# 倾向近邻关联的神经机器翻译

王 坤 段湘煜
(苏州大学计算机科学与技术学院 江苏 苏州 215006)

**摘 要** 现有神经机器翻译模型在对序列建模时,仅考虑目标端对应源端的关联性,未对源端关联性及目标端关联性建模。文中分别对源端以及目标端关联性建模,并设计合理的损失函数,使得源端隐藏层与其近邻 $K$ 个单词隐藏层更相关,目标端隐藏层与其历史 $M$ 个单词隐藏层更相关。在大规模中英数据集上的实验结果表明,相比于神经机器翻译中仅考虑目标端对应源端的关联性,所提方法可以构建更好的近邻关联表示,提升机器翻译系统的译文质量。

**关键词** 机器翻译,近邻关联,注意力机制
**中图法分类号** TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.05.030

## Neural Machine Translation Inclined to Close Neighbor Association

WANG Kun DUAN Xiang-yu
(School of Computer Science & Technology, Soochow University, Suzhou, Jiangsu 215006, China)

**Abstract** The existing neural machine translation model only considers the relevance of the target end corresponding to the source end when modeling the sequences, and does not model the source end association and the target end association. In this paper, the source and target associations were modeled separately, and a reasonable loss function was designed. The source-hidden layer is more related to its neighboring $K$ word-hidden layers. The target-side hidden layer is more related to its historical $M$ word-hidden layers. The experimental results on the large-scale Chinese-English dataset show that compared with the neural machine translation which only considers the relevance of the target end to the source, the proposed method can construct a better neighbor correlation representation and improve the translation quality of the machine translation system.

**Keywords** Machine translation, Close neighbor association, Attention machanism

## 1 引言

机器翻译(MT)的任务是通过计算机将一种语言转化成另一种语言。随着国内外交流的增加,普适性的机器翻译系统紧缺,如何获得更好的翻译性能成为了众多研究者的研究目标。

随着注意力(Attention)机制的引入以及图形处理器(GPU)的发展,神经机器翻译(NMT)[1-3]快速发展并且在众多任务上优于传统的统计机器翻译方法(SMT),使得 NMT 成为了机器翻译领域的新范式。

目前,基于注意力的神经机器翻译模型在对源端序列和目标端序列建模时,只考虑目标端对应源端的关联性信息,未考虑源端关联性以及目标端关联性信息,因此将关联性信息融入到神经机器翻译模型中成为了新思路。

本文在注意力机制的基础上,通过增加源端和目标端的近邻关联来提升机器翻译效果。具体而言:在源端,优化目标使得源端隐藏层与其近邻 $K$ 个单词隐藏层相关;在目标端,优化目标使得目标端隐藏层与其历史 $M$ 个单词隐藏层相关。实验结果表明,相比于基准系统,本文所提方法能显著改善神经机器翻译的性能。

本文第2节介绍相关工作;第3节介绍神经机器翻译基准系统;第4节介绍提出的倾向近邻关联的神经机器翻译方法;第5节介绍实验设置以及实验结果;最后总结全文并展望未来。

## 2 相关工作

近年来,机器翻译领域的研究者对关联性信息进行了诸多研究工作。

Luong 等[4]提出局部注意力机制,使得目标端在生成单词时,更关注于源端某一部分信息;Liu 等[5]、Mi 等[6]和 Chen 等[7]通过使用目标端对应源端的对齐指导,使得神经机器翻译系统获得了更好的词对齐信息,以此指导 NMT 获得更好的译文;Vaswani 等[8]完全使用注意力对源端和目标端建模,使得源端和目标端可以捕获长距离关联性信息;Gehring

到稿日期:2018-04-18 返修日期:2018-09-01 本文受国家自然科学基金(61673289),国家重点研发计划"政府间国际科技创新合作"重点专项(2016YFE0132100)资助。

