

文章编号:

基于序列到序列的中文短文本省略补全

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: 省略作为一种常见的语言现象, 在上下文中普遍存在, 特别是在问答、对话等短文本中出现的频度更高。不同于传统的机器学习方法, 该文针对问答、对话这样的短文本, 构建了一个序列到序列的神经网络模型来实现对上下文中出现的省略进行识别和补全。在搜集和整理的短文本问答和对话语料上进行了各种实验, 验证了该模型在省略识别和恢复上能够取得较好的性能。

关键词: 序列到序列; 对话; 省略

中图分类号: TP391

文献标识码: A

Sequence to Sequence Model to Ellipsis Recovery for Chinese Short Text

ZHENG Jie, KONG Fang*, and ZHOU Guodong

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

Abstract : As a common linguistic phenomenon, ellipsis is ubiquitous in context, especially in short texts such as QA and dialogue. Unlike traditional machine learning methods, this paper builds a sequence-to-sequence neural network model for short texts such as QA and dialogue to identify and recover ellipsis that appears in the context. Various experiments were conducted on the collected and sorted short text corpus for QA and dialogue, verifying that the model can achieve good performance in ellipsis identification and recovery.

Key words: sequence to sequence; dialogue; ellipsis

0 引言

省略是一种很常见的语言现象, 需要通过上下文来理解被省略的内容。省略现象在中文、英文等多种语言中都有存在, 据 Kim^[1]统计, 在中文和英文的语料中, 英文大约有 96%是显式主语, 然而在中文中却只有 64%, 由此可以看出中文是一种省略现象十分频繁的语言。特别是在问答、对话等短文本中, 由于参与问答、对话的双方的语言习惯、表达方式各有差异, 会导致表达的内容呈现出随意和不规范的特点, 因此省略现象在这类短文本中尤为突出。由于缺少谈话双方的背

景知识, 当机器在面对这类会出现大规模的省略现象的短文本时, 如果不经过省略恢复处理, 机器将很难理解短文本的含义, 进而导致在问答、对话等系统中给出“错误”的回复, 带给用户极为糟糕的体验。因此, 对中文文本, 特别是对话、问答这种中文短文本进行省略恢复工作有着非常重要的意义。

本文的主要工作主要是针对中文短文本中的省略进行补齐, 结合深度学习的网络模型给出了一种基于序列到序列^[2]的省略检测和补全模型。

本文后续内容安排如下: 第一章简要介绍了中英文省略恢复的研究现状; 第二章详细介绍了引入注意力机制的序列到序列的神经网络模型; 第

收稿日期: 2017-03-16; 定稿日期: 2017-04-26

*通讯作者, E-mail: kongfang@suda.edu.cn

基金项目: 国家自然科学基金面上项目 (61472264); 人工智能应急项目 (61751206); 国家重点研发计划子课题 (2017YFB1002101); 国家自然科学基金青年基金 (61502149)

三章是实验过程及实验结果分析；第四章给出了总结。

1 相关工作

目前英文的省略恢复研究主要是针对动词短语进行，取得了一定的成果，代表性工作有：Dalrymple 等人^[3]和 Shieber 等人^[4]对英文进行语言学分析，在英文动词短语省略恢复方面建立了一套语言学理论体系；Nielsen 等人^[5]首次提出一种端到端的可计算的系统来对原始输入文本进行英文动词短语省略恢复；Liu 等人^[6]继续 Nielsen 等人的工作，提出一种目标检测、先行词词首消解以及先行词边界识别的动词短语省略识别步骤。

中文方面，相关研究主要集中在主语位置的省略和空语类的恢复方面，代表性工作包括：Kong 等人^{[7][8]}，Chen 等人^{[9][10][11]}，在标准公开中文数据集上进行中文零指代的研究；Cai 等人^[12]，Kong 等人^[13]，Xue 等人^[14]在中文标准公开数据集上进行关于中文空语类省略恢复研究。但这些研究多关注省略的恢复，对于省略的检测多采用规则方法。

在问答、对话等短文本的省略研究方面，代表性工作包括：Huang 等人^[15]对中文口语对话系统中的省略现象进行研究，这个中文口语对话系统是清华大学的校园导航 EasyNav，对话的形式一般是用户主导的连续型请求模式。Huang 等人基于这个中文口语对话系统提出了一种基于主题结构的省略恢复方法。但是他们提出的方法一方

