文章编号：1003-0077(2020)04-0077-08

面向短文本理解的省略恢复研究

郑 杰，孔 芳，周国栋

（苏州大学 计算机科学与技术学院，江苏 苏州 215006）

摘 要：省略作为一种普遍存在的语言现象，在中文文本尤其是对话、问答等短文本中频繁出现。该文从服务于短文本理解的视角出发，针对省略恢复问题提出了一种多重注意力融合的省略恢复模型。该模型融合交叉注意力机制和自注意力机制，借助门控机制将上下文信息与当前文本信息进行有效结合。在短文本问答语料上的多组实验结果表明，该文给出的模型能有效地识别并恢复短文本中的省略，从而更好地服务于短文本的理解。

关键词：省略；短文本；注意力

中图分类号：TP391 **文献标识码**：A

A Study of Ellipsis Recovery for Short Text Comprehension

ZHENG Jie, KONG Fang, ZHOU Guodong

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

Abstract: As a common linguistic phenomenon, ellipsis is common in texts, especially in short texts such as QA and dialogue. In order to understand the semantic information of short texts, we propose a multi-attention fusion model for Chinese ellipsis recovery. This model combines the context and the text information by gate mechanism, multi-attention and self-attention. Experiments on several short text corpora show that this model can efficiently detect ellipsis position and recover ellipsis content, facilitating better comprehension of short text.

Keywords: ellipsis; short text; attention

0 引言

省略——作为一种语言现象，在自然语言中普遍存在。根据 Kim[1] 统计，英文中大约 96% 为显式主语，而中文里显式主语仅占 64%。由此可见，相较于英文，中文里的省略现象更为频繁。此外，在短文本应用场景，如对话系统中，对话双方有相同的对话背景，对话的多个轮次共享了一些信息，因此在对话里省略现象更为常见。类似地，系列问答系统中，连续的多个问题间存在关联关系，后续问题与前面问题的答案间也存在一定的关联关系，它们之间都可能共享一些信息，这些共享信息在后续陈述中都可能会被省略。显然，对于短文本的理解，省略的识别以及恢复发挥着重要作用。

1 相关工作

面向短文本的省略恢复研究是一个很有潜力的领域，受限于语料，目前相关研究很少。代表性工作包括：Huang 等[2] 对中文口语对话系统中的省略现象进行研究，提出了一种基于主题结构的省略恢复方法。但他们提出的方法是面向特定领域的，不具有通用性，而且该方法只是停留在理论层，并没有给出具体的可计算模型。Ren 等[3] 从依存关系的角度定义了省略类型。尹庆宇等[4] 针对中文省略中特殊的零代词构建了恢复和消解的框架，并采用联合模型减少管道模型带来的误差传播，但其服务对象并不是短文本。Kumar 等[5] 针对英文短句使用序列到序列模型来融合前文信息，并对当前短句进行重

收稿日期：2019-08-22 **定稿日期**：2019-11-12

基金项目：国家自然科学基金(61876118,61751206)

写,对短句进行语义信息补充和指代消解,但这种重写不能完全还原被重新写语句的信息。和 Kumar 等人类似,郑杰等[6]在单轮中文短文本中通过使用序列到序列的方式对中文短句进行重写,从而达到对短句进行省略恢复的目的。同样地,该方法不能保存原语句的结构信息,生成结果容易产生语义偏离。

目前,面向对话领域的省略恢复研究出现了新进展。Su 等[7]使用 Transformer 并结合指针网络来处理中文多轮对话的省略和指代现象。Quan 等[8]针对特定领域的多轮对话构建了一个端到端的省略恢复和指代消解框架。同样地,上面两种方法都是使用序列到序列的模型结构,都会带来语义偏离的问题。Yang 等[9]针对多轮对话中的频繁出现的零代词,构建了一种序列标注模型。但是该方法将省略内容转化为标签,无法适用真实对话场景下省略复杂多样的情况,并且序列标注的方法无法处理多个省略内容的情况。Yin 等[10]使用基于注意力的零指代方法来处理对话中省略内容的消解步骤,却没有考虑省略位置的识别,并且这种管道结构会带来传播误差。

