基于联合选择机制的篇章级神经机器翻译

陈林卿，李军辉[1]，贡正仙[2]

（苏州大学 自然语言处理实验室，江苏 苏州 215006）

摘 要：以往的篇章神经机器翻译研究工作大多将研究重心放在句子级上下文的利用方面，通过不同方式获取句子级上下文并将其与机器翻译模型结合以提高翻译性能。而利用篇章级上下文的研究工作大多需要对篇章语料中的整个文档进行计算，存在计算量过大及信息冗余的情况。该文提出使用软硬结合的上下文选择机制，使用基于句向量的轻量级计算即硬选择机制在全篇章内获取与当前句高度相关的上下文，再通过软选择机制将获取的全局上下文分配给源语言当前语句。实验表明该方法在有效约束模型参数及运算量的前提下从文档全局获取上下文帮助翻译模型并获得了有意义的性能提升。该文进一步分析了文档中篇章级上下文在当前句子周围的分布情况并观察到一些值得思考的实验现象。

关键字：神经机器翻译；篇章上下文；联合选择机制

中图分类号：TP391.2 文献标识:A

近年来神经机器翻译由于卓越性能引起学界关注成为机器翻译的主流研究方法[1-4]。在之前的句子级翻译模型中，给定一个段落翻译模型逐句进行翻译，句子与句子之间相互独立，忽略了篇章上下文信息对翻译过程的影响。已有研究表明优质有效的上下文有利于提升翻译质量，这一结论不仅适用于统计机器翻译[5,6]，也适用于神经机器翻译[7,8]。首先缺失有效信息可能造成翻译结果在语义上的错误从而造成句子不连贯通顺，并造成错误累积及传递。其次上下文中的有效信息可以帮助减少翻译结果中的歧义。

随着神经机器翻译需求场景逐渐从单独句子翻译向整个篇章翻译拓展，篇章翻译由此获得越来越多的关注。研究者们为了得到更好的上下文帮助模型进行篇章翻译，提出了一系列不同方法。利用篇章级上下文的传统方法通常使用人为筛选的上下文，比如指定为当前句的前若干句，或者对语料做一定的修改及扩充。这种上下文选择方式使得篇章上下文的获取具有一定局限性，仍然不能充分利用篇章级上下文的全局性。另一种常见的上下文选择方式则使用广为人知的软注意力(soft attention),通过注意力机制将较广范围内的候选上下文结合在一起，这种上下文结合方式具备上下文选择范围较广的优势，但同时存在计算量巨大，无关噪音较多的弱点。

本文以当前炙手可热的 Transformer 模型为基础，将两种不同的上下文选择机制结合起来拓展出基于联合选择机制的上下文编码器，以获得简洁有效的来自篇章全文的上下文信息。其中，本文使用句子向量生成全篇章的句子间依赖关系权重矩阵，并针对当前句选择与其最相关的若干句子作为当前句的篇章级上下文，这种上下文选择方式在本文中被称为硬选择机制。本文接着通过多头注意力将硬选择产生的上下文分配给源语言当前句，这种上下文选择方式在本文中被称为软选择机制。利用联合选择机制产生的上下文既来源于篇章全局，又可以在减少冗余信息的前提下使得源语言的每个单词都获得上下文中的有效信息。中-英和西-英，英-德语料上的实验表明本文提出的方法使翻译模型性能产生了有意义的提升。

---

[1] 国家自然科学基金(61876120)

[2] 国家自然科学基金(61976148)

本文主要贡献如下：

- 首次提出基于句向量依赖权重获取来自篇章全局的上下文；
- 本文模型训练过程中以篇章为单位更新模型参数，无需额外语料作为上下文；
- 本文提出一种收敛较快，结构简洁，计算开销较小的上下文利用机制。

1 研究背景

在通用神经机器翻译模型中，编码器读取由$\mathrm{x}=(x_1,x_2,\cdots,x_M)$表示的源语言句子，并通过注意力机制将其映射为连续向量$\mathrm{z}=(z_1,z_2,\cdots,z_M)$。解码器则根据给定的 z，利用注意力机制以从左到右的顺序逐个生成目标语句$\mathrm{y}=(y_1,y_2,\cdots,y_N)$中的词。当前广泛使用的 Transformer[4]通过使用由自注意力和全连接网络组成的堆栈结构实现以上机制。Transformer 在自注意力子层中使用掩码以防止当前位置参与后续位置的权重计算，通用机器翻译的目标函数如下：

$$\max_{\theta}\sum_{n=1}^{N} log\big(P_\theta(y^n|x^n)\big), \tag{1}$$

其中：θ表示模型中的参数，x^n表示第 n 个源语言语句，y^n表示第 n 个目标语言语句。

1.1 基线模型

Transformer 结构如图 1 所示，主要由编码器及解码器两部分组成，分别对应源语言编码及目标语言解码。两部分模块都由若干具有相同结构的层堆叠而成，这些模块的核心部分即多头注意力函数。

