融合篇章上下文有效识别的篇章级机器翻译

汪浩，贡正仙*，李军辉

（苏州大学计算机科学与技术学院，江苏 苏州 215006）

摘要：神经机器翻译在机器翻译领域的发展势头正猛，在很多语言对之间取得的效果已超过传统的统计机器翻译。篇章翻译是近来兴起的研究热点，如何能够在翻译文档时充分利用篇章信息一直是该研究的关键点和难点。在篇章级机器翻译中，如何选取当前句的篇章上下文是非常关键的。虽然相关研究使用的篇章上下文不尽相同，但是却少有在选取之前对上下文信息进行识别筛选。本文提出了一种融合篇章上下文有效识别的篇章级翻译模型，引入判别篇章上下文是否有效的分类任务，并根据判别结果来控制目标端对篇章上下文的利用。在中英、英德翻译任务上，与基准系统相比，都得到了显著的提升。

关键词：神经机器翻译；联合学习；篇章翻译

中图分类号： TP-391.2　　**文献标志码：**A

机器翻译研究的是如何利用计算机技术，来实现两种语言甚至多种语言之间的相互翻译。近些年来，神经机器翻译飞速发展[1-2]。相比较于传统的统计机器翻译，神经机器翻译展现出更加优越的性能、更强的适应性，使得神经机器翻译的应用和影响越来越广。神经机器翻译通过搭建神经网络模型，使用序列到序列的翻译方法[3]，极大地提升了翻译的性能。2017 年，Vaswani 等[4]提出全新的模型架构 Transformer，该模型采用经典的编码器-解码器结构，使用新颖的多头注意力机制来进行建模，达到了目前非常卓越的性能，在 NLP 各个研究方向得到广泛应用。

神经机器翻译研究对于句子级的翻译，取得了非常卓越的效果，在中英新闻翻译方面取得了巨大的成功，已经达到接近人类的水平[5]。然而，相较于句子级翻译研究，篇章级的神经机器翻译受到的关注较少。由于难以有效地利用篇章上下文信息，篇章级机器翻译的研究仍然具有挑战性。对于篇章翻译，即使是 Transformer 模型性能也有待提高，因为它独立地翻译篇章中的每句话，存在难以捕捉篇章上下文信息的问题。Bawden 等[6]指出利用篇章级上下文来处理与上下文相关的现象是非常关键的，例如共指、词法衔接和词法歧义化等现象，对于机器翻译的性能有着重要的影响。翻译过程中出现的语句不通顺、不连贯的现象，很有可能就是没有考虑篇章级信息而造成的。

近来，篇章级别的 NMT 研究越来越受到关注[7-12]。Zhang 等[7]提出了一种基于 Transformer 的篇章翻译模型，通过使用两个编码器分别计算当前句的表示和篇章级上下文的表示，再将篇章级别上下文信息分别融入到当前句子的编码器和解码器中，显著地提高了篇章翻译的性能。HAN[11]是第一个使用分层注意力机制以结构化和动态的方式来捕捉篇章上下文信息的模型。Yang[10]等提出了一种以查询为导向的胶囊网络，将上下文信息聚类到目标翻译可能涉及的不同角度。但是，篇章上下文的选取存在着很多问题，Zhang 选取前一、二、三句作为当前句子的上下文，HAN 选取了前三句作为上下文，Yang 同样是选取了前三句作为上下文。 然而在实际翻译场景中，当前句的前句所蕴含的篇章信息并非总能对翻译当前句发挥作用，因此，如何让模型能够充分利用有效的篇章信息是一个值得研究的问题。

为了让模型能够学习到更多有价值的篇章上下文信息，本文提出一种基于联合学习的篇章翻译结构。其核心思想是利用一个共享的编码器分别对当前句及其上下文进行编码，同时引入分类任务，分类器用于判断当前句的上下文编码表征是否蕴含有效的篇章信息。本文对分类器的数据生成设计了多种方案，在努力提高分类性能的同时帮助模型提升编码器对句子的表示能力。本文描述的篇章

