: target sentence; D: name dictionary; P: English pronunciation rules
Output: O: name pairs

```
Create NC saves the Chinese names extracted from S
Create NE saves the English names extracted from T
for each English name en in NE do
    if D(en) in NC then
        add en→D(en) to O
    end if
    else
        for each Chinese name cn in NC do
            if match(P(en), lazy_pinyin(cn)) is TRUE then
                add en→cn to O
            end if
        end for
    end else
end for
```

首先,使用本实验室内部开发的中文人名识别工具和 Stanford Corenlp(http://nlp.stanford.edu/software/stanford-english-corenlp-2018-10-05-models.jar)开源工具分别对中英文语料中的人名进行识别; 其次,利用中英人名互译词典、汉语拼音及英文发音规律等方法对分别识别出来的单语人名进行匹配处理; 对于匹配成功的人名对,按照句子中人名出现的顺序用带编号的人名标志符“$name_$i$”分别替换掉中英文中的人名对; 对于无法匹配的人名,则保持其在语料中的原有存在形式,不作处理.

在对识别出来的中英文人名进行匹配时,先利用 pypinyin 库(https://pypi.org/project/pypinyin/)中的 lazy_pinyin 函数将中文人名由汉字转换为拼音后再进行后续处理,从而避免由于同音异体字造成匹配不成功的现象. 为降低匹配难度,首先利用汉语拼音完成中国人名的筛选和匹配,而后再处理剩余的外国人名. 对于外国人名,首先利用中英文人名翻译词典进行匹配. 对于人名词典没有匹配成功的情况,再利用英文字母(组合)发音规律和中文汉语拼音进行模糊匹配,选取英文字母(组合)作为匹配特征,表 2 列举出部分英文字母(组合)发音及汉语拼音对照示例. match 函数用于判断中英文发音是否具有交集. 由于在同一句子中人名发音重复概率较低,因此虽然该方法采用模糊匹配,但仍具有较高的准确率.

表 2 部分英文字母(组合)发音与汉语拼音对照示例

英文字母(组合)	中文拼音	示例
a	a、y 等	Anthonio(安东尼奥)、Aristotle(亚里士多德)
j	j、y 等	Jack(杰克)、John(约翰)
s	s、x 等	Selby(塞尔比)、Sichel(西奇尔)
p	p、b 等	Pompeo(蓬佩奥)、Popkin(波普金)
ph	f	Phillip(菲利普)

2.3.2 时间表达式 由于不同语料库中时间表达式的中文表达形式不统一,往往存在着许多含义相同但书写方式不同的时间表达,如“二零零二年”、“二〇〇二年”、“2002 年”、“二 00 二年”等. 这些时间表达式是由中文数字或阿拉伯数字组合而成,数量繁多且重复率低,若不作替换处理则会占用较大一部分词汇表空间. 此外,由于时间表达式的翻译形式较为固定,因此可以按照一定规则将其由中文翻译成英文. 通过对语料进行统计分析,将时间表达式归为 2 类,如表 3 所示. 在识别匹配时,提供多种英文时间表达式的识别匹配规则,以尽可能地获得更多的匹配结果.

表 3 时间表达式匹配示例

时间表达式类别	中文时间表达式	英文时间表达式
$ date	2018 年 11 月 8 日	November 8(th),2018; Nov. 8(th),2018; 8 November,2018 等
	20 世纪 80 年代	1980s
	1980 年代	1980s
$ time	2 小时 10 分 06 秒	2:10:06
	2 时(点)15 分	2:15; a quarter past two 等

2.3.3 数字表达 与时间表达式相似,数字表达按照“百分数”、“分数”、“大写数字”、“阿拉伯数字”分为 4 类,在识别匹配时,采用多种匹配规则对英文中的数字表达进行匹配. 如中文数字表达“百分之五”、“5%”匹配英文中的“5%”、“five percent”、“5 percent”等表达形式;“五分之一”匹配英文中的“one fifth”、“one in five”等表达形式;“3 000”匹配英文中的“3 000”、“three thousand”等表达形式;“三十六点八五”匹配英文中的“36.85”等表达形式. 在处理数字表达时,还需要考虑到英文中的惯用表达形式,例如“二分之一”通常译为“a half”,而不是“a twice”;“四分之三”通常译为“three quarters”,而不是“three forths”;“十亿”通常译为“one billion”等. 对于原文中出现的上述惯用表达形式,通过总结归纳,按照固定用法翻译即可.