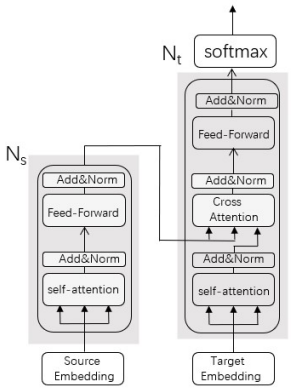

图 1 基线模型结构

Fig.1 Model Construction of Transformer

编码器

编码器通过词向量嵌入层（Word Embedding Layer）将输入的源语言编码为向量，该层通过嵌入算法将每个输入的词转换为词向量(Word Vector)。词嵌入仅发生在模型底层的编码器中，其他编码器接收到的是堆栈结构中前一层编码器的输出。由于建模过程中失去了语句原有的时序信息，Transformer 采用位置编码(Positional Encoding)给每个词向量添加一个位置编码，确定每个词的位置或者序列中不同单词之间的距离。其公式如下：

$$PE_{pos,2i}=sin\left(\frac{pos}{1000^{2i/d_{model}}}\right), \tag{2}$$

$$PE_{pos,2i+1}=cos\left(\frac{pos}{1000^{2i/d_{model}}}\right), \tag{3}$$

经过编码的源语言向量E_x通过自注意力层(Self-attention Layer)构建输入序列之间的依赖关系，捕获源语言句子中的内部结构。编码器中第 n 层的计算如下所示：

$$A^{(n)} = \text{MultiHead}\left(H^{(n-1)}, H^{(n-1)}, H^{(n-1)}\right), \tag{4}$$

其中，$A^{(n)} \in \mathbb{R}^{d_{model} \times l}$为第 n 层隐状态，$\text{H}^{(n-1)}$第 n-1 层编码器的隐状态。当 n=1 时，第一层编码器的输入$H^{(0)} = E_x$。$\text{MultiHead}(Q, K, V)$表示多头注意力函数，在自注意力层中 Q=K=V。

Transformer 在每个子层之间使用残差连接(Residual Connection)及层规范化(Layer Normalization)处理子层间传递的数据，子层实际输出如下：

$$\tilde{A}^{(n)} = \text{LayerNorm}\left(A^{(n)} + \text{H}^{(n-1)}\right), \tag{5}$$

为了便于阐述本文在之后的作图或介绍时可能省略残差及层规范化。

在经过注意力层捕获相关依赖信息后，隐藏状态向量经过全连接前馈神经网络层(Feed-Forward Layer)。其表达公式如下：

$$H^{(n)} = \left[\text{FNN}\left(\tilde{A}_{\cdot,1}^{(n)}\right); \cdots; \text{FNN}\left(\tilde{A}_{\cdot,I}^{(n)}\right)\right] \tag{6}$$

其中$H^{(n)} \in \mathbb{R}^{d_{model} \times l}$表示第 n 层编码器对源语言序列的向量表示。$\tilde{A}_{\cdot,I}^{(n)}$是第 n 子层对第 t 个词的表示。

解码器

解码器与编码器的堆栈结构类似。每一层的核心部分都有两个多头注意力层和一个前馈网络层构成。解码器的自注意力层表达如下：

$$F^{(n)} = \text{MultiHead}\left(S^{(n-1)}, S^{(n-1)}, S^{(n-1)}\right), \tag{7}$$

解码器的第二个多头注意力层为编码器-解码器注意力层(Encoder-Decoder Attention Layer), 输入该层的K，Q，V矩阵来自不同模块，其中 K，V 来自编码器的输出，而 Q 来自解码器自有的自注意力层。其表达式如下：

$$G^{(n)} = \text{MultiHead}\left(F^{(n-1)}, H^{(n-1)}, H^{(n-1)}\right), \tag{8}$$

类似的，编码器的注意力层输出也需要经过全连接前馈神经网络层(Feed-Forward Layer)。其表达式如下：

$$S^{(n)} = \left[\text{FNN}\left(G_{\cdot,1}^{(n)}\right); \cdots; \text{FNN}\left(G_{\cdot,J}^{(n)}\right)\right], \tag{9}$$

其中$S^{(n)} \in \mathbb{R}^{d_{model} \times l}$为第 n 层解码器的隐状态(n=1, ……，$N_t$)，线性层及紧随其后的归一化层(softmax)将解码器输出的浮点数向量转换为单词。