综上,面向中文单轮对话场景下的省略现象,本文在郑杰等人的单轮对话数据集上,提出了一种融合多重注意力机制并且综合省略位置识别和消解的省略恢复联合模型。相比郑杰等人的序列到序列模型,有如下创新点:

(1) 增加字符级嵌入,帮助缓解因为词表维度过高带来的语义稀疏的问题。

(2) 在模型编码层加入多头自注意力机制提取短文本特征,并融合前文和当前文本的注意力信息来对短对话进行建模。

(3) 在郑杰等人提出的序列到序列解码方式基础上进行改进,模型假设省略序列中相邻词之间都是省略候选位置,针对每一候选位置采用文本生成的方式来填充省略内容。省略内容为空的位置不存在省略。这种策略不会丢失文本结构,并且能够针对省略补充后的文本构建语言模型,学习文本省略模式。

2 任务定义

我们将每一个训练实例表示为(H,U,R)。U 表示需要恢复的短句,H 表示 U 可能存在的上文,R 表示省略恢复后的结果。省略恢复任务的目标是学习一种映射关系 f:H,U→R,利用这种关系,我们可以借助 U 和 U 的上文 H 来对 U 进行省略内容的填充。下一节中,我们会对模型进行详细的介绍。

3 融合多注意力机制的基于编码解码框架的省略恢复模型

3.1 嵌入层

嵌入层(embedding)的主要功能是将离散的词语单元映射到低维语义空间,用几十到几千不等维度的向量进行表征,从而让模型理解词语的语义信息。这里,为了综合考虑词信息以及字级别信息,本文采用了词向量和字符向量相结合的方法。

3.1.1 词级嵌入

词级别嵌入(word embedding),是以语料经过分词后形成的词典为基础,构建一个词嵌入矩阵 $D_w \in R^{v_w \times f_w}$,其中 v_w 表示词典长度,f_w 表示词向量的维度。在实验中,本文没有使用预训练的词向量,采用的是最小值$-\sqrt{3}$、最大值$+\sqrt{3}$之间的正态分布随机数来初始化嵌入矩阵,参数随模型一起训练。

3.1.2 字符级嵌入

字符级嵌入(character embedding),是将所有词语分解为字,构建一个字嵌入矩阵 $D_c \in R^{v_c \times f_c}$,其中 v_c 表示字典长度,f_c 表示字向量的维度。使用字符级嵌入的原因,是我们在对语料进行统计时发现,很多词语并没有真正分开,例如,“感谢有你”就被当作一个词,只使用词嵌入无法很好地表征词语含义,而将词语拆成字级别表征后可以获得词语内细粒度的语义信息。

本文使用的字符级嵌入,是最小值为$-\sqrt{3}$,最大值$+\sqrt{3}$之间的正态分布随机数来初始化嵌入矩阵。在实验中,每个词语首先都会被填充到最大字长度,然后通过字嵌入矩阵映射为字向量集合。为了去除填充内容对词语语义信息的干扰,对每个词语的字集合加上了 mask,即将有效字置为 1,填充字置为 0。和词嵌入矩阵一样,字嵌入矩阵参数随模型一起训练,最后将词向量和字向量拼接后的向量作为词语的表征。字符级嵌入的原理如图 1 所示。

3.2 编码层

编码层(encoder)通过子模块对嵌入层得到的

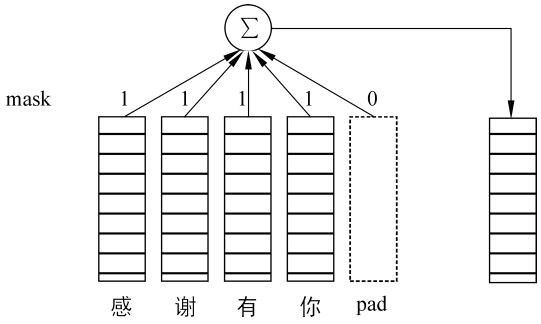

图 1　字符级嵌入原理

词语向量进一步提取特征，并融合各类特征作为词语新的语义表征。编码层主要分为三个子模块，包括 Bi-LSTM、多头自注意力机制以及门控机制。编码层的结构如图 2 所示。

值得注意的是，编码层对原省略序列和它对应的上下文序列都进行了编码。由于本文的语料是由带省略的一轮问答对组成，因此若待恢复序列为回答序列，那么其上下文为问句序列；若待恢复序列为问句序列，那么其上下文为空。