基金项目：国家自然科学基金（61976148）；国家自然科学基金（61876120）

***通信作者：**zhxgong@suda.edu.cn

翻译系统建立在通用的句子级 Transformer 模型之上，除了联合分类器，模型进一步引入门控机制，通过改进基准模型的残差连接，使得编码端识别出来的上下文信息能够在解码端得到充分利用。本文提出的方法对机器翻译模型的性能有明显的提升，尤其在中英翻译任务上，比基准模型的性能提升了 1.6 BLEU（Bilingual Evaluation Understudy）值。

1 基准 Transformer 模型

本文选用的基准 NMT 模型是基于注意力机制的神经机器翻译模型 Transformer。2017 年，谷歌的研究者提出了一种全新的神经机器翻译模型 Transformer[4]。Transformer 是基于编码器-解码器结构的模型。编码器采用 N_e 层相同的网络层叠加而成的，每一层编码器层是由一个多头自注意力子层和一个全连接前馈神经网络构成的。类似的，解码器也是采用 N_d 层完全相同的网络层叠加起来，每一层解码器层是由一个多头自注意力子层，一个多头上下文注意力子层和一个全连接前馈神经网络构成。除此之外，为了解决梯度爆炸或消失和训练过程不稳定等问题，Transformer 在子层之间加入残差连接，在每层中加入了层正则化操作。

Transformer 采用一种非常独特的多头注意力机制（multi-head attention）。多头注意力机制的优点在于能够直接对序列中任意两个单元之间的关系进行建模，这使得长距离依赖等问题可以更好地被求解。此外，多头注意力机制克服了循环神经网络无法并行化的缺点，因此模型训练的速度更快。多头注意力机制会得到多个不同的表征 Query（Q），Key（K），Value（V），然后进行注意力机制的运算，这个过程可以用式（1）进行表述：

$$Attention(Q,K,V) = Soft\max\left(\frac{QK^T}{\sqrt{d^k}}\right)V \tag{1}$$

其中， d^k 表示 Key 的维度。

因为，注意力机制会在输入序列顺序上的出现缺失，Transformer 采用了一种位置编码的方式来模拟输入序列的时序信息，通过将词向量与位置编码相加生成最终的源端句表示。位置编码的计算方式如式（2）~式（3）所示：

$$PE_{pos,2i} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2}$$

$$PE_{pos,2i+1} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{3}$$

其中， pos 表示单词在输入序列中的位置， d_{model} 表示模型的隐层单元个数。

2 融合篇章上下文有效识别的篇章级机器翻译

2.1 系统模型架构

受 Zhang 等[7]工作的启发，本文采用了如图 1 所示的系统模型架构，基本思想是使用多头自注意力机制计算篇章级上下文的表示，并将编码得到的上下文信息集成到解码器中。与 Zhang 等[7]不同，出于计算成本和注意力负担的考量，本文对上下文和当前句采用的是共享编码器，即在 Transformer 的原始模型基础上使用同一个编码器来对源端句的上下文句进行编码。为了有效利用篇章级别的上下文信息，系统加入了分类模块（图 1 所示的 binary classifier），分类器将对两个编码器的输出隐藏状态进一步处理，目的是从经过编码的信息中，判断上下文与当前句是否存在关联，一方面通过分类任务来提升句子的表示能力，另一方面将联合上下文注意力和门控机制来指导模型对上下文的有效利用。篇章上下文注意力层（图 1 所示的 Context Attention）位于解码器的自注意子层

和上下文注意力子层之间，为了指导解码器有效使用篇章上下文信息；门控机制（图 1 所示的 Context Gate）的引入为了进一步控制上下文信息在目标端的利用。下面将详细描述分类器的设计和门控机制。

2.2 篇章上下文有效识别

2.2.1 分类器设计

句子级的 Transformer 模型在翻译时不需要考虑上下文，因此经过编码器编码的句子表示（sentence representation）在体现句间关系上能力较弱；与此相对，篇章翻译的编码输出应该在句间关系的判断上更具优势。因此，存在这样一个假设：性能越好的篇章翻译编码器，它的编码输出（即句子表示）在句间关系判断上应该能力越强（本文 4.4 节的实验结果验证了该假设的合理性）。根据这一假设，本文的翻译系统增加了一个联合分类任务：利用编码器输出判断当前句子的上下文是不是真正有效的上下文。分类性能的提升，一方面说明系统内的句子表示得到增强，另一方面也将说明模型在学习过程中越能有效关注到上下文信息。