面是基于他们的导航系统 EasyNav，不具有通用性；另一方面，他们提出的方法也只是停留在理论上，并没有提出一种具体的可计算模型。Yin 等人^[16]主要针对对话中的零代词省略进行恢复，提出了零代词恢复和零代词消解的框架，并采用联合模型减少管道模型带来的误差传播，但是他们采用的 OntoNote4.0 语料，是经过标注的具有句法规范的语料，无法体现中文口语对话的随意性和不规范性，并不具有代表性。Kumar 等人^[17]针对英文问答中指代消解问题分别构建了基于句法和语义的序列到序列模型，并结合这两种模型生成的联合模型来学习英文问句的语义和语言学的模式。

综上所述，我们可以发现，在中文方面，相关工作大都集中于规范的长文本数据集，而关于中文短文本的省略恢复研究很少；此外这些研究工作都是基于传统机器学习方法。本文首次针对中文不规范的短文本中的省略识别及恢复任务展开，提出了一个完整的端到端的神经网络模型，并通过实验验证了该模型在短文本的省略检测和恢复上的有效性。

2 省略恢复模型

这一章主要介绍本文应用在中文短文本省略识别和恢复任务中的序列到序列的神经网络模型。序列到序列模型是 2014 年被提出的，起初被应用在机器翻译中，后来在摘要生成、语音翻译等其他领域都得到了广泛的应用。

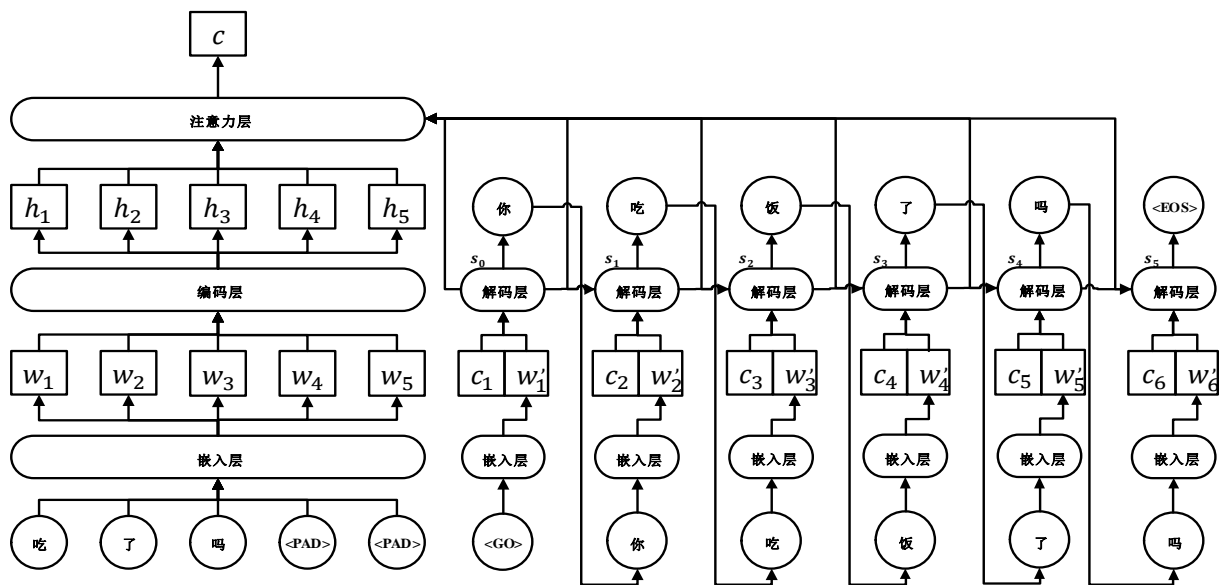


图 1. 序列到序列的省略补全模型

本文首次将序列到序列模型应用到中文短文本省略补全的研究中，从实验结果可以看出该模型在该任务中也取得了不错的表现。图 1 是本文模型的一个神经网络结构图，该框架主要包含三个层次，即嵌入层、编码层和解码层。接下来本文将从模型的这三个层面分别展开，进行模型详细细节的说明