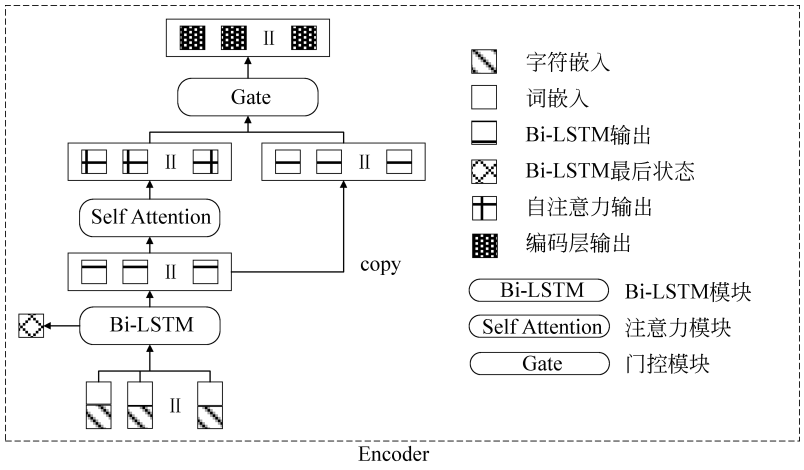

图 2　编码层结构图

3.2.1　Bi-LSTM

Bi-LSTM 是一种经典并且有效的特征提取模块。首先，借助 LSTM[11] 中的三种门机制来有效解决传统 RNN 神经网络对序列编码时出现的梯度消失和梯度爆炸的问题。另外，双向编码的结构[12] 可以有效捕获序列的历史和未来信息。

我们记 x_{elp} 为需要省略恢复的词索引序列，x_{ctx} 为其上下文词索引序列。v_{elp} 和 v_{ctx} 分别表示将 x_{elp} 和 x_{ctx} 通过嵌入层表示为词向量序列。之后，通过双向 LSTM 编码，如式(1)、式(2)所示。

$$h_{\text{elp}}^{\text{enc}}, c_{\text{elp}}^{\text{enc}} = \overleftrightarrow{\text{BiLSTM}}(v_{\text{elp}}) \tag{1}$$

$$h_{\text{ctx}}^{\text{enc}}, c_{\text{ctx}}^{\text{enc}} = \overleftrightarrow{\text{BiLSTM}}(v_{\text{ctx}}) \tag{2}$$

其中，v_{elp} 和 v_{ctx} 被编码为融入序列时序信息的输出向量 $h_{\text{elp}}^{\text{enc}}$ 和 $h_{\text{ctx}}^{\text{enc}}$。值得注意的是，我们使用同一种编码层权重参数来对当前待恢复序列以及上下文序列编码。

3.2.2　多头自注意力

自注意力[13]（self-attention）来自 Google 机器翻译团队 2017 年提出的模型 Transformer。考虑到中文短文本句长短，表达方式不规范，所以本文借鉴多头自注意力机制来从不同角度、不同层次提取更多文本自身特征，来更好地帮助模型理解文本和进行省略恢复。

多头自注意力机制主要分为两个部分：放缩点积注意力机制和多头机制。放缩点积注意力机制，主要是提取文本中词语之间的相似度信息，学习句子内部的词依赖关系，捕获句子中的内部结构。它的计算如式(3)所示。

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{\text{T}}}{\sqrt{d_K}}\right)V \tag{3}$$

其中 $Q,K,V \in R^{\text{batch}\times n\times dK}$。$d_K$ 是 Q,K,V 的最后一个维度，$\sqrt{d_K}$ 起到了调节作用，控制 Q,K 的内积不会太大。

另一个重要的内容是多头机制。考虑到只使用单一的放缩点积注意力，特别是在不规范中文短文本中，不能够从多角度、多层面捕获到重要的特征，所以本文使用了多头注意力机制。

多头注意力机制在参数不共享的前提下将 Q,K,V 通过参数矩阵映射后再做放缩点积注意力，并将这个过程分别做 h 次，最后将结果进行拼接，从而获得较全面的特征信息。它的计算如式(4)、式(5)所示。

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_i, \cdots, \text{head}_h) \quad (5)$$