2.3.4 网址及特殊表达 在语料中还存在着网址(如“www.thepaper.cn”等)和部分特殊表达(如“iPhone 6s”、“F35B”等),由于此部分表达或者长度过长,或者由英文字母、数字、标点符号混合组成,若不进行处理,翻译难度极大,容易出现错翻译的情况. 因此对于这种问题也进行泛化处理,将其分别用“$ web”和“$ special”标志符进行替换.

3 解码策略

3.1 检查点平均

检查点平均[14]是指将同一模型在不同的训练时刻保存的参数进行平均,从而得到鲁棒性更强的模型参数. 在训练过程中对表现最好的 N 个模型进行参数平均,并为每个模型的参数赋予相同的权重.

3.2 模型集成

模型集成是利用多个神经机器翻译系统协同进行解码的方法,在神经机器翻译领域中得到了广泛使用[15-16]. 设有 K 个已经训练好的神经机器翻译系统,这 K 个神经机器翻译系统可以是同构或者异构的系统,可以是在相同的训练数据或者不同的训练数据上得到的模型. 一般而言,结构和初始化方式均不同的模型通常更具有差异性,往往能够带来更大提升[15]. 实验中,在同一框架下利用不同随机初始化参数训练出 N 个相同的模型,并分别利用同一模型的不同检查点、N 个模型中的不同检查点以及 N 个平均模型进行了集成解码,以消除随机初始化对翻译结果带来的影响.

4 后处理

4.1 泛化部分翻译和恢复

在训练和测试过程中,由于对语料进行了泛化处理,经过解码后的译文中还包含泛化符号,因此需要对泛化部分进行翻译和恢复后才能得到最终的翻译结果. 在泛化过程中,由于同一句子中的同类泛化成分已经进行了序号标注,因此在对泛化部分进行翻译和恢复时,无需考虑同一句子中同类泛化成分的对齐匹配问题. 在实验中,对于人名、时间表达式、

数字表达、网址及特殊表达分别采用不同的翻译恢复方式.

对于人名,采用人名词典和拼音相结合的方式进行翻译和恢复,先查询中英人名词典查找英文翻译结果,若词典中包含查询人名的翻译结果,则输出对应的人名翻译; 若词典中无此查询结果,则直接将中文人名的拼音作为英文翻译结果; 对于时间表达式和数字表达,首先统计出常见的英文表达方式,然后根据统计结果编写固定的翻译规则(见表 4) ,最后利用规则生成翻译结果并进行恢复; 对于网址和特殊表达,由于中英文表达方式相同,因此无需翻译,只需将译文中相应的替换标志符恢复即可.

表 4　时间表达式及数字的翻译示例

类别	原文	翻译示例
时间表达式	2018 年 10 月 12 日	October 12,2018
	二十世纪七十年代	1970s
	上午 8 时	8 am
	3 分 56 秒	3′56″
数字表达	五分之三	three fifths
	百分之十	10%
	两千万	20 million
	六千三百七十八点二	6 378.2

4.2　大小写转换方法

由于翻译模型输出的英文翻译结果均为小写,而评测系统采用大小写敏感的评测工具进行评分,因此需要将翻译结果还原为大小写混合文本. 处理方法是: 首先使用 Moses 中提供的 recase 脚本(https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/recase.perl) 对模型中的大小写进行转换,再利用程序将句首字母强制大写来对 recase 脚本的结果进行补充修正,得到最终的译文.

5　实验结果

5.1　实验参数

基线系统使用清华大学开源框架 THUMT(https://github.com/THUNLP-MT/THUMT) 中提供的 Transformer 模型,具体实验参数如下: 编码器和解码器均为 6 层,多头注意力机制采用 8 个头,词向量维度和隐藏层状态维度均设置为 512,前馈神经网络层中的隐藏层状态维度设置为 2 048. 在训练阶段,每个 batch 包含 6 250 个中文或英文 token,模型训练 20 万个 step,并且每 2 000 个 step 保存 1 个 checkpoint,并在训练过程中保存最优的 10 个 checkpoint 以用于测试阶段的参数平均. 使用极大似然估计作为训练的损失函数,并使用 Adam 优化算法[17] (其中 $\beta_1=0.9$,$\beta_2=0.98$,$\varepsilon=10^{-9}$) ,初始学习率为 1.0,warm up 为 4 000. 训练集的中英语料在分词之后均采用 BPE 算法进行子词处理,中英文词表大小均为32 000,2 者不共享词向量. 在测试阶段,使用 beam search 算法和 length normalization[18] 以选择最佳的翻译结果. 实验中训练了 3 个独立的 Transformer 模型,并采用了检查点平均和集成解码等技术. 训练和测试过程均基于单个 Tesla K40c GPU. 在实验中采用大小写敏感的 BLEU-SBP 值[19] 作为评价指标,在评测之前,首先使用脚本将翻译文本处理为 CCMT 2019 规定的 xml 文件,而后使用 CCMT 2019 提供 BLEU-SBP 脚本作为评测工具.