王 坤(1994-),男,硕士生,主要研究方向为自然语言处理、机器翻译,E-mail:liesun1994@gmail.com;段湘煜(1976-),男,副教授,主要研究方向为自然语言处理、机器翻译,E-mail:xiangyuduan@suda.edu.cn(通信作者)。

等[9]提出完全使用卷积神经网络(CNN),使得源端和目标端可以捕获局部关联性信息;Mikolov 等[10]提出使用近邻信息来获得更好的词向量表示。

区别于文献[4-7],本文提出的方法为在源端和目标端分别添加近邻关联;区别于文献[8-9],本文提出在循环神经网络的基础上增加源端目标端注意力机制,以获得更好的近邻关联表示,辅助神经机器翻译系统;区别于文献[10],本文所提方法能够获得更好的隐藏层近邻关联表示,并以此来指导 NMT。

## 3　神经机器翻译基准模型

目前神经机器翻译常用的结构为编码器解码器(Encoder-Decoder)结构,具体如图 1 所示。神经机器翻译通常使用循环神经网络(RNN)来获得语言中的长期依赖关系,在实际应用中通常使用长短时记忆单元(LSTM)[11]或者门循环单元(GRU)[12]。

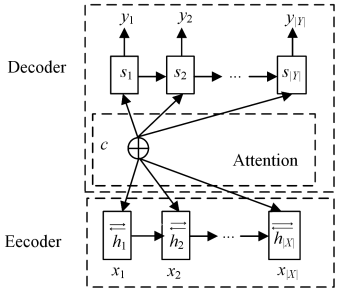



图 1　基准系统结构图

Fig. 1　Baseline structure

### 3.1　编码器

编码器通常使用双向循环神经网络对源端序列建模:对于给定的源端词嵌入(Word Embedding)序列 $x=x_1,x_2,\cdots,x_{|X|}$,通过正反向编码器将其编码成源端向量表示 $h=h_1,h_2,\cdots,h_{|X|}$。

$$\overrightarrow{h_i}=f_1(\overrightarrow{h_{i-1}},x_i) \tag{1}$$

$$\overleftarrow{h_i}=f_2(\overleftarrow{h_{i-1}},x_i) \tag{2}$$

$$h_i=[\overrightarrow{h_i},\overleftarrow{h_i}] \tag{3}$$

其中,$|X|$ 为源端句子长度;实际应用中通常使用 LSTM 或者 GRU 作为函数 $f_1$ 和 $f_2$;[,]为向量拼接。

### 3.2　解码器

解码器使用循环神经网络并通过多层感知器预测目标端单词 $y_t$。单词的预测由解码端 RNN 计算的隐藏层 $s_t$、前一时刻的预测单词 $y_{t-1}$ 和上下文向量 $c_t$ 计算所得,计算方式如下:

$$p(y_t\mid y_{<t};x)=g(y_{t-1},s_t,c_t) \tag{4}$$

$$s_t=f_3(s_{t-1},y_{t-1},c_t) \tag{5}$$

其中,$f_3$ 通常使用 LSTM 或者 GRU,$g$ 为多层感知器。

在注意力模型中,上下文向量 $c_t$ 由源端隐藏层($h_1,h_2,\cdots,h_{|X|}$)的加权和计算所得:

$$e_{tn}=\vartheta^T\tanh(Ws_{t-1}+Uh_n) \tag{6}$$

$$\alpha_{tn}=\frac{\exp(e_{tn})}{\sum_{i=1}^{|X|}\exp(e_{ti})} \tag{7}$$

$$c_t=\sum_{n=1}^{|X|}\alpha_{tn}h_n \tag{8}$$

其中,$\alpha_{tn}$ 是目标端 $t$ 时刻对应 $h_n$ 的注意力权重,权重越大,源端 $h_n$ 提供给上下文向量 $c_t$ 的信息越丰富;$\vartheta\in R^{d_{mod}}$,$W\in R^{d_{mod}*d_{thid}}$,$U\in R^{d_{mod}*d_{shid}}$ 是可训练的参数矩阵,$d_{mod}$ 为模型维度,$d_{thid}$ 和 $d_{shid}$ 分别为目标端和源端的隐藏层维度。