分类器设计的核心之一是构建训练语料。受 Bert[13]工作的启发，系统将当前句的前一句作为有效上下文，即为正例。原理上篇章中除去当前句子的前一句都有可能作为负例，为了提升分类效果，系统选取篇章中相似度最低的句子作为无效上下文，即为负例。在构造负例过程中，本文使用了 TF-IDF 和 KL_div 两种策略来计算句子之间的相似度。在 TF-IDF 策略中，利用 Sklearn 库中的 TfidfVectorizer 可以把原始文本转化为 tf-idf 的特征矩阵，然后计算句子表征之间的余弦相似度。给定转化好句子表征 $X^{(i)}$ 和 $X^{(j)}$，TF-IDF 策略的相似度计算如式（4）所示：

$$sim_{TF}\left(X^{(i)},X^{(j)}\right)=\frac{X^{(i)}\cdot X^{(j)}}{\left|X^{(i)}\right|\left|X^{(j)}\right|} \tag{4}$$

在 KL_div 策略中，利用训练好的 BERT 预训练模型[1]将句子转化为特征向量，计算句子表征之间的 KL 距离。KL_div 策略的相似度计算如式（5）所示：

$$sim_{KL}\left(X^{(i)}\left\|X^{(j)}\right.\right)=\sum X^{(i)}\log\left(\frac{X^{(i)}}{X^{(j)}}\right) \tag{5}$$

本文生成数据的正负例比为 1:3 或 1:1。

2.2.2 分类器结构

分类器的结构如图 2 所示，记 $h\in\Re^{n_t\times d}$ 和 $s\in\Re^{n_c\times d}$ 分别表示当前句 $x^{(t)}$ 和当前句上下文 $c^{(t)}$ 经编码器的输出，n_t 表示源端当前句包含的单词数，其中 n_c 表示对应上下文包含的单词数，d 表示隐藏状态的大小。将隐藏状态 h 和 s 进一步通过 Max-Pooling 和 Mean-Pooling 操作，分别表示为向量 u 和 v，即

$$u=\left[\max-pooling\left(h\right),mean-pooling\left(h\right)\right] \tag{6}$$

$$v=\left[\max-pooling\left(s\right),mean-pooling\left(s\right)\right] \tag{7}$$

其中 $u,v\in\Re^{2d}$。为了捕获 u 和 v 之间的关系，系统为分类器构建了 3 种输入：（1）将 u，v 拼接在一起 $[u,v]$；（2）逐个元素乘积 $u*v$；（3）逐元素的绝对差 $|u-v|$；即：

$$o=\left[u,v,\left|u-v\right|,u*v\right] \tag{8}$$

其中 $o\in\Re^{6d}$。最后，通过全连接层和二元分类 softmax 层判断上下文 $c^{(t)}$ 是否为当前句 $x^{(t)}$ 的有效上下文：