2 基于联合选择机制的篇章翻译

本文提出的上下文联合选择机制的逻辑结构如图 2 所示，候选上下文及源语言各自编码后在硬选择环节嵌入为句向量，利用句向量通过注意力机制计算获取当前句与篇章中其他所有句子的依赖关系。通过依赖权重矩阵选取与当前句最相关的若干句子并拼接为一个长句。随后将硬选择环节生成的上下文向量通过软关注(soft attention)的方式分配给当前句中的每个单词。解码器结构则与基线系统 Transformer 一致。本文将源语言编码器与上下文编码器参数共享，减小增加参数及编码器对模型性能对比造成的不公平影响。

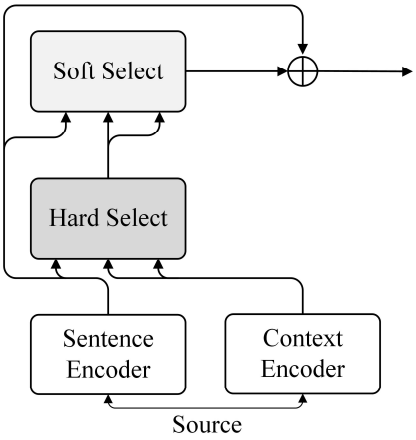

图 2 上下文联合选择机制

Fig.2 Joint Context Selection Mechanism of Model

2.1 问题定义

前文已经对通用神经机器翻译的问题做了定义。篇章级翻译中需要考虑以篇章为单位进行翻译并充分利用篇章上下文。通常，给定源文档 X，文档翻译 Y 的概率由下式给出：

$$P_\theta(Y|X) = \prod_{j=1}^{J} P_\theta\left(y^j|x^j, D_{-j}\right), \tag{10}$$

其中y^j和x^j分别表示第 j 个目标语言语句和源语言语句，而$D_{-j} = \{X_{-j}, Y_{-j}\}$是源篇章和目标篇章中与第 j 句相关的上下文集合。由于翻译模型一次翻译一个单词，因此，表达式(10)可理解为：

$$P_\theta(Y|X) = \prod_{j=1}^{J}\prod_{n=1}^{N} P_\theta\left(y_n^j|y_{<n}^j, x^j, D_{-j}\right). \tag{11}$$

其中 y_n^j是第 j 个目标端句子中的第 n 个单词，$y_{<n}^j$是先前生成的单词。

2.2 篇章句向量

受 Lin 等人[11]的启发，本文使用一个线性结合层获取句子向量。该层通过自注意力机制将整个句子所有单词产生的隐藏状态结合在一起从而生成句子向量。句中单词映射为句子向量的权重计算方法如下：

$$\alpha = \mathrm{softmax}\left(W^2 tanh\left(W^1\left(S_i^{(n)}\right)^T\right)\right) \tag{12}$$

其中W^1，W^2是模型的参数矩阵，$S_i^{(n)}$则代表第 i 个句子经过编码后输出的隐状态向量。使用前文所述编码器自注意力层输出的隐藏状态及计算出的映射权重获得篇章中的句子向量：

$$v_{X_i}^{(n)} = \sum_{j=1}^{n} \alpha_{i,j}\, s_{i,j}^{(n)} \tag{13}$$

其中$v_{X_i}^{(n)}$表示篇章中句子X_i经过句向量嵌入层后生成的句向量，$\alpha_{i,j}$表示句子X_i中各单词映射为句向量的权重，$s_{i,j}^{(n)}$表示句子X_i中的词经过编码器输出的隐藏状态，这里的句子既可能是来自上下文的句子也可能是来自源语言的句子。如图 3 所示，本文模型以篇章为单位学习更新参数，由此可以获取具有篇章结构的句向量集合。

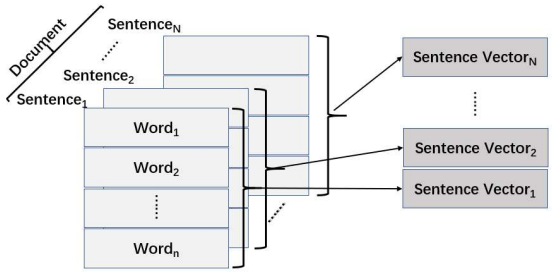

图 3 篇章句向量

Fig.3 Document Level Sentence Vector

2.3 上下文硬选择机制

本文首次提出基于句向量计算篇章内句子之间的依赖关系，并以此选取与当前句关联度最高的若干句作为当前句的上下文。

利用句向量嵌入层的输出计算当前句与篇章中所有句子间的依赖关系，公式如下：

$$u_i = \mathrm{softmax}\left(v_{X_i}^{(n)} V^{(n)} / \sqrt{d_{V^{(n)}}}\right) \tag{14}$$

其中$V^{(n)}$表示一个篇章中所有句子向量的集合。u_i表示句子X_i与篇章中所有句子的依赖权重。根据依赖权重矩阵中高权重值的分布即可得知与当前句高度关联的语句在篇章中的对应位置。将这些句子拼接成为一个长句作为当前句的篇章上下文。硬选择机制的逻辑流程如图 4 所示。