5.2　实验结果及分析

在训练阶段,使用 CCMT 2019 中英新闻领域翻译任务提供的语料进行训练,系统在验证集上的实验结果如表 5 所示.

表 5　CCMT2019 中英新闻领域翻译任务验证集 BLEU-SBP 值

系统		beam size	BLEU-SBP
基线系统	Transformer 模型	4	20.96
①	单模型中 5 个检查点平均解码	4	21.42
②	3 个平均模型集成解码	4	22.44
③	3 个独立训练模型集成解码	4	22.63
④	3 个独立训练模型中 7 个检查点集成解码	4	22.81

5.2.1　基本实验结果分析　如表 5 所示,基线系统是使用平行语料在单模型上的测试结果,方法①是在单模型训练过程中选择结果最好的 5 个检查点,利用检查点平均技术对其参数进行加权平均,再利用平均后的模型得到的翻译结果; 方法②是独立训练的 3 个相同模型按照方法①得到平均模型后,再进行集成解码得到的实验结果; 方法③是利用独立训练的 3 个相同模型各自最好的检查点进行集成解码后得到的翻译结果; 方法④是在 3 个独立训练的相同模型中选取最好的 7 个检查点进行集成解码后得到的翻译结果.

根据上述实验结果可以得出以下结论: (i) 检查点平均和模型集成解码技术均能有效地提升翻译效

果,应用这 2 种技术后得到的翻译结果的 BLEU-SBP 值分别比基线系统提升了 0.46% 和 1.48%,且集成解码提升效果更为明显. (ii) 对比方法②和方法③的结果可知,虽然模型直接进行集成解码和模型平均后再集成解码 2 种方式都能提升翻译结果的 BLEU-SBP 值,但是前者比后者的结果高 0.19%. 分析这 2 种方法得到的译文后可以发现,前者得到的译文长度明显长于后者,这说明后者的结果具有趋向短句翻译的倾向,因此出现了一些漏翻译现象,从而造成了 BLEU-SBP 值的降低. (iii) 方法④比方法③的结果提升了 0.18%,这说明增加联合解码的模型个数还可以进一步提升实验结果. 由于单个 GPU 能力的限制,实验中最大只测试了 7 个检查点的集成解码,没有对更多模型的情况下进行进一步实验.

5.2.2 beam size 和长度惩罚分析 在实验中,在方法③的基础上研究了不同的 beam size 和长度惩罚因子 α 对实验结果的影响,结果如表 6 所示.

表 6 束搜索宽度和长度惩罚因子 α 对 BLEU-SBP 值的影响

beam size	α							
	0.6	0.8	1.0	1.1	1.2	1.3	1.4	1.5
4	22.63	22.81	23.01	23.07	22.99	22.62	22.03	21.09
10	22.56	22.85	23.22	23.24	23.18	22.90	22.34	21.38

由表 6 的实验结果可以得出以下结论: (i) 在长度惩罚因子相同的情况下,当 beam size 由 4 增加到 10 时,翻译结果的 BLEU-SBP 值也获得了提升,但由于条件限制,没有测试在更大的 beam size 下的实验结果,因此还有待进一步研究; (ii) 在 beam size 相同的情况下,随着长度惩罚因子的增大,BLEU-SBP 值的变化是先提升后下降,这说明过大的长度惩罚因子可能会导致束搜索无法选择出正确的结果,从而导致 BLEU-SBP 值降低. 根据表 6 的实验结果,在系统中将长度惩罚因子设置为 1.1.

6 总结

本文介绍了大连理工大学自然语言处理 & 机器翻译实验室在 CCMT2019 中英新闻领域翻译任务上使用的主要方法和技术. 以基于自注意力的 Transformer 作为基线系统,从语料预处理、解码策略和后处理等方面进行了改进. 在对语料预处理时,对语料中的人名、时间表达式、数字、网址及特殊表达进行识别替换. 在译码时使用了模型平均、集成解码的策略. 在后处理阶段对泛化部分进行了自动翻译和恢复. 实验结果显示,这些方法能够提升翻译质量.

由于时间仓促和实验环境的限制,本次评测仅仅在 Transformer 模型上利用常用处理方法进行了实验,还有很多方法和技术没有尝试. 同时,在实验和后期处理当中还发现了一些问题和不足,这些有待于今后进一步研究.

7 参考文献

[1] Sutskever I, Vinyals O, Le Quoc V. Sequence to sequence learning with neural networks [EB/OL]. [2019-06-17]. https://arxiv.org/abs/1409.3215.

[2] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.

[3] Cho K, Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [EB/OL]. [2019-06-17]. https://arxiv.org/abs/1406.1078.

[4] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [EB/OL]. [2019-06-17]. https://arxiv.org/abs/1409.0473v2.

[5] Koehn P, Och F J, Marcu D. Statistical phrase-based translation [EB/OL]. [2019-06-17]. http://citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.69.4271.

[6] Chiang D. A hierarchical phrase-based model for statistical machine translation [EB/OL]. [2019-06-17]. https://dl.acm.org/citation.cfm? doid=1219840.1219873.

[7] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [EB/OL]. [2019-06-17]. https://arxiv.org/abs/1706.03762.

[8] Ba J L, Kiros J R, Hinton G E. Layer normalization [EB/OL]. [2019-06-17]. http://arxiv.org/abs/1607.06450.

[9] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Identity mappings in deep residual networks [C]. Amsterdam: Springer, 2016: 630-645.

[10] Xiao Tong, Zhu Jingbo, Zhang Hao, et al. NiuTrans: an open source toolkit for phrase-based and syntax-based ma-

chine translation [C]. Jeju: Association for Computational Linguistics,2012: 19-24.

[11] Senntich R, Haddow B, Brich A. Neural machine translation of rare words with subword units [C]. Berlin: Association for Computational Linguistics,2016: 1715-1725.

[12] Huang Fei, Vogel S, Waibel A. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization [C]. Sapporo: Association for Computational Linguistics,2003: 9-16.

[13] Feng Donghui, Lü Yajuan, Zhou Ming. A new approach for English-Chinese named entity alignment [C]. Barcelona: Association for Computational Linguistics,2004: 372-379.

[14] Sennrich R, Haddow B, Birch A. Edinburgh neural machine translation systems for WMT 16 [C]. Berlin: Association for Computational Linguistics,2016: 371-376.

[15] Sennrich R, Birch A, Currey A, et al. The University of Edinburgh's Neural MT Systems for WMT17 [C]. Copenhagen: Association for Computational Linguistics, 2017: 389-399.

[16] Wang Yuguang, Cheng Shanbo, Jiang Liyang, et al. Sogou neural machine translation systems for wmt17 [C]. Copenhagen: Association for Computational Linguistics, 2017: 410-415.

[17] Kingma D P, Ba J. Adam: a method for stochastic optimization [EB/OL]. [2019-06-17]. https://arxiv.org/abs/1412.6980.

[18] Wu Yonghui, Schuster M, CHEN Zhifeng, et al. Google's neural machine translation system: bridging the gap between human and machine translation [EB/OL]. [2016-10-08] [2019-08-29]. http://arxiv.org/pdf/1609.08144.pdf.

[19] Chiang D, Deneefe S, Chan Yee Seng, et al. Decomposability of translation metrics for improved evaluation and efficient algorithms [C]. Honolulu: Association for Computational Linguistics,2008: 610-619.

The Neural Machine Translation System of Multiple Data Generalization Fusion

LIU Junpeng, SONG Dingxin, ZHANG Yiming, HUANG Degen*

(College of Computer Science and Technology, Dalian University of Technology, Dalian Liaoning 116024, China)

Abstract: The improvements on the Transformer baseline system are described from three aspects, including data generalization, multiple decoding strategies and post-processing. Multiple data generalization fusion method is used to recognize, generalize and translate different types of rare words, which reduces mistranslation in the neural machine translation. Multiple decoding strategies such as checkpoint averaging and model ensemble can further boost the translation performance. Experimental results on CCMT 2019 Chinese-English news translation task show that the proposed methods significantly improve translation performance by about 1.85% BLEU-SBP points than baseline system.

Key words: neural machine translation; self-attention; data generalization; Chinese-to-English translation

(责任编辑: 冉小晓)