### 3.3　损失函数

神经机器翻译模型的损失函数定义如下:

$$loss_{word}=\sum_{t=1}^{|Y|}-\log(p(y_t\mid y_{<t};x)) \tag{9}$$

其中,$|Y|$ 为译文长度,$p(y_t\mid y_{<t};x)$ 为 $t$ 时刻目标端单词对应的概率。在神经网络训练过程中,通过最小化损失来优化模型参数。

## 4　倾向近邻关联的神经机器翻译

本节在基准系统的研究基础上,介绍本文提出的倾向近邻关联的神经机器翻译模型,所提出的结构分别对源端以及目标端近邻关联建模。通过在向量之间建模,注意力机制(Attention)可以衡量向量与向量之间的关联性信息。基于此,本文在基准系统的基础上提出 3 个模型:倾向源端近邻关联模型、倾向目标端近邻关联模型以及混合近邻关联模型。

### 4.1　倾向近邻关联整体框架

对于本文所提方法,给定一个句对,整体损失定义如下:

$$loss=loss_{word}+(\Delta_{src}+\Delta_{trg}) \tag{10}$$

其中,$loss_{word}$ 为式(9)定义的交叉熵损失,$\Delta_{src}$ 为倾向源端近邻关联损失,$\Delta_{trg}$ 为倾向目标端近邻关联损失。通过优化所增加的倾向源端近邻关联和倾向目标端近邻关联模型的损失函数,使得 NMT 在获得较好的译文表示的同时也可以获得更好的源端近邻关联以及目标端近邻关联表示。

### 4.2　倾向源端近邻关联的神经机器翻译

鉴于基准系统未对源端关联性建模,本文提出倾向源端近邻关联的神经机器翻译模型。具体实现为:对应源端第 $i$ 个单词,我们选取其左右 $K$ 个窗口,使得第 $i$ 个源端单词隐藏层与其左右 $K$ 个近邻单词隐藏层更关联,如图 2 中第 1 部分虚线框所示,$h_i$ 与近邻 $K$ 时刻的关联性如黑色加粗实线所示。

倾向源端近邻关联模型的具体实现分为两个部分:1)计算每个源端单词隐藏层与所有源端单词隐藏层的关联性权重;2)设计合理的损失函数使得每个源端单词与其近邻 $K$ 个单词更相关。

#### 4.2.1　关联性计算

$\alpha_{im}^{src}$ 为源端第 $i$ 个单词隐藏层与第 $m$ 个单词隐藏层的关联性权重,权重值越大,对应的关联性越强。该权重由注意力机制计算所得:

$$\alpha_{im}^{src}=\frac{\exp(e_{im}^{src})}{\sum_{s=1}^{|X|}\exp(e_{is}^{src})} \tag{11}$$

$$e_{im}^{src}=\vartheta^{src\,T}\tanh(W^{src}h_i+U^{src}h_m) \tag{12}$$

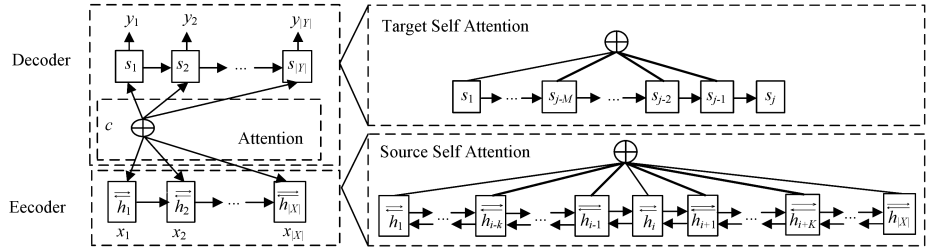
其中,$|X|$ 为源端句子长度,$\vartheta^{src\,T}$,$W^{src}$ 和 $U^{src}$ 为可训练的权重矩阵。