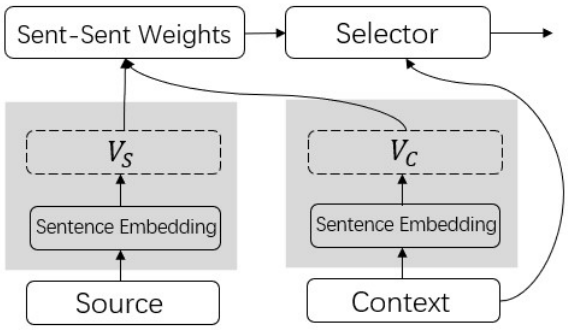

图 4 上下文硬选择机制结构

Fig.4 Construction of Hard Selection Mechanism

其中 V_s 及 V_c 分别是源端语料经过不同编码器及句向量嵌入层后生成的篇章级句子向量集合。本文模型可以自动获取语料篇章边界，所以无需额外语料作为源语言的上下文，模型利用源语言编码器输出的当前句向量与上下文编码器输出的篇章句向量集合计算当前句与篇章中其他句子之间的依赖权重。

2.4 上下文软选择机制

多头自注意力机制能够通过矩阵运算捕获同一句子内词之间或者同一篇章内不同句子之间的依赖关系，并根据依赖权重分配词或句所需要的上下文。例如图 5 所示，图中线条的粗细代表句中词对当前词的重要程度，线条越粗依赖权重越大。当编码器对子词化后的句子“He_ called_ ses ame_ maz hi_”[3]进行编码时，自注意力机制通过权重计算将“He called”以较高权重分配给未登录词“mazhi”作为后者的上下文。该机制将对相关单词的“理解”融入当前正在处理的单词中。

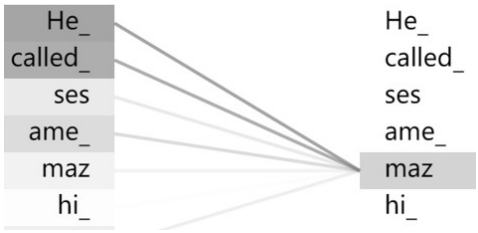

图 5 软注意力机制

Fig.5 Soft Attention Mechanism

由于硬选择机制选出的上下文句子数量固定，但并不是所有当前句中的词都需要相同长度的上下文，源语句中每个词所需上下文各不相同。因此本文采用多头注意力模块实现软选择机制，将硬选择产生的上下文按需分配给当前句中的每个词，计算过程表达公式如下：

$$O^{(n)} = \text{MultiHead}\left(S^{(n)}, C^{(n)}, C^{(n)}\right), \tag{15}$$

其中$C^{(n)}$代表硬选择生成的篇章上下文，$S^{(n)}$代表编码后的源语言，$O^{(n)}$即最终产生的篇章级全局上下文。

2.5 上下文与翻译模型结合

篇章上下文的结合方式比较多元化，常见的方式按照计算开销递增排序有：直接与源语言相加；通过一定机制动态的与源语言结合；以及在源语言编码器或目标语言解码器中增加相应的注意力子层，使得上下文与源语言/目标语言进行注意力运算等。本文出于控制训练速度以及编码器共享参数的原因，通过一个门控单元学习句子和篇章信息之间的关联，动态控制句子信息和篇章信息对解码端的影响。上下文结合过程及门控单元参数的学习过程表达式如下：

$$\lambda = \text{sigmoid}\left([S^{(n)}; O^{(n)}]W^{G}\right) \tag{16}$$

$$O_{sum} = \lambda S^{(n)} + (1-\lambda)\, O^{(n)} \tag{17}$$

其中$O^{(n)}$为联合选择机制输出的全局上下文，$S^{(n)}$为源语言编码器的输出。W^{G}表示参数矩阵，λ 经过 sigmoid 函数计算后输出数值在 0 到 1 之间，控制篇章信息的流通。最终输出的隐藏状态O_{sum}是篇章上下文与源语言的混合表达，代替原始 Transformer 中编码器的输出进入解码器，解码器结构与原始 Transformer 相同，这里不再赘述。

[1] 对应的原句为“He called sesame mazhi”。

3 实验

3.1 数据集

在中-英实验中，篇章级平行语料来自 LDC2002T01，LDC2004T07，LDC2005T06，LDC2005T10，LDC2009T02，LDC2009T15，LDC2010T03，训练集包括 4.7 万个文档中的 78 万个句子对，平均而言，整个集合中的每个文档包含 22.9 个句子。我们使用 NIST MT 2006 数据集作为开发集，并使用 MT 2002、2003、2004、2005、2008 数据集作为测试集，其中测试集的合集标记为 All。开发和测试集共包含 627 个文档，5,833 个句子。平均每个文档包含 9.9 个句子。本文使用 Jieba[4]分词将汉语句子按词切分，而英语句子则使用 Moses 脚本[12]进行分词和小写处理。我们通过字节对编码(BPE)[13]使用 3 万大小的词表分别将源语言和目标语言中的单词进一步分割成子词