在上下文序列自注意力计算中，Q,K,V 都是双向 LSTM 编码结果 $h_{\text{ctx}}^{\text{enc}}$。在省略序列自注意力计算中，$Q,K,V$ 都是双向 LSTM 编码结果 $h_{\text{elp}}^{\text{enc}}$。

3.2.2 编码门控

门控(gate)单元旨在控制多路信息的流通。本文为了减少人工干预，让模型自适应学习调控，在编码层和解码层都增加了门控机制。这里主要介绍编码层中门控机制的使用原理。

在编码层中，对于融合了时序信息的 Bi-LSTM 输出结果和蕴含词语间依赖关系的自注意力计算结果，只考虑一种或者简单相加都是不合理的。因此，我们在编码层输出结果上增加 Sigmoid 门控单元，让模型自适应调节权重比例。门控单元的计算原理如式(6)、式(7)所示。

$$\text{gate}_{\text{ctx}} = \text{sigmoid}(w_{\text{ctx}} h_{\text{ctx}}^{\text{enc}} + b_{\text{ctx}}) \quad (6)$$

$$\text{gate}_{\text{elp}} = \text{sigmoid}(w_{\text{elp}} h_{\text{elp}}^{\text{enc}} + b_{\text{elp}}) \quad (7)$$

其中，$h_{\text{ctx}}^{\text{enc}}, h_{\text{elp}}^{\text{enc}}$ 分别表示上下文以及待恢复序列的 Bi-LSTM 输出。$w_{\text{ctx}}, b_{\text{ctx}}$ 和 $w_{\text{elp}}, b_{\text{elp}}$ 分别表示上下文序列和待恢复序列的权重和偏置。最后，编码层最终输出结果由门控单元自适应控制，计算原理如式(8)、式(9)所示。

$$h_{\text{ctx-self}}^{\text{enc}} = \text{gate}_{\text{ctx}} h_{\text{ctx}}^{\text{enc}} + (1 - \text{gate}_{\text{ctx}}) \text{MultiHead}_{\text{ctx}} \quad (8)$$

$$h_{\text{elp-self}}^{\text{enc}} = \text{gate}_{\text{elp}} h_{\text{elp}}^{\text{enc}} + (1 - \text{gate}_{\text{elp}}) \text{MultiHead}_{\text{elp}} \quad (9)$$

其中，$\text{MultiHead}_{\text{ctx}}$ 和 $\text{MultiHead}_{\text{elp}}$ 分别代表上下文序列以及待恢复序列的多头自注意力机制计算结果。

3.3 解码层

解码层(decoder)通过接收编码层提取的特征信息，并采取一种策略来输出预测的结果。解码层也分为三个模块，分别是交叉注意力模块、门控模块以及解码策略模块。其中，交叉注意力模块和门控模块起到连接编码层和解码层的作用，它们将编码层提取的特征信息以交叉注意力的形式，借助门控的融合方式传输给解码层，原理如图 3 所示。

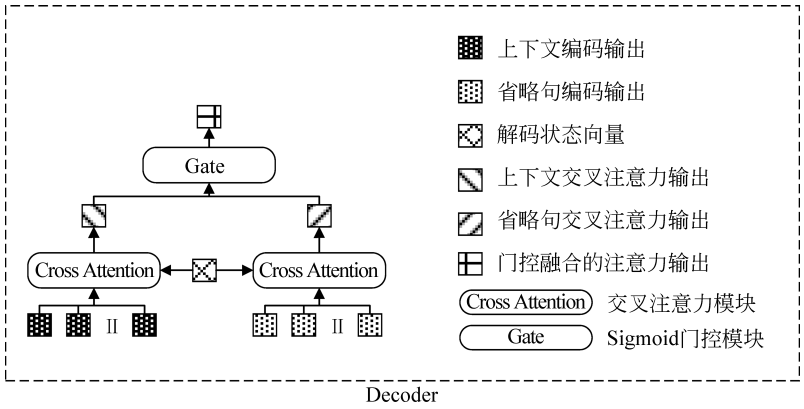

图 3 交叉注意力和门控模块原理图

3.3.1 交叉注意力

交叉注意力[14](Bahdanau attention)最早由 Bahdanau 提出并将其使用在神经机器翻译任务中，本质上起到对齐的作用，它将翻译中的目标语句与源语句中的词进行对齐，进而大大提升了翻译质量。在本文中，为了能够在解码端动态参照编码端序列信息，实现“软对齐”，我们引入了交叉注意力。