$$z=soft\max\left(\sigma\left(oW_1+b_1\right)W_2+b_2\right) \tag{9}$$

[1] https://github.com/hanxiao/bert-as-service

其中 $W_1, W_2 \in \Re^{6d\times 2}$， $b_1, b_2 \in \Re^2$ 为模型参数， σ 为 sigmoid 函数。

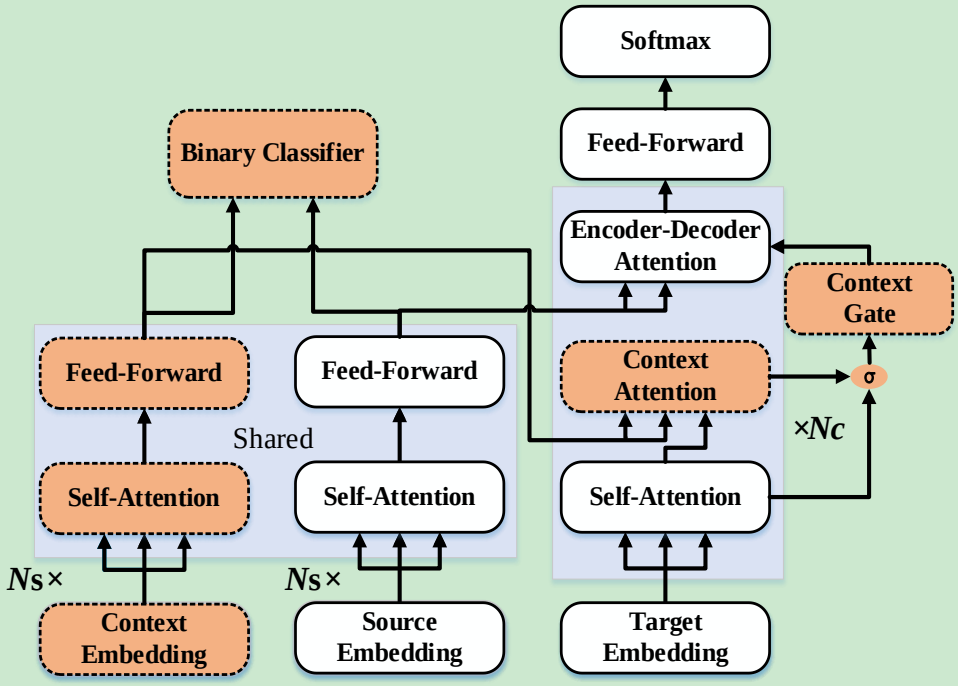

图 1 融合篇章上下文有效识别的 Transformer 模型

Fig. 1 Transformer model that effectively recognizes and exploits context

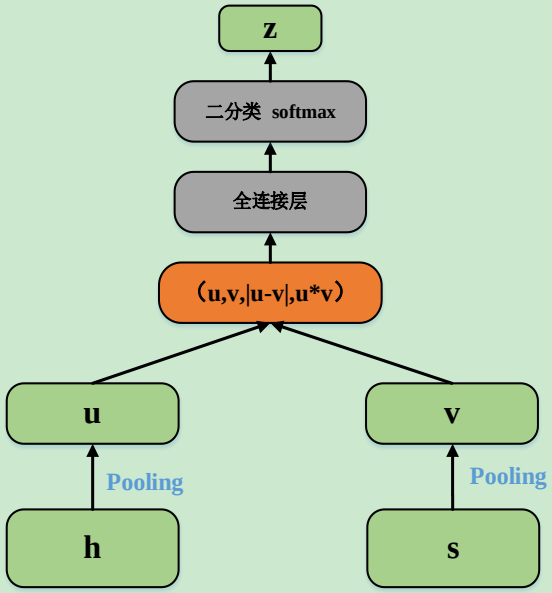

图 2 处理篇章信息的分类器结构图

Fig. 2 Structure diagram of classifier processing document-level information

2.3 带门控机制的残差网络

在图 1 所示的解码器端，引入的上下文注意力子层（context attention sub-layer）用于融合篇章上下文信息。为方便起见，记解码端自注意力层的输出为 $p \in \Re^{m_t\times d}$ ，上下文注意力层通过使用多头注意力模型，融入篇章上下文信息 $s \in \Re^{n_c\times d}$ ，即：

$$q = multi-head\left(p,s,s\right) \tag{10}$$

其中 $q \in \Re^{m_t \times d}$ 。为了获取的篇章上下文信息，一般通过残差网络与上下文注意力子层的输入进行融合，即：

$$r = q + p \tag{11}$$

不难看出，式（11）所示的残差网络将采用全盘接收的方式融合获取的篇章上下文 q 。为了区别篇章上下文是否有效，受文献[7]的启发，本文引入带门控机制的残差网络，即通过门控层按一定比例融合获取的篇章上下文 q ，如式（12）所示:

$$r = \alpha q + p \tag{12}$$

其中权重 α 为衡量该篇章上下文信息重要程度的指标，计算如式（13）所示：

$$\alpha = \sigma\left(W_i p + W_j q\right) \tag{13}$$

公式中σ表示 sigmoid 函数， W_i, W_j 是权重参数。一般来讲，当篇章上下文有效时，我们期望权重 α 接近于 1 ，反之接近于 0。

3 训练方式

3.1 联合学习

与第 2 节内容相照应，本文联合学习的损失函数分为三部分，分别涉及传统神经机器翻译目标端的预测、有效篇章上下文识别和带门控机制的残差网络中篇章上下文重要程度 α 。