西-英翻译任务中的训练集为 IWSLT 2014 和 2015[14]，开发集为 dev2010，测试集为 tst2010、tst2011 和 tst2012。英-德翻译任务中的训练集来自 IWSLT2017，本文使用 tst2016 和 tst2017 作为测试集，余下数据集作为开发集。所有数据集均使用 Moses 脚本进行分词和 Truecase 处理。并使用 3 万大小的联合词表将源端及目标端语料中的单词分割成子词。我们将长篇章切分为最大长度为 30 句的段落。实验数据集的篇章数，句子数及平均篇章长度等统计信息如表 1 所示。

表 1 数据集统计信息

Tab.1 Data Set

Set	**ZH-EN**		**ES-EN**		**EN-DE**	
	#SubDoc	#Sent	#SubDoc	#Sent	#SubDoc	#Sent
Training	47,758	781,524	6,531	180,853	7,491	206,126
Dev	82	1,664	33	887	326	8,967
Test	627	5,833	165	4,706	87	2,271

3.2 实验设置

本文基于 OpneNMT [15]实现以平行句对为单位更新参数的基准模型 Transformer，并进一步拓展为以篇章为单位更新参数的翻译模型。以篇章为单位更新使得模型可以自动感知语料的篇章边界，从而进一步获取全局上下文。本文借鉴 Zhang 等[19]的思想将模型训练分为两个阶段，使用的训练方法类似在神经机器翻译中广泛使用的预训练[20]。区别在于 Zhang 等人[19]训练方法在第二步训练时固定部分参数以防止模型在相对较小的篇章级别平行语料上过度拟合。而本文第二步训练着重于微调(finetune)。

本文将词向量隐藏状态维度设置为 512，训练批量大小(batch_size)为 8192，为多头注意力机制设置 8 个独立的头，设编码器层数$N_s = N_t = 6$，上下文编码器层数N_c则设置为 1。在训练中使用自适应的 Adam 算法[17]进行调整学习，使用 Vaswani 等[4]描述的学习率衰减策略使学习率自动变化。在解码过程中，柱形搜索(beam search)的窗口大小设置为 5，使用长度惩罚策略[32]并将超参数α设置为 0.6。在第二步训练时适当降低 learning rate。

所有实验皆使用单块显存 32G 的 Nvidia V100 显卡进行训练。

[4] https://github.com/fxsjy/jieba

3.3 评估指标

对于中-英翻译任务，本文报告了使用 multi-bleu.perl 脚本计算的不区分大小写的 BLEU 得分[16]。对于其他翻译任务，本文报告了根据 multi-bleu.perl 脚本计算的区分大小写的 BLEU 分值和 Meteor 得分[17]。以上数据集和评估方法与本文比较实验的设置是一致的。我们使用 paired bootstrap[18]重采样方法评测 BLEU 值提升的显著性。

3.4 实验结果

为了客观对比，本文选择基于 Transformer 并可获取开源代码的篇章翻译模型作为对比试验对象。并在训练过程中使用相同的训练设置。

表 2 列出了本文在中-英语料上进行的实验及其对比，其中硬选择上下文的长度为 3 句。†和‡表示与 Transformer 基准模型相比显著性 p 值小于 0.05/0.01。可以看出，本文提出使用联合不同类型上下文选择机制进行翻译的结果好于仅使用句子前两句作为上下文的翻译结果。实验结果同时表明联合软硬两种选择机制产生的篇章级上下文对翻译模型性能提升幅度比仅使用硬选择机制产生的上下文要大。

表 2: 本文模型中-英翻译任务的性能(BLEU).

Tab.2 Chinese-English

模型	**MT06**	**MT02**	**MT03**	**MT04**	**MT05**	**MT08**	**All**
Transformer	36.27	42.71	43.51	41.25	41.07	31.54	39.64
+ 硬选择上下文	37.08‡	43.49†	44.39‡	42.38‡	41.73†	32.55‡	40.59‡
+ 联合选择上下文	**37.50‡**	**43.87‡**	**44.95‡**	**42.81‡**	**41.90‡**	**32.64‡**	**40.90‡**
Transformer(Zhang et al., 2018)	36.20	42.41	43.12	41.02	40.93	31.49	39.53
Transformer-DocNMT(Zhang et al., 2018)	37.12	43.29	43.70	41.42	41.84	32.36	40.22