这里，由于编码层分别对上下文序列以及待恢复序列编码，所以我们分别对两种序列引入交叉注意力机制。记编码端第 i 个输出向量为 h_i^{enc}，解码层第 j 个时刻下的状态为 c_j^{dec}，V, W_h, W_c 分别是权重矩阵参数。交叉注意力的计算如式(10)～式(12)所示。

$$e_{ij} = V\tanh(W_h h_i^{\text{enc}} + W_c c_j^{\text{dec}} + b) \quad (10)$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{kj})} \quad (11)$$

$$cv_j = \sum_i a_{ij} h_i^{\text{enc}} \quad (12)$$

这样，每一时刻 j 下的解码状态 c_j^{dec} 分别与待恢复序列 $h_{\text{elp}}^{\text{enc}}$ 和上下文序列 $h_{\text{ctx}}^{\text{enc}}$ 计算交叉注意力，将各自得到的向量 cv_j 作为序列间关联信息融入到解码端，实现编码端与解码端之间的语义对齐。

3.3.2 解码门控

值得注意的是，不同于编码层的门控，解码层所使用的门控是用来自适应融合上下文特征以及待恢复序列特征。这是由于编码层对上下文序列以及待

恢复序列都进行了编码，并且模型在解码过程中对两者信息的需求程度是不同的。为了模拟这个特点，我们在解码层也增加了门控机制，具体计算原理如式(13)、式(14)所示。

$$\mathrm{gate}_{c,e}=\mathrm{sigmoid}(W_{c,e}(c_{\mathrm{ctx}}^{\mathrm{enc}}+c_{\mathrm{elp}}^{\mathrm{enc}})+b_{c,e}) \quad (13)$$

$$cv_{c,e}=\mathrm{gate}_{c,e}cv_{\mathrm{ctx}}+(1-\mathrm{gate}_{c,e})cv_{\mathrm{elp}} \quad (14)$$

其中，$\mathrm{gate}_{c,e}$表示调节上下文信息和待恢复序列信息权重的门控单元，$c_{\mathrm{ctx}}^{\mathrm{enc}}$和$c_{\mathrm{elp}}^{\mathrm{enc}}$分别表示经过编码层编码的上下文状态向量和待恢复序列的状态向量。cv_{ctx}和cv_{elp}分别表示经过交叉注意力计算的上下文和待恢复序列结果，$cv_{c,e}$则表示模型自适应得到的上下文和待恢复序列的融合信息表征。

3.3.3　解码策略

省略恢复常见的思路是使用 Pipeline 结构，即先进行省略位置的识别，再从识别出省略的位置进行省略内容的填充。这种方法会带来误差传播，即整体模型的性能受限于各个子模型的性能。所以，本文采用一种联合的方法。我们不去检测省略出现的位置，而是假设句子内相邻词之间都可能存在省略，我们在每一对相邻词之间都进行省略内容的预测，具体过程如图 4 所示，其中浅色字体为省略内容。

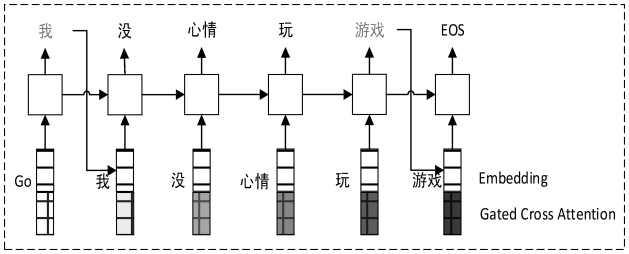

图 4　解码策略

假设有待恢复序列为{“Go”，“没”，“心情”，“玩”}，首先解码层接受句子开始标志“Go”作为第一个时序的词向量，并和第一个时序解码状态得到的交叉注意力结果$CV_{c,e}$拼接作为解码层输入，经过解码层计算后从词表选择概率最大的词“我”，但是“我”并不是待恢复序列下一个词“没”，所以解码层将“我”作为下一个时序的词向量。在第二个时序，解码层接受“我”并和交叉注意力拼接作为输入，经过计算输出词“没”，“没”是待恢复序列的下一个词，所以“Go”和“没”之间的省略内容为“我”。在第三个时序，解码层接受“没”作为输入，输出词“心情”，但是“心情”是待恢复序列下一个词，所以“没”和“心情”之间没有省略。以此类推直到输出句子结束标志“EOS”，最终经过模型省略恢复后的序列为{“我”，“没”，“心情”，“玩”，“游戏”}。