记模型参数为 θ ，神经机器翻译目标端预测相关损失函数如下所示：

$$L_1 = -\sum_{n=1}^{N}\sum_{t=1}^{T_n}\log p\left(y^{(t)} \mid X^n, Y_{<t}^n; \theta\right) = -\sum_{n=1}^{N}\sum_{t=1}^{T_n}\sum_{j}^{m_{n,t}}\log p\left(y_j^{n,(t)} \mid x^{n,(t)}, c^{n,(t)}, y_{<j}^{n,(t)}; \theta\right) \tag{14}$$

其中 N 指训练集中平行篇章数， T_n 为第 n 个平行篇章的句子数， X^n 和 Y^n 分别为第 n 个平行篇章的源端和目标端。

篇章上下文有效识别相关损失函数如下所示：

$$L_2 = -\sum_{i=1}^{M} k_i \log(z(k_i \mid \theta)) + (1-k_i)\log(1-z(k_i \mid \theta)) \tag{15}$$

其中 M 为有效篇章上下文识别训练样例数，即为机器翻译训练集中句子总数； $k_i \in \{0,1\}$ 为第 i 个句子的篇章上下文有效性标签， $z(k_i \mid \theta)$ 为模型预测的概率。

带门控机制的残差网络中篇章上下文重要程度 α 相关的损失函数如下所示：

$$L_3 = -\sum_{i=1}^{M}\sum_{j=1}^{6}\frac{1}{2}(k_i - \alpha_{i,j})^2 \tag{16}$$

其中 $\alpha_{i,j}$ 表示第 i 个句子在解码器第 j 层中的带门控机制的残差网络中篇章上下文重要程度 α 值，6 表示解码器的层数。

最终，我们的联合学习的训练目标如式（17）所示：

$$q^* = \arg\min_{q}(L_1 + L_2 + L_3) \tag{17}$$

3.2 两步训练法

由于大规模篇章级平行语料库通常无法获取，即便是对于语料丰富的语言（如英语和中文）也

是如此。在语料库规模很小的条件下，篇章级神经机器翻译模型的表现往往不及句子级模型性能。本文采用 Zhang[7]中提到的两步训练的训练方式。该训练方式额外使用了一个比篇章级语料数据 D_d 更大的句子级平行语料数据 D_s，具体训练步骤如下：

第一步，在组合的句子级平行语料库上 $D_d \cup D_s$ 上训练句子级参数。训练句子级参数时，模型与原始 Transformer 模型相同，篇章级模块没有参与训练，如图 1 所示，进行第一步训练的时候只有实线无色模块参与训练。

第二步，仅在篇章级平行语料库 D_d 上训练篇章级参数。如图 1 所示，在第二步中虚线橙色模块和实线无色模型同时参与训练。

值得注意的是，本模型在训练篇章级参数时，并没有固定第一步句子级参数，这主要因为分类器模板需要两个编码器同时参与训练。

4. 实验

4.1 实验设置

本文在中英、英德翻译任务上验证本文提出的模型。

在中英翻译任务上，训练集包含 200 万个中英平行句子对，训练集包括句子级平行语料 LDC2002E18、LDC2003E07、LDC2003E14、LDC2004T08 的新闻部分；篇章级平行语料包括 LDC2002T01、LDC2004T07、LDC2005T06、LDC2005T10、LDC2009T02、LDC2009T15、LDC2010T03。篇章级平行语料包括 41K 个篇章，由 940K 个句子对组成。本文选择 NIST 2006 数据集作为开发集，包含 1,664 个句子，将 NIST 2002、 2003、 2004、 2005 、2008 作为测试集，分别包含 878、919、1,788、1,082、1,357 个句子。翻译任务的评估指标是对大小写不敏感的 BLEU 值，由 mteval-v11b.pl 脚本计算得出的。中英翻译中对源语言和目标语言都使用 BPE[14]，以减少未登录词的数量，操作数为 40K。