表 3 列举了本文模型在西-英及英-德两个篇章级翻译任务上的 BLEU 和 Meteor 得分，硬选择上下文的长度仍然为 3 句。与中-英任务相似的是，在这两个翻译任务中本文所提出使用联合不同类型上下文选择机制进行翻译的结果好于仅使用句子前两句作为上下文的翻译结果。此外，利用联合机制选择上下文比只使用硬选择机制产生的拼接向量作为上下文更有帮助。在两个翻译任务上，我们的方法相比 Transformer 基线模型在 BLEU(Meteor)评测标准上提了 2.07(2.01)和 1.71(1.86)。

表 3:西班牙-英及英语-德语翻译性能(BLEU 和 Meteor)

Table3. Spanish-English and English-Germany

模型	西-英		英-德	
	BLEU	Meteor	BLEU	Meteor
Transformer	35.50	34.60	23.02	43.66
+ 硬选择上下文	37.27	36.29	24.13	44.96
+ 联合选择上下文	**37.57**	**36.61**	**24.73**	**45.52**
Transformer-DocNMT(Zhang et al., 2018)	37.07	36.16	24.00	44.69
HAN-DocNMT(Miculicich et al., 2018)	37.35	36.50	24.58	45.48

4 分析与讨论

4.1 不同长度上下文的影响

本文提出的硬选择机制可以在整个篇章中筛选出与当前语句最相关的语句。出于观察上下文利用情况，我们分别使用不同长度句数的硬选择上下文，并通过软选择机制分配给当前句中的每个词。该实验表明对篇章翻译而言，使用联合选择方式选出的上下文句子数量并不是越多越好。从表 4 中可以观察到上下文句数为 3 句时翻译性能取得了最佳效果，当上下文长度进一步增加时性能出现了整体下滑，我们认为与当前语句相关性较低的语句所含有效上下文较少，过长的硬选择上下文可能会造成信息冗余。

表 4 不同长度篇章上下文

Tab.4 Different Lengths of Document-level Context

测试集	2 句	3 句	5 句
NIST2006	35.72	**37.50**	37.17
NIST2002	41.75	**43.87**	43.04
NIST2003	42.24	**44.95**	44.41
NIST2004	**42.86**	42.81	40.45
NIST2005	40.07	**41.90**	41.69
NIST2008	30.99	32.64	**32.79**
All	39.80	**40.90**	40.71

4.2 上下文结合方式的影响

表格 5 列出了不同上下文结合方式造成的翻译性能差异。正如本文 2.5 节中所述，将上下文与翻译模型结合的方法有多种。本文在使用 3 个句子作为硬选择机制上下文长度的实验中对比了表中所列几种上下文结合方式。可以观察到使用门控单元动态混合上下文与源语言编码器输出内容比将两者直接相加的效果更好。而在编码器侧增加单独的注意力子层除了增加计算开销及训练时间外并没有获得性能显著提升。

表 5 不同上下文结合方式的影响

Tab.5 Effects of the combination of different contexts

结合方式	**BLEU**
直接相加	40.64
门控单元	40.90
注意力函数	**40.97**

4.3 上下文在篇章中的分布

本文通过导出硬选择机制中的句子依赖权重矩阵获取当前句上下文(最相关句)与当前句的距离，统计篇章中上下文的分布情况。表中第一列的数值代表上下文与当前句的距离。如表 6 所示，有 37.31%的句子与其最先关的上下文与其距离不超过 3 句，如果我们把这个范围扩大到 5 句这一比例可上升到 51.8%。由此可见上下文分布具有聚集性，大部分上下文聚集在当前句附近。但值得注意的是，仍有一部分上下文分布在距离当前句较远的位置，这一观察结果可能随语种及实验语料的更换而发生变化，但仍可以从侧面表明本文提出将上下文选择范围扩大到篇章级别的必要性。

表 6 上下文分布情况

Tab.6 Distribution of Document-level Context

上下文分布	**百分比**
3 句以内	37.31
3-5 句	14.49
5 句以外	**48.20**

4.4 实例分析

本文通过测试集翻译结果的样例观察联合选择机制对翻译质量的改进。如表 7 中的样例可见，使用联合选择机制生成的篇章级上下文可以改善篇章翻译中的代词翻译质量。进

一步的对比可以发现，使用联合选择机制的模型在句子意义表达及流畅度方面比仅使用硬选择机制上下文的模型要好。

表 7 实例分析

Tab.7 Case Study

源 语 言	. . . . . 今天晚上的十一二点钟左右吧。
目标语言	. . . . . **it** will arrive around 11 : 00 or 12 : 00 tonight.
Transformer	. . . . . **that** about 11.2 pm today.
硬选择机制	**it** will be around 11.2 pm today .
联合选择机制	**it** will arrive around 12 : 00 tonight ..