#### 4.2.2　损失定义

源端近邻关联损失 $\Delta_{src}$ 定义如下:

$$\Delta_{src}=-\sum_{i=1}^{|X|}\log\Big(\sum_{m=i-K}^{i+K}\alpha_{im}^{src}\Big) \tag{13}$$

其中，$m\neq i$，$K$ 为选定的左右窗口大小。对于 $i<K$ 部分，我们设置左近邻窗口为 $i-1$，右近邻窗口为 $K$；对于 $i+K>|X|$ 部分，我们设置左近邻窗口为 $K$，右近邻窗口为 $|X|-K$。通过设计的损失函数，使得每时刻源端隐藏层与其左右 $K$ 近邻隐藏层更相关。




注：第 1 部分为源端近邻关联，第 2 部分为目标端近邻关联

图 2 倾向源端近邻和目标端近邻模型的结构图

Fig. 2 Illustration of model inclined to source and target networks

### 4.3 倾向目标端近邻关联的神经机器翻译

鉴于基准系统未对目标端关联性建模，本文提出倾向目标端近邻关联的神经机器翻译模型。具体实现为：对应目标端第 $j$ 个词，由于目标端单词未来信息不确定，我们选取的窗口为历史 $M$，使得第 $j$ 个目标端单词隐藏层与其历史 $M$ 个单词隐藏层更关联，如图 2 中第 2 部分虚线框所示，$s_j$ 与历史 $M$ 时刻的关联性如黑色加粗实线所示。

倾向目标端近邻关联模型的具体实现分为两部分：1）计算每个目标端单词隐藏层与所有历史目标端单词隐藏层的关联性权重；2）设计合理的损失函数，使得每个目标端单词隐藏层与其历史 $M$ 个单词隐藏层更相关。

#### 4.3.1 关联性计算

$\alpha_{jk}^{trg}$ 为目标端第 $j$ 个单词隐藏层与第 $k$ 个单词隐藏层的关联性权重，权重越高，对应的关联性越强。该权重由注意力机制计算所得：

$$\alpha_{jk}^{trg}=\frac{\exp(e_{jk}^{trg})}{\sum_{o=1}^{|j-1|}\exp(e_{jo}^{trg})} \tag{14}$$

$$e_{jo}^{trg}=\vartheta^{trg^T}\tanh(W^{trg}s_j+U^{trg}s_o) \tag{15}$$

其中，$|j-1|$ 为历史 $j-1$ 时刻，$\vartheta^{trg^T}$，$W^{trg}$ 和 $U^{trg}$ 为可训练的权重矩阵。

#### 4.3.2 损失函数定义

目标端近邻关联损失 $\Delta_{trg}$ 定义如下：

$$\Delta_{trg}=-\sum_{j=1}^{|Y|}\log\Big(\sum_{k=j-M}^{j-1}\alpha_{jk}^{trg}\Big) \tag{16}$$

其中，$|Y|$ 为目标端句子长度，$M$ 为选定的历史窗口大小。对于 $j<M$ 的部分，我们设置对应的近邻关联损失为 0。通过设计的损失函数，使得每时刻目标端隐藏层与其历史 $M$ 个隐藏层更相关。

### 4.4 混合近邻关联的神经机器翻译

鉴于基准系统未对源端以及目标端关联性建模，在混合模型中，我们同时对源端以及目标端近邻关联建模，使得源端以及目标端包含更丰富的近邻关联表示，以此促进神经机器翻译的效果。倾向源端近邻关联和倾向目标端近邻关联及其损失函数详见 4.2 节和 4.3 节。

## 5 实验

### 5.1 实验设置

本文通过中英机器翻译任务验证提出模型的有效性。实验训练语料为包含 125 万句的 LDC 双语平行语料[1)]。使用 NIST06 作为开发集，NIST02，NIST03，NIST04，NIST05，NIST08 作为测试集；使用 4 元的 NIST BLEU 作为评测标准，评测脚本为 multi-bleu. perl[2)]，英文单词不区分大小写。