最终，模型会将解码过程中预测的词保存，作为模型预测的省略内容输出。若相邻词之间为空，则表明模型预测这两个词间不存在省略；若相邻词不为空，则相邻词之间的预测内容就是模型预测的省略内容。还有一点值得注意，为了增强模型的鲁棒性，防止模型进入无限预测的死循环中，我们在实验中设置了最长连续解码步长为 8，即最长省略内容长度为 8。

4　实验

4.1　实验设置

本文的研究工作，都是基于搜集到的文献[6]所使用的中文单轮短对话数据集而开展的。关于最近对话省略研究，一方面由于单轮与多轮对话研究存在许多不同之处，所以无法与其进行对比实验。另一方面，受限于程序源码，本文也很难将其实验结果在本文数据集上复现。因此，本文选择文献[6]的研究工作进行对比。

本文使用的数据集是文献[6]所使用的中文短文本数据集，是面向通用领域的单轮闲聊对话文本。表 1 给出了几组单轮对话中省略情况的示例，其中省略恢复部分用“(*)”进行了标识。可以看到，省略的成分、位置及数量都没有限制，省略可以是主语、宾语，可以出现在句首、句中或句尾，也可以同时出现多处省略。标注的省略内容有依据前文的语义补充名词，也有完善句法结构的代词等。

表 1　中文短文本省略样例

问	为什么 不 吃
答	不 想 吃
问补	你(*) 为什么 不 吃 饭(*)
答补	我(*) 不 想 吃 饭(*)
问	感冒 好 了 吗
答	好 了
问补	你(*) 感冒 好 了 吗
答补	我(*) 感冒(*) 好 了
问	去　医院　没有
答	去了
问补	你(*) 去 医院 没有
答补	我(*) 去 医院(*)了

数据集的省略分布情况如表 2 所示。

表2 短文本语料省略分布统计

	问题比	答案比	总占比/%
包含省略	7 855	10 467	61.7
无省略	6 989	4 377	38.3
总数	14 844	14 844	100

关于实验参数，Batch大小设为50，编码层和解码层输出都加了丢弃层，丢弃率大小为0.2。训练阶段初始学习速率为0.001，学习速率都是以每两轮以0.75的衰减速率递减。多头自注意力 h 设置为3。根据多轮对比试验，我们得到编码层和解码层的层数都设置为2，模型具有最佳性能。另外，实验中我们发现在解码层加入残差连接能够使模型表现得更好，所以我们在模型中默认都添加了残差连接[15]。

关于评价指标，为了和文献[6]的工作做对比，我们采用了句子级完全匹配正确率来评价模型的性能。具体方法如式(15)所示。

$$\text{正确率}=\frac{\text{有省略预测正确}+\text{无省略预测正确}}{\text{预测总数}} \tag{15}$$

其中，预测正确是指预测结果必须和标准答案完全一致才算正确预测。有省略预测正确是指模型正确识别有省略句子并做正确的省略恢复，无省略预测正确指模型正确识别无省略句子并且不做省略内容恢复。