在英德翻译任务上，训练集包括 News Commentary v11 corpus[1]，并且从 WMT14 英德[2]中抽取 0.3M 个句子对用来训练句子级参数。开发集设置为 news-test2015，测试集为 news-test2016。对于该翻译任务数据，为了提高训练效率，本文将长篇章分成最多 30 个句子的子篇章，而且同样对源端和目标端语言使用了 BPE，操作数为 30K。

本文配置的 Transformer 模型来源于 OpenNMT[3]。[15]本文将编码器与解码器的层数设置为 6 层，多头注意力机制中含有 8 个头，同时设置 Dropout 值为 0.01，隐层维度和前馈神经网络中间层单元数分别为 512 和 2,048。本文实验只使用一个 GTX1080Ti 显卡训练模型，训练时批处理大小为 4,096 个 token 以内，。进行解码时，设置 beam_size 为 5，所有其他的设置使用 Vaswani 系统中的默认设置。本实验中，第一步句子级模型的训练时间为 28 个小时，第二步篇章级模型的训练时间为 12 个小时。

4.2 实验结果

为了验证本文提出模型的性能，我们分别在中英和英德翻译任务进行实验。表中 TF-IDF 策略指在本文提出的模型下使用 TF-IDF 方法生成的负例训练数据进行篇章级翻译实验，生成数据的正负例比为 1:3； KL_div 策略为通过计算 KL 距离生成的训练数据进行实验，生成的正负例比为 1:1。这两种策略选出来的篇章上下文长度都为一句。

表 1 展示的为中英翻译任务上，本文提出模型与基准系统在各测试集上的翻译性能。由表 1 可以看出，本文提出的两种方法均能够提高系统的翻译性能。尤其是，与基准系统相比，我们的模型使用 TF-IDF 策略提高了 1.6 个 BLEU 值，使用 KL_div 策略的方式提高了 1.45 个 BLEU 值。

[1] http://www.casmacat.eu/corpus/news-commentary.html
[2] http://www.statmt.org/wmt14/translation-task.html
[3] https://github.com/OpenNMT/OpenNMT-py

表 2 展示是在英德翻译任务上，本文提出模型与基准系统在各测试集上的翻译性能。由表 2 可看出，在英德任务上，本文提出的模型与基准系统相比也同样有显著提升，尤其使用 TF-IDF 策略构建数据的方法提高了 0.53 个 BLEU 值。

这充分表明了本文提出的融合篇章上下文有效识别的篇章翻译模型能够有效地提升机器翻译模型的效果。

表 1 本文模型与基准系统在中英翻译任务的 BLEU 值

Tab.1 The BLEU value of our model and the baseline in the Chinese-English translation tasks

System	NIST02	NIST03	NIST04	NIST05	NIST08	平均值	提升
Transformer	48.94	49.22	49.19	49.69	41.51	47.71	—
TF-IDF 策略	**51.03**	**51.75**	**50.89**	**51.25**	41.65	**49.31**	**+1.60**
KL_div 策略	50.89	51.48	50.72	50.78	**41.94**	49.16	+1.45

表 2 本文模型与基准系统在英德翻译任务的 BLEU 值

Tab.2 The BLEU value of our model and the baseline in the English-German translation tasks

System	Dev	Test	提升
Transformer	24.22	26.58	—
TF-IDF 策略	**24.26**	**27.11**	**+0.53**
KL_div 策略	24.24	26.90	+0.32

4.3 与相关工作的对比

本文复现了篇章机器翻译中非常经典的工作 Zhang[7]的模型代码，并在本文中英数据集上测得实验结果。对于 Zhang 的模型设置，本文采用了前一句作为上下文，上下文编码器的层数为 6 层，跟论文中设置保持一致。除此之外，本文模型与 Zhang 模型都是在同一个句子级模型参数上微调得到的。表 3 展示了在中英数据集上 Zhang 的系统和本文模型的 BLEU 比较。通过与 Zhang 模型相比，本文建议的模型具有更优的性能，在均值上提升了 0.32 个 BLEU，进一步证明了本文提出的基于联合学习的篇章翻译模型具有一定的优势。本文建议的模型使用的上下文为 TF-IDF 生成的正负例数据，正负例比为 1:3。