通过将本文方法与基准系统进行对比来验证所提模型的有效性。在本实验中，使用基于 dynet[3)] 的神经机器翻译系统 Lamtram[14]，系统中使用 LSTM 作为 RNN 单元。

在 Lamtram 基准系统中，我们只保留了中英语料中前 3 万个高频词，高频词覆盖了大约 97.7% 和 99.3% 的中文和英文词汇。实验中未对双语句对做长度限制。我们使用 Adam[15] 作为优化器，分别设置学习率为 0.001（$\text{Lamtram}_1$）和 0.0001（$\text{Lamtram}_2$）作为两套参考基准系统，设置 Dropout 为 0.5。每个系统训练 20 轮并选取开发集上困惑度（PPL）最低的模型作为最好模型解码。在解码时，我们使用集束搜索策略（beam-search），设置 beam 的大小为 10。其他训练参数使用默认配置。我们在两种不同学习率的配置上验证所提模型的有效性。

### 5.2 实验结果

基于 Lamtram，我们实现了倾向近邻关联的神经机器翻译方法。实验结果如表 1、表 2 所列，表 2 中的实验结果为在测试集上使用 multi-bleu. perl 测得。

表 1 不同 $K$ 以及 $M$ 的实验对比

Table 1 Comparision between different $K$ and $M$

| 系统 | NIST02 | NIST03 | NIST04 | NIST05 | NIST08 | AVG | |
|---|---|---|---|---|---|---|---|
| $\text{Lamtram}_1$ | 37.42 | 35.46 | 38.32 | 35.71 | 26.03 | 34.59 | — |
| $+\Delta_{src}(K=1)$ | 38.77 | 36.27 | 38.88 | 36.45 | 26.75 | 35.42 | +0.83 |
| $+\Delta_{src}(K=2)$ | 38.74 | 36.49 | 38.95 | 36.10 | 27.09 | 35.47 | **+0.88** |
| $+\Delta_{src}(K=3)$ | 38.41 | 36.57 | 38.90 | 36.45 | 26.75 | 35.42 | +0.83 |
| $+\Delta_{trg}(M=1)$ | 37.94 | 37.00 | 38.54 | 36.27 | 27.38 | 35.43 | **+0.84** |
| $+\Delta_{trg}(M=2)$ | 37.83 | 36.08 | 38.62 | 36.03 | 26.18 | 34.95 | +0.36 |
| $+\Delta_{trg}(M=3)$ | 38.53 | 36.01 | 38.63 | 36.06 | 26.38 | 35.17 | +0.58 |

1) 语料包含 LDC2002E18，LDC2003E07，LDC2003E14 以及 LDC2004T07，LDC2004T08，LDC2005T06 中议事录部分

2) https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

3) https://github.com/neubig/lamtram

表 2　中英实验结果

Table 2　Chinese-English results

| 系统 | NIST02 | NIST03 | NIST04 | NIST05 | NIST08 | AVG | |
|---|---|---|---|---|---|---|---|
| $\text{Lamtram}_1$ | 37.42 | 35.46 | 38.32 | 35.71 | 26.03 | 34.59 | — |
| $+\Delta_{src}$ | 38.74 | 36.49 | 38.95 | 36.10 | 27.09 | 35.47 | +0.88 |
| $+\Delta_{trg}$ | 37.94 | 37.00 | 38.54 | 36.27 | 27.38 | 35.43 | +0.84 |
| $+\Delta_{src}+\Delta_{trg}$ | **38.72** | **37.04** | **39.08** | **37.40** | **27.72** | **35.99** | **+1.40** |
| $\text{Lamtram}_2$ | 38.87 | 36.30 | 39.21 | 37.00 | 26.86 | 35.65 | — |
| $+\Delta_{src}$ | 39.11 | 37.09 | 39.81 | 37.70 | 27.77 | 36.30 | +0.65 |
| $+\Delta_{trg}$ | 39.25 | 37.47 | 39.65 | 37.50 | 27.55 | 36.28 | +0.63 |
| $+\Delta_{src}+\Delta_{trg}$ | **39.97** | **37.60** | **40.39** | **38.21** | **28.15** | **36.86** | **+1.21** |