本文采用联合模型同时进行省略位置识别和省略内容恢复两个子任务，为了更直观地分析两个子任务的性能，我们进一步给出了单个子任务的 F_1[16]值。

4.2 实验结果

本文的实验内容主要分为两部分：一是和文献[6]使用的Seq2Seq[17]模型做对比；二是探究各个模块对本文模型性能的影响。

首先，为了和Seq2Seq模型做对比，我们在同样的数据集上做了十折交叉验证，将十折平均正确率作为衡量模型泛化性能的标准。对比结果如表3所示。

表3 模型十折平均正确率对比表

模型	十折平均正确率
文献[6]使用的模型(Seq2Seq)	0.424
我们的模型	0.506

从表3可以看出，我们的模型相较于Seq2Seq完全匹配正确率有8.2个百分点的提升。这主要是因为在Seq2Seq模型中，待恢复序列的结构信息都是来自Bi-LSTM的编码信息，Seq2Seq模型通过最大似然估计来隐式地学习待恢复序列的语义结构，从而导致在预测时，模型会丢失部分源序列结构信息，造成完全匹配的正确率较低。而本文给出的模型则假定句子内词与词之间都可能存在省略，模型在训练和预测时充分考虑了待恢复序列的结构信息，有效解决Seq2Seq带来的“病句”问题。例如，对样例“下雨 特别 大”使用Seq2Seq模型输出的结果为“今天 下雨 巨 大”，Seq2Seq除了补充时间状语“今天”，还将原句中的词语“特别”替换成“巨”，修改了原始语句的信息。

此外，为了剖析模型的运行机制，增强模型的可解释性，本文主要对注意力模块、上下文编码、门控机制做了对比实验。实验数据集中训练集、开发集和测试集比例为8∶1∶1。实验结果如表4所示。

表4 模型各模块对比实验结果

模型	省略位置识别			省略内容恢复			
	P	R	F_1	P	R	F_1	Acc
1.Baseline	0.717	0.596	0.651	0.476	0.396	0.432	0.612
2.Baseline+交叉注意力	0.723	0.600	0.656	0.481	0.399	0.436	0.615
3.Baseline+交叉注意力+多头自注意力	0.763	0.584	0.662	0.492	0.376	0.426	0.622
4.Baseline+交叉注意力+多头自注意力+上下文	0.613	0.572	0.592	0.326	0.304	0.314	0.522
5.Baseline+交叉注意力+多头自注意力+上下文+门控	**0.754**	**0.668**	**0.708**	**0.519**	**0.460**	**0.488**	**0.637**

在表 4 中，为了能够更加细粒度地分析模型在省略恢复任务中的表现，本文将实验结果拆分为省略位置识别和省略内容恢复两个部分，Acc 评价指标沿用文献[6]采用的句子级别完全匹配的正确率。Baseline 为两层编码层、两层解码层以及融合字符级嵌入的结构。另外，为了能更直观地观察加入不同模块后模型性能的变化特点，我们从测试数据中选取一个样例进行分析，模型生成的结果如表 5 所示，括号内是省略补充的内容。

表 5　模型各模块恢复结果对比

上下文	生意 怎么样 不 是 很 好
省略句输入	不 是 很 好
标准答案	（我）（生意）不 是 很 好
模块预测结果对比	
1.Baseline	不（好）是 很 好
2.Baseline＋交叉注意力	（我）（觉得）不 是 很 好
3.Baseline＋交叉注意力＋多头自注意力	不 是 很 好
4.Baseline＋交叉注意力＋多头自注意力＋上下文	（生意）（生意）不 是 很 好
5.Baseline＋交叉注意力＋多头自注意力＋上下文＋门控	（我）（生意）不 是 很 好

从表 4 和表 5 可以看出：

（1）在 Baseline 模型中加入交叉注意力后，无论是在省略位置识别或者省略内容恢复中，模型的 P、R、F_1 值以及正确率都有小幅度提升。这是由于我们利用交叉注意力对前文和当前文本建模，将前文和当前文本借助交叉注意力机制进行语义表征，并在模型解码时能够动态参照前文和当前文本的语义信息，在编码端和解码端之间建立了一种“软对齐”。从表 5 测试样例看到模型在开始位置补充了“我”和“觉得”，这是因为数据集中存在大量“我觉得不是”开头的样例，编码端和解码端的语义对齐会使模型倾向于学习高频样例中的表达模式。

（2）当我们在模型 2 中加入多头自注意力机制（模型 3）后，在省略位置的识别上，模型的 F_1 值有 0.6 个百分点的提升，正确率也有 0.7 个百分点的提高。这是因为我们的实验语料为中文不规范短文本，很少存在规范的句式模板和语法格式，因此使用 Bi-LSTM 的编码结构很难让模型学习到省略补充的模式。而自注意力机制会使词语与词语之间进行相似度计算，语义相近的词会有较高的相似度，反之语义区别很大的词的相似度会很小。这种机制就可以很好地把富含语义信息的省略内容候选词和语义信息较少的助词、代词等进行区分，对把握句子内部语义结构有很大的帮助。在模型 3 中，Bi-LSTM 和自注意力结果是以 0.5 比例融合，变相削弱了 Bi-LSTM 的时序特征以及同解码端的语义对齐关系。从测试样例也可以看出模型 2 学到的高频表达模式被移除。