2.1 嵌入层

嵌入层 (embedding) 的主要作用是获得词的分布式表示。它会维护一个嵌入矩阵 $D \in \mathbb{R}^{V \times f}$ ，其中 V 表示词表长度， f 表示词向量的维度。如图 1 所示，假设输入序列为 {“吃”，”了”，”吗”}，在预处理中首先会将该输入序列用填充符“<PAD>”补全到 batch 最大长度：{“吃”，”了”，”吗”，”<PAD>”，”<PAD>”}，之后该序列会被词典映射为标签序列：

$$\mathbf{X} = (x_1, \dots, x_m)$$

其中 x_1, \dots, x_m 表示输入序列中的单词在词典里的序号，这里 $m = 5$ 。嵌入层会根据标签序列的值 x_j 从嵌入矩阵 D 中找到序号所对应的向量 $w_j \in \mathbb{R}^f$ ，最终标签序列 \mathbf{X} 都会对应一个分布式表示，也就是词向量：

$$\mathbf{W} = (w_1, \dots, w_m)$$

本文在具体实验过程中，考虑到没有合适的大规模外部公共对话语料，所以并没有采用外部公共词向量来初始化嵌入矩阵，对嵌入矩阵 D 采用了随机初始化的方法，即用最小值为 $-\sqrt{3}$ ，最大值 $+\sqrt{3}$ 之间的正太分布随机数来初始化嵌入矩阵。设置嵌入矩阵 D 可训练，并让它的参数随着反向传播算法动态更新，学习词语的潜在语义。

2.2 编码层

2.2.1 LSTM

循环神经网络 (RNN) 由于其能够捕捉时序信息而被在自然语言处理任务中广泛应用。但是，面对较长的序列，循环神经网络会出现梯度消失和梯度爆炸的问题。而 LSTM^[18] (Long Short-Term Memory) 1997 年被提出，作为 RNN 的一个变体，可以解决梯度消失的问题。一个 LSTM 神经元由 3 个门组成，分别是输出门，输入门和遗忘门，这三个门会控制传输到下一个时序信息的多少。一般的，LSTM 神经单元的在 t 时刻的更新规则如

公式 (1) (2) (3) (4) (5) (6) 所示：

$$i_t = \sigma(W_i h_{t-1} + U_i X_t + b_i) \#(1)$$

$$f_t = \sigma(W_f h_{t-1} + U_f X_t + b_f) \#(2)$$

$$\tilde{c}_t = \tanh(W_c h_{t-1} + U_c X_t + b_c) \#(3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \#(4)$$

$$o_t = \sigma(W_o h_{t-1} + U_o X_t + b_o) \#(5)$$

$$h_t = o_t \odot \tanh(c_t) \#(6)$$

σ 是 sigmoid 函数， \odot 是对应元素相乘， X_t 是 t 时刻的输入 (经过嵌入层后的向量)， h_t 是 t 时刻隐层状态向量。 U_i, U_f, U_c, U_o 表示输入 X 的权重矩阵， W_i, W_f, W_c, W_o 表示隐层状态的权重矩阵， b_i, b_f, b_c, b_o 表示偏置。

2.2.2 Bi-LSTM

一般的单向 LSTM 网络只能学习历史信息，而 Bi-LSTM 的思想是将前向传播的状态和后向传播的状态进行拼接，并且已有实验^[19]证明这种双向结构可以捕捉历史信息和未来信息。它的结构如图 2 所示：

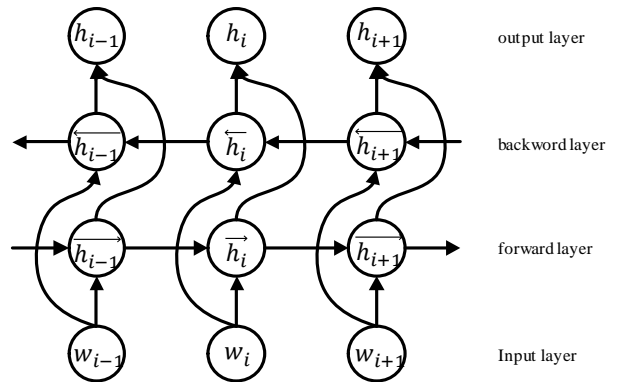


图 2. 编码层双向 LSTM 结构图

在本文的序列到序列模型中，编码层是用来获得源端句子的语义表征的。Bi-LSTM 模型可以获得源端序列的历史信息和未来信息，能够更好地表征句子的语义信息，所以在编码层我们采用 Bi-LSTM 的神经网络结构。它的具体运算过程如下。