5 相关研究

研究者们[5-8]在使用篇章信息提高统计翻译质量的研究领域已经做了大量工作。机器翻译研究热点从统计翻译转向神经翻译后不久篇章级神经机器翻译的研究也蓬勃发展起来。根据获取上下文的范围，我们将相关研究分为两类:(1)使用篇章中部分语句作为上下文来源; (2)使用篇章作为上下文来源。

在第一类研究中，Tiedemann 等人[21]提出的直接拼接语句作为上下文的方法是基于循环神经网络(RNN)的早期尝试。随后 Jean [22]；Wang[7]；Zhang[19]；Bawden[9]；voita[23]等人的研究在 RNNSearch 和 transformer 中使用具有不同注意力机制的多编码器。Miculicich 等人[24]提出一种分层注意网络(HAN)，它通过两层抽象表示为当前句从前面的句子中提取上下文。Yang 等人[25]在 HAN 的基础上提出一种将上下文信息进行聚类的胶囊网络。Tu[8]；Kuang [26]等人提出基于缓存的方法利用前面句子中的词和翻译。李京谕[33]等人提出利用强化学习选取上下文的方法由于限制了文档长度最多为 7 句且需要额外上下文语料，也属于这一类方法。同时该模型由于优化目标不同，需要将强化学习模型及翻译模型反复冻结交叉训练，训练过程较为繁琐。

另一类篇章级神经机器翻译将篇章作为翻译单元，针对篇章中的每个句子动态获取有用的篇章级信息。Maruf 等人[27]使用额外的存储网络将篇章转换为上下文与基于 RNN 的神经机器翻译模型结合。Mace 等人[28]在每个源句中添加篇章标签，并将其替换为篇章级嵌入向量。Xiong 等人[29]提出了一种二次优化策略，通过激励机制来完善第一轮翻译以优化整个文本的连贯性。受 HAN 模型的启发，Maruf 等人[30]提出了一种层次注意力，使用稀疏注意力机制选择性地捕获与当前句相关联的句子，然后进一步选择关键词。Tan 等人[31]提出层次网络获取上下文向量，并将其分配给当前句中的词。这些研究经常伴随着计算开销大，训练时间长以及获取上下文中含有较多冗余的情况。

与上述研究不同的是，本文提出联合软硬两种选择机制对来自整个篇章的全局上下文进行建模。该模型基于句子向量计算篇章内句子间的依赖关系，避免了对全篇章进行词级计算的巨大开销及信息冗余。利用注意力机制分配硬选择机制产生的上下文，使当前句的每一个词按需分配到高质有效的来自整个篇章的上下文。同时，因为本文模型以篇章为单位更新模型参数，可以自然的感知篇章语料边界，在训练过程不需要使用额外语料作为上下文。由于模型各部分训练目标一致，该模型在训练过程收敛较快，训练过程简便。

6 总结与展望

本文提出一种联合软硬上下文选择机制生成篇章级上下文用以提升篇章神经机器翻译性能的方法。该模型将篇章信息获取范围扩大至源语句周边乃至整个篇章文档，一定程度上缓解了以往仅使用源语句相邻语句作为上下文导致不能较好获取篇章信息的问题。该模型中的硬选择机制规避了以往研究工作中单纯使用软注意力方法可能造成的信息冗余及计

算开销过大的问题。在此基础上本文使用软选择机制将硬选择方法选出的上下文与翻译模型有效结合，实验证明本文提出的联合选择机制帮助翻译模型在中-英，西-英，英-德等语料上取得了有意义的性能提升。同时进一步探讨了上下文长度对篇章翻译质量的影响。本文通过导出硬选择上下文语句的位置信息观察当前语句上下文在文档中的分布情况。最后，本文还通过样例分析实际观察本文提出的模型对翻译模型的提升，样例表明联合选择机制不仅产生了有意义的 BLEU 值提升，还有效改善了翻译流畅度和准确性及代词翻译质量。我们将在以后的工作中在约束计算开销增加幅度的前提下进一步探索其他可以更好获取有效篇章信息的方法，例如从语义相似度等其他角度考虑上下文的选择方式。

参考文献：

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Proceedings of NIPS. 2014: 3104-3112.

[2] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of ICLR. 2015.

[3] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 1243-1252.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Proceedings of NIPS. 2017: 5998-6008.

[5] Zhengxian Gong, Min Zhang, and Guodong Zhou. Cache-based document-level statistical machine translation. In Proceedings of EMNLP. 2011: 909-919.

[6] Christian Hardmeier, Joakim Nivre, and J¨org Tiedemann. Document-wide decoding for phrase-based statistical machine translation. In Proceedings of EMNLP. 2012: 1179-1190

[7] Longyue Wang, Zhaopeng Tu, Andy Way, and Liu Qun. Exploiting cross-sentence context for neural machine translation. In Proceedings of EMNLP. 2017: 2826-2831

[8] Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. Learning to remember translation history with a continuous cache. In Proceedings of TACL. 2018:407-420.