（3）模型 4 在模型 3 的基础上加入了上下文信息后正确率有 10 个百分点的下滑。但是，从表 5 的预测结果中可以发现，加入上下文信息后模型在开始位置补充了“生意”“生意”，而“生意”正是上下文中的省略候选。因此，我们认为加入上下文信息确实能够帮助模型从上下文中提取省略候选，但是实验中上下文信息也是以固定比例融合，可能由于比例过高导致其特征强度过大，在开始位置填充了两次“生意”，模型整体性能下滑。

（4）为了处理模型 3 和模型 4 在特征信息融合方面存在的问题，模型 5 在模型 4 基础上分别针对自注意力信息和上下文信息加入门控单元，取代固定值，自适应学习特征融合比例。从表 4 看到，模型 5 相较于模型 4 正确率提升了 11.5 个百分点。从表 5 测试样例结果也可以看到模型 5 准确地填充了省略内容。

5　总结

本文提出了一种基于上下文场景的多重注意力融合的中文单轮短对话省略恢复模型，旨在解决中文不规范文本中频繁出现的省略现象。其中，上下文场景以及多重注意力为模型提供多种信息来源，门控机制让模型自适应融合多源信息。联合的方法能够很好地处理误差传播问题。最后实验结果也表明该模型在省略恢复任务中有较好的表现。

真实对话场景下，多轮对话中的省略现象更为普遍。因此，今后的工作中，我们会开展关于多轮对话省略恢复的研究。

参考文献

[1] Kim Y J. Subject/object drop in the acquisition of Korean: A cross-linguistic comparison[J]. Journal of East Asian Linguistics, 2000, 9(4): 325-351.

[2] Huang Y, Zheng F, Su Y, et al. A theme structure method for the ellipsis resolution[C]//Proceedings of the 7th European Conference on Speech Communication and Technology, 2001.

[3] Ren X, Xu S U N, Wen J, et al. Building an ellipsis-aware Chinese dependency treebank for web text[C]//Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC-2018), 2018.

[4] 尹庆宇，张伟男，张宇，等.省略识别及恢复联合模型研究[J].计算机研究与发展，2015，52(11)：2460-2467.

[5] Kumar V, Joshi S. Non-sentential question resolution using sequence to sequence learning[C]//Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016: 2022-2031.

[6] 郑杰，孔芳，周国栋.基于序列到序列的中文短文本省略补全[J].中文信息学报，2018，32(12)：92-99.

[7] Su H, Shen X, Zhang R, et al. Improving multi-turn dialogue modelling with utterance ReWriter[J]. arXiv preprint arXiv: 1906.07004, 2019.

[8] Quan J, Xiong D, Webber B, et al. GECOR: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019: 4539-4549.

[9] Yang J, Tong J, Li S, et al. Recovering dropped pronouns in Chinese conversations via modeling their referents[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019: 892-901.

[10] Yin Q, Zhang Y, Zhang W, et al. Zero Pronoun Resolution with Attention-based Neural Network[C]//Proceedings of the 27th International Conference on Computational Linguistics. 2018: 13-23.

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

[12] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673-2681.

[13] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of Advances in Neural Information Processing Systems, 2017: 5998-6008.

[14] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv: 1409.0473, 2014.

[15] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[16] Chinchor N. MUC-4 evaluation metrics[C]//Proceedings of the 4th Conference on Message Understanding. Association for Computational Linguistics, 1992: 22-29.

[17] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Proceedings of Advances in Neural Information Processing Systems. 2014: 3104-3112

郑杰(1994—)，硕士研究生，主要研究领域为自然语言处理、篇章分析、指代消歧、省略恢复。
E-mail: 20175227008@stu.suda.edu.cn

周国栋(1967—)，博士，教授，主要研究领域为机器学习、自然语言处理、篇章理解。
E-mail: gdzhou@suda.edu.cn

孔芳(1977—)，通信作者，博士，教授，主要研究领域为机器学习、自然语言处理、篇章分析。
E-mail: kongfang@suda.edu.cn