标签序列 \mathbf{X} 在经过嵌入层得到的词向量序列 \mathbf{W} ，在 Bi-LSTM 中，前向和后向 LSTM 网络都是单独计算的，所以 \mathbf{W} 会分别传给前向和后向 LSTM 作为输入向量。假设在第 i 时刻 ($i \leq m$)，在前向 LSTM 中，第 i 时刻的输出如公式 (7) 所示：

$$\vec{c}_i, \vec{h}_i = f(\vec{c}_{i-1}, \vec{h}_{i-1}, w_i) \#(7)$$

在后向 LSTM 中, 第 i 时刻的输出如公式 (8) 所示:

$$\vec{c}_i, \vec{h}_i = f(\vec{c}_{i+1}, \vec{h}_{i+1}, w_i) \#(8)$$

\vec{c}_i 和 \vec{h}_i 作为编码层隐层状态向量分别在前向 LSTM 和后向 LSTM 中传递, \vec{h}_i 和 \vec{h}_i 分别作为 Bi-LSTM 中前向 LSTM 和后向 LSTM 的输出向量, 所以编码层最后的输出如公式 (9) 所示:

$$h = (h_1, h_2, \dots, h_m) \#(9)$$

其中, 在第 i 时刻下, h_i 为 \vec{h}_i 和 \vec{h}_i 的拼接, 如公式 (10) 所示:

$$h_i = [\vec{h}_i || \vec{h}_i] \#(10)$$

在图 1 例子中, 编码层最终将输入序列 {“<GO>”, “你”, “吃”, “饭”, “了”, “吗”} 编码成向量 (h_1, h_2, \dots, h_m) 。

2.2 解码层

2.3.1 注意力机制

注意力机制最初是指人的心理活动指向或集中于某些事物的能力, 引入到神经网络中就是指神经网络模型对一些输出向量“注意”, 对另一些输出向量“忽略”。而模型的这种“注意”和“忽略”是用权重来动态模拟的, 和当前状态相关的向量会被分配高权重, 不相关的向量会被分配低权重。那些被分配高权重的向量在数值计算中影响较大, 而被分配低权重的向量对数值运算的结果影响较小, 这就是模型会动态选择“注意”相关的内容而忽视不相关内容的原因。注意力机制的原理^[20]如图 3 所示:

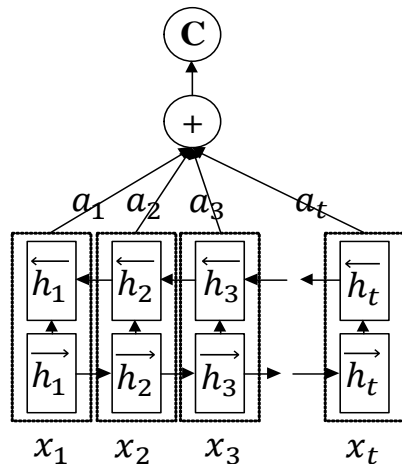


图 3. 注意力机制原理图

2.3.2 解码

根据 RNN 网络结构具有捕捉时序状态信息的特点, 传统的编码解码 (Encoder-Decoder) 模型认为编码层最后一个时序的输出向量可以作为源端序列的语义表征。这种做法存在两种弊端, 一方面是在对长序列进行处理时, 很难把握长句的依赖关系, 不可避免会造成语义信息的丢失; 另一方面是这种表征方式也无法获得源端序列的焦点信息。因此, 我们采用了 Bahdanau Attention, 对编码层的输出向量计算求得注意力权重, 并得到上下文向量 (Context Vector) 加入到解码层的输入。具体做法如下:

首先, 在解码第一步, 序列开始标志符“<GO>”在经过字典映射到标签 y_1 并通过嵌入层得到词向量 $w_1 \cdot c_1$ 表示第 1 时刻的上下文向量, 要计算 c_1 首先需要计算得分 e_{i1} , 如 (11) 所示:

$$e_{i1} = V \tanh(W_h h_i + W_s s_0 + b) \#(11)$$

V, W_h, W_s 都是权重矩阵, b 是偏置项。 h_i 表示第 i 时刻的编码层输出, 即 $[\vec{h}_i || \vec{h}_