基准系统的实验结果分别如表 2 中 $\text{Lamtram}_1$ 以及 $\text{Lamtram}_2$ 所示。通过训练得到参数模型 $\text{Model}_{\text{base}}$。

倾向源端近邻关联的具体实现为:固定模型 $\text{Model}_{\text{base}}$ 中除了源端 RNN 部分的参数,在此基础上增加源端近邻关联参数(见式(11)－式(12)),得到模型参数 $\text{Model}_{\text{base+src}}$,在最小化交叉熵损失 $loss_{word}$ 的同时最小化 $\Delta_{src}$,来优化源端近邻关联。我们分别选取窗口 $K=1,2,3$ 进行实验,实验结果如表 1 中 $+\Delta_{src}$ 部分所示。通过分析实验结果可知:当窗口 $K$ 设置为 2 时 BLEU 提升了 0.88,优于窗口 $K=1$(+0.83)以及 $K=3$(+0.83)的结果,因此后续实验源端近邻窗口 $K$ 默认设置为 2。表 2 中 $+\Delta_{src}$ 部分为在两套基准系统基础上分别增加了源端近邻关联指导。通过实验结果可知,$+\Delta_{src}$ 分别比基准系统高出 0.88 以及 0.65 的 *BLEU* 值。

倾向目标端近邻关联的具体实现为:在 $\text{Model}_{\text{base}}$ 的基础上添加目标端近邻关联参数(见式(14)－式(15)),最小化 $\Delta_{trg}$ 以及交叉熵损失 $loss_{word}$,以此优化目标端近邻关联以及正确单词概率,训练所得参数模型为 $\text{Model}_{\text{base+trg}}$。我们分别选取了窗口 $M=1,2,3$ 进行实验,实验结果如表 1 中 $+\Delta_{trg}$ 部分所示。通过分析实验结果可知:当窗口 $M$ 设置为 1 时 *BLEU* 提升了 0.84,优于窗口 $M=2$(+0.36)以及 $M=3$(+0.58)时的结果,因此后续实验中目标端历史窗口 $M$ 默认设置为 1。表 2 中 $+\Delta_{trg}$ 部分为在两套基准系统基础上分别增加目标端近邻关联指导。通过实验结果可知,$+\Delta_{trg}$ 部分别比基准系统高出 0.84 以及 0.63 的 *BLEU* 值。

混合近邻关联的具体实现为:固定模型参数 $\text{Model}_{\text{base+trg}}$ 中除了源端 RNN 部分的参数,在此基础上添加源端近邻关联参数,得到模型参数 $\text{Model}_{\text{base+src+trg}}$,在目标端近邻关联的基础上最小化交叉熵损失 $loss_{word}$ 的同时最小化优化 $\Delta_{src}$,来优化源端近邻关联,通过两部分模型融合训练,来获得更好的源端以及目标端近邻关联表示。我们使用了源端近邻窗口 $K=2$ 以及目标端近邻窗口 $M=1$ 的默认配置进行实验,实验结果如表 2 中 $+\Delta_{src}+\Delta_{trg}$ 所列。通过实验结果可知,$+\Delta_{src}+\Delta_{trg}$ 分别比基准系统高出 1.40 以及 1.21 的 *BLEU* 值。

### 5.3　训练速度

本节分析了不同实验系统下训练时间的对比,具体结果如表 3 所列。

对于倾向源端近邻关联的实验,我们使用训练了 20 轮的基准系统最好模型作为参数初始化,并在此基础上添加源端近邻关联参数,继续训练了 20 轮,该部分训练时间与基准系统持平。