[9] Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. Evaluating discourse phenomena in neural machine translation. In Proceedings of NAACL. 2018:1304-1313.

[10] Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. Does neural machine translation benefit from larger context? 2017.

[11] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos x Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In Proceedings of ICLR.

[12] Koehn, Philipp and Hoang, Hieu and Birch, Alexandra and Callison-Burch, Chris and Federico, Marcello and Bertoldi, Nicola and Cowan, Brooke and Shen, Wade and Moran, Christine and Zens, Richard and Dyer, Chris and Bojar, Ond˘rej and Constantin, Alexandra and Herbst, Evan. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. Proceedings of ACL, (Jun):177–180.

[13] Rico Sennrich and Barry Haddow and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of ACL, 1715–1725.

[14] Mauro Cettolo and Christian Girardi and Marcello Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. Proceedings of EAMT, 261–268.

[15] Klein, Guillaume and Kim, Yoon and Deng, Yuntian and Senellart, Jean and Rush, Alexander. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. Proceedings of ACL, 67–72.

[16] Kishore Papineni and Salim Roukos and Ward Todd and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. Proceedings of ACL, 311–318.
[17] Lavie, Alon and Agarwal, Abhaya. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. Proceedings of WMT, (Jun):228–231.
[18] Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. Proceedings of EMNLP, 388–395.
[19] Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. Improving the transformer translation model with document-level context. In Proceedings of EMNLP. 2018: 533–542.
[20] Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. In Proceedings of ACL. 2016: 1683-1692.
[21] Tiedemann, Jo¨rg and Scherrer, Yves. 2017. Neural Machine Translation with Extended Context. In Proceedings of the Third Workshop on Discourse in Machine Translation, ”82–92.
[22] Sebastien Jean and Stanislas Lauly and Orhan Firat and Kyunghyun Cho. 2017. Does neural machinetranslation benefit from larger context?. In Computing Research Repository, arXiv:1704.05135.
[23] Elena Voita and Rico Sennrich and Ivan Titov. 2019. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion. Proceedings of ACL, 1198– 1212.
[24] Lesly Miculicich and Dhananjay Ram and Nikolaos Pappas and James Henderson. 2018. Document-Level Neural Machine Translation with Hierarchical Attention Networks. Proceedings of EMNLP, 2947–2954.
[25] Zhengxin Yang and Jinchao Zhang and Fandong Meng and Shuhao Gu and Yang Feng and Jie Zhou. 2019. Enhancing Context Modeling with a Query-Guided Capsule Network for Document-level Translation. Proceedings of EMNLP, 1527–1537.
[26] Shaohui Kuang and Deyi Xiong and Weihua Luo and Guodong Zhou. 2018. Modeling Coherence for Neural Machine Translation with Dynamic and Topic Caches. Proceedings of COLING, 596—606.
[27] Sameen Maruf and Gholamreza Haffari. 2018. Document Context Neural Machine Translation with Memory Networks. Proceedings of ACL, 1275–1284.
[28] Valentin Mace and Christophe Servan. 2019. Using whole document context in neural machine translation. In Proceedings of IWSLT.
[29] Hao Xiong and Zhongjun He and Hua Wu and Haifeng Wang. 2019. Modeling coherence for discourse neural machine translation. In Proceedings of AAAI, 7338–7345.
[30] Sameen Maruf and Andre´ F. T. Martins and Gholamreza Haffari. 2019. Selective Attention for Context-aware Neural Machine Translation. Proceedings of NAACL, 3092–3102.
[31] Xin Tan and Longyin Zhang and Deyi Xiong and Guodong Zhou. 2019. Hierarchical Modeling of Global Context for Document-Level Neural Machine Translation. In Proceedings of EMNLP-IJCNLP, 1576–1585.
[32] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. 2016.
[33] 李京谕,冯洋. 基于联合注意力机制的篇章级机器翻译[J]. 中文信息学报, 2019, 33(12): 45-53.

Integrating Document-level Context into Neural Machine Translation via Joint Selection Mechanism

Abstract: Neural machine translation has become a very attractive branch in the field of machine translation, in which document translation has attracted the attention of researchers. A considerable portion of the previous research has focused on the sentence-level neural machine translation. To improve translation performance, the sentence level context is obtained in different ways and combined with the machine translation model. However, most of the research work using text-level context obtains the context from the entire text corpus, which may result in excessive computation and information redundancy. In this paper, we propose the use of a joint selection mechanism, the use of a hard selection mechanism to obtain a finite length of context within the whole text, and then the use of a soft selection mechanism to obtain the global context for the current sentence. Experiments have shown that this method can improve the performance of the translation model. This paper further analyzes the distribution of text level context around the current sentence and observes some meaningful phenomena.
Key words: nural machine translation; document-level context; joint selection mechanism