4.4 上下文有效识别探究

本文中通过引入分类器模块，对篇章上下文进行有效识别，使得源端编码表征能学习到更加有效的信息。因此，本文做了两组探究实验来证明本文模型增强源端表征的能力，额外做了一组实验来探究本文提出的带门控机制的残差网络对模型性能的影响。

第一组实验，通过对本文提出的分类器性能进行对比实验，来证明本文模型的编码器可以学习到更加有效的篇章信息。我们先利用句子级基准模型的编码器，分别对源端句和上下文句进行编码，将两个源端表征输入至分类器模块，测试分类器的性能。相同地，接着利用本文提出的篇章翻译模型的编码器进行编码，得到的两个源端表征来测试分类器性能。对比实验是在中英任务上进行的，上下文的选取方式采用 TF-IDF 策略，正负例比为 1:3。结果显示，使用基准模型的编码器测出分类

器的准确度为 73.0%，使用本文建议的模型编码器测出的分类器准确度为 76.7%，提高了 3.7%。可以看出，使用本文模型的编码器能够明显提升分类器性能，能够识别出更多有效的篇章上下文信息。

第二组实验，对比不同的上下文选取方式在篇章翻译任务上的性能，我们采用的比照对象为从篇章中随机挑选句子作为上下文来生成负例。对比实验是在中英翻译任务上进行的，生成数据的正负例比为 1:3。由表 4 展示，三种数据生成方式下模型在测试集上的性能。可以看出，本文提出的选取方式较盲目随机的选取上下文方式，具有明显的优势，尤其 TF-IDF 的选取方式较随机选取提高了 0.32 个 BLEU。可以得出，本文提出的数据生成方式具有优越性，更利于分类器识别出有效与无效上下文。

第三组实验中，本文将带门控的残差网络换成普通的残差网络，保留分类器模块。在中英翻译任务上进行验证试验，其他的实验设置同 4.1 中英实验设置，采用上下文长度为一句，TF-IDF 策略的正负例比为 1:3，KL_div 策略的正负例比为 1:1。由表 5 展示，普通的残差网络和带门控的残差网络对模型性能的影响。可以看出，移除带门控的残差网络后，在 TF-IDF 策略的上下文选择方式下模型性能有 0.32 个 BLEU 的下降，在 KL_div 策略的上下文选择方式中模型性能有 0.31 个 BLEU 的下降。可以得出，本文提出的带门控的残差网络能够在解码端对上下文信息进行更加有效的利用，能够使得模型的性能得到进一步提升。

表 3 本文模型与前人工作比较

Tab.3 Comparison of our model with previous work

System	NIST02	NIST03	NIST04	NIST05	NIST08	平均值	提升
Transformer	48.94	49.22	49.19	49.69	41.51	47.71	—
TF-IDF 策略	**51.03**	**51.75**	**50.89**	**51.25**	**41.65**	**49.31**	**+1.60**
Doc-NMT(Zhang 等)	51.02	50.83	50.51	51.20	41.40	48.99	+1.28

表 4 生成数据方式的 BLEU 对比

Tab.4 BLEU comparison of data generation methods

System	NIST02	NIST03	NIST04	NIST05	NIST08	平均值	提升
Transformer	48.94	49.22	49.19	49.69	41.51	47.71	—
随机策略	50.86	51.48	50.6	51.23	40.78	48.99	+1.28
TF-IDF 策略	**51.03**	**51.75**	**50.89**	**51.25**	41.65	**49.31**	**+1.60**
KL_div 策略	50.89	51.48	50.72	50.78	**41.94**	49.16	+1.45

5 总结

针对篇章机器翻译如何更高效利用上下文信息的问题，本文提出了一种新的模型结构。首先在现有 NMT 模型基础上融入了篇章信息，其次对篇章信息进一步识别处理，最终通过解码端的改进使得识别出的有效篇章信息得到充分利用。在中英和英德翻译任务中表明，本文提出的模型和方法可以有效地提高机器翻译的质量。

然而，本文模型在处理过多、过复杂的篇章上下文信息时，存在着处理能力不足，识别能力下降，无法让模型利用更有效的上下文信息。在未来的研究中，会针对模型的识别部分进行进一步改进和研究，努力完善模型，争取达到更好的效果。