对于倾向目标端近邻关联的实验,我们在基准系统基础上添加目标端近邻关联参数,并训练了 20 轮,该部分训练时间为基准系统的 1.7 倍左右。

混合近邻关联为在倾向目标端近邻关联的实验的基础上添加源端近邻关联,该部分训练时间与基准系统持平。

表 3　训练时间

Table 3　Training time

| 实验系统 | 时间/倍 |
|---|---|
| Baseline | +0 |
| +src | +1.0 |
| +trg | +0.7 |
| +src+trg | +1.7 |

### 5.4　一元 *BLEU* 值

一元 *BLEU* 值可以较好地衡量单个单词在译文中出现的比率。如表 4 实验结果所示,混合近邻关联(+src+trg)对应的一元 *BLEU* 值提升明显(+1.58),倾向源端以及目标端近邻关联的一元 *BLEU* 值有略微提升,分别为 +0.04 以及 +0.8*BLEU* 值。

表 4　测试集上的平均一元 *BLEU*

Table 4　Averaged 1-gram *BLEU* scores on test sets

| 实验系统 | 一元 *BLEU* 值 |
|---|---|
| Baseline | 73.84 |
| +src | 73.88 |
| +trg | 74.68 |
| +src+trg | **75.42** |

### 5.5　开发集困惑度

图 3 为所提出方法在开发集合 NIST06 上对应的不同训练轮次困惑度(PPL),*PPL* 越低,对应的模型训练效果越好。图 3 中横轴为训练轮次,纵轴为开发集上对应的 *PPL*。

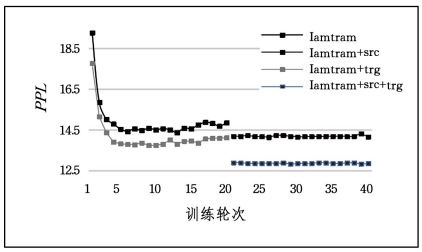



图 3　训练轮次变化对应的开发集上困惑度

Fig. 3　*PPL* changes in terms of numbers of training epochs

通过图 3 可知:增加倾向源端近邻关联模型(lamtram+src)对应的 *PPL* 低于基准系统(lamtram)的 *PPL*,增加倾向目标端近邻关联模型(lamtram+trg)对应的 *PPL* 与增加源端近邻关联模型(lamtram+src)对应的 *PPL* 持平,同时对源端和目标端近邻关联建模(lamtram+src+trg)对应的 *PPL* 低于分别对源端及目标端建模的 *PPL*。

### 5.6　源端不同长度的译文 *BLEU* 值

本节测试了在所有测试集中不同输入长度对应的译文 *BLEU* 值,对应数据如图 4、表 5 所示。

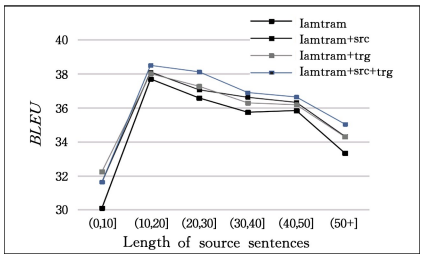



图 4 不同源端长度对应的译文 BLEU 值

Fig. 4 BLEU scores of generated translations on test set with respect to lengths of sentences

表 5 不同源端长度译文 BLEU 值增幅

Table 5 Growth of BLEU with respect to lengths of sentences

| | base | $+\Delta_{src}$ | $+\Delta_{trg}$ | $+\Delta_{src}+\Delta_{trg}$ |
|---|---|---|---|---|
| (0,10] | — | +1.53 | **+2.15** | +1.53 |
| (10,20] | — | +0.41 | +0.31 | **+0.80** |
| (20,30] | — | +0.48 | +0.69 | **+1.54** |
| (30,40] | — | +0.88 | +0.55 | **+1.16** |
| (40,50] | — | +0.48 | +0.34 | **+0.80** |
| (50+) | — | +0.99 | +0.96 | **+1.71** |