表 5 不同残差网络的 BLEU 比较

Tab.5 BLEU comparison of different residual networks

System	NIST02	NIST03	NIST04	NIST05	NIST08	平均值
TF-IDF 策略	**51.03**	**51.75**	**50.89**	**51.25**	**41.65**	**49.31**
w/o context gating	50.91	51.21	50.54	50.98	41.29	48.99
KL_div 策略	**50.89**	**51.48**	**50.72**	50.78	**41.94**	**49.16**
w/o context gating	50.71	51.06	50.45	**51.05**	40.96	48.85

参考文献：

[1] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[EB/OL]. [2020-7-21]. http://arxiv.org/abs/1409.0473

[2] WU Y H, SCHUSTER M, CHEN Z, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation[EB/OL]. [2020-7-21]. http://arxiv.org/abs/1609.08144

[3] SUTSKEVER I, VINYALS O, LE Q B. Sequence to Sequence Learning with Neural Networks[C] // Advances in neural information processing systems. [S.I.]: NIPS, 2014: 3104-3112.

[4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C] //Advances in Neural Information Processing Systems. [S.I.]: arXiv, 2017: 5998-6008

[5] HASSAN H, AUE A, CHEN C, et al. Achieving Human Parity on Automatic Chinese to English News Translation[EB/OL]. [2020-7-21]. http://arxiv.org/abs/1803.05567

[6] BAWDEN R, SENNRICH R, BIRCH A, et al. EVALUATING DISCOURSE PHENOMENA IN NEURAL MACHINE TRANSLATION[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics. New Orleans: NAACL-HLT, 2018: 1304-1313.

[7] ZHANG J, LUAN H SUN M, et al. Improving the transformer translation model with document-level context [C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: empirical methods in natural language processing. [S.I.]: EMNLP, 2018: 533-542.

[8] XIONG H, HE Z, WU H, et al. Modeling coherence for discourse neural machine translation[C]//The Thirty-Third {AAAI} Conference on Artificial Intelligence. Honolulu: AAAI, 2019: 7338-7345

[9] VOITA E, SENNRICH R, TITOV I, et al. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion[C]//Proceedings of the 57th Conference of the Association for Computational Linguistics. [S.I.]: ACL, 2019: 1198-1212.

[10] YANG ZX, ZHANG JC, MENG FD, et al: Enhancing Context Modeling with a Query-Guided Capsule Network for Document-level Translation[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: EMNLP/IJCNLP, 2019: 1527-1537.

[11] WERLEN LM, RAM D, PAPPAS N, et al. Document-Level Neural Machine Translation with Hierarchical Attention Networks[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: EMNLP, 2018: 2947-2954.

[12] WANG L, TU Z, WAY A, et al. Exploiting cross-sentence context for neural machine translation[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. [S.I.]: EMNLP, 2017: 2826-2831.

[13] DEVLIN J, CHANG MW, LEE K, et al. Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis: NAACL-HLT, 2019: 4171-4186

[14] SENNRICH R, HADDOW B, BIRCH A, et al. Neural Machine Translation of Rare Words with Subword Units[EB/OL]. [2020-7-21]. http://arxiv.org/abs/1508.07909

[15] KLEIN G, KIM Y, DENG Y T, et al. OpenNMT: Open-source toolkit for neural machine translation[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. [S.I.]: ACL, 2017: 67-72.

Effectively Recognizes and Exploits Context for Document-level Neural Machine Translation

WANG Hao, GONG Zhengxian*, LI Junhui

(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

Abstract: Neural machine translation is gaining momentum in the field of machine translation, and its effects in many language pairs have surpassed traditional statistical machine translation. Recently, document-level machine translation is a research hotspot. However, how to make full use of document-level information when translating documents has always been the key and difficultly. Similarly, how to select the context of the current sentence is very critical. There are many selection methods in previous studies, however, it is rare to filter the context information before selection. This paper proposes a document-level translation model that incorporating effective context, introduces the classification task of identifying whether context is effective, and controls the use of context according to the identification results. Compared with the baseline system, our model has achieved strong improvement in Chinese-English and English-German translation tasks.

Keywords: neural machine translation; jointly learning; document-level neural machine translation