由图 4 和表 5 的数据可以看出：

1)所提方法在不同源端长度上解码的结果均优于基准系统。

2)分别对源端近邻关联建模($+\Delta_{src}$)和对目标端近邻关联建模($+\Delta_{trg}$)对应的不同源端输入长度的译文 *BLEU* 值的增幅基本持平，在源端输入长度为(0,10]时$+\Delta_{trg}$增幅最大，为+2.15*BLEU* 值。

3)对源端以及目标端近邻关联($+\Delta_{src}+\Delta_{trg}$)建模对应的不同源端输入长度的译文 *BLEU* 值远高于基准系统(base)。在源端句子长度为(20,30]以及句长大于 50(50+)时，译文 *BLEU* 值提升得更为明显，分别提升了 1.54 和 1.71 个 *BLEU* 值。

**结束语** 本文提出了倾向近邻关联的神经机器翻译模型，在基准系统的基础上分别对源端近邻关联以及目标端近邻关联建模。通过表 1、表 2 的实验结果可知，添加了源端和目标端近邻关联指导的模型译文效果优于基准系统，添加了融合源端目标端近邻关联指导的模型的译文效果明显优于基准系统。

目前，我们采用的方法仅考虑了源端左右 $K$ 个信息以及目标端历史 $M$ 个信息，未考虑丰富的语言学信息，而句法树以及依存树等提供了更丰富的关联性信息。在未来的工作中，我们考虑将句法树以及依存树信息融入到机器翻译模型中，使得翻译系统能融入更多的语言学信息，以此进一步提升机器翻译的性能。

## 参 考 文 献

[1] LI Y C, XIONG D Y, ZHANG M. A survey of neural machine translation [OL]. http://cjc.ict.ac.cn/online/bfpub/lyc-20171229152034.pdf.

[2] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[3] SUTSKEVER I, VINYALS O, LE Q V. Sequence to Sequence Learning with Neural Networks[C]// Advances in Neural Information Processing Systems. 2014:3104-3112.

[4] LUONGM T, PHAM H, MANNING C D. Effective Approaches to Attention-based Neural Machine Translation[J]. arXiv preprint arXiv:1508.04025, 2015.

[5] LIUL M, UTIYAMA M, FINCH A, et al. Neural Machine Translationwith Supervised Attention[J]. arXiv preprint arXiv:1609.04186, 2016.

[6] MI H T, WANG Z G, ITTYCHERIAH A. Supervised Attentions for Neural Machine Translation[J]. arXiv preprint arXiv:1608.00112, 2016.

[7] CHEN W H, MATUSOV E, KHADIVI S, et al. Guided Alignment Training for Topic-aware Neural Machine Translation[J]. arXiv preprint arXiv:1607.01628, 2016.

[8] GEHRING J, AULI M, GRANGIER D, et al. Convolutional Sequence to Sequence Learning[J]. arXiv preprint arXiv:1705.03122, 2017.

[9] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All You Need [C]// Advances in Neural Information Processing Systems. 2017:5998-6008.

[10] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vectorspace[J]. arXiv preprint arXiv:1301.3781, 2013.

[11] HOCHREITER S, SCHMIDHUBER J. Long Short-term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[12] CHO K, VAN MERRIENBOER B, GULERHRE C, et al. Learning Phrase Representations using RNN Encoder-decoder for Statistical Machine Translation [J]. arXiv preprint arXiv:1406.1078, 2014.

[13] NEUBIG G, DYER C, GOLDBERG Y, et al. Dynet: The Dynamic Neural Network Toolkit[J]. arXiv preprint arXiv:1701.03980, 2017.

[14] NEUBIG G. lamtram: A Toolkit for Language and Translation Modeling using Neural Networks[OL]. http://www.github.com/neubig/lamtram.

[15] KINGMA D, BA J. Adam: A Method For Stochastic Optimization[J]. arXiv preprint arXiv:1412.6980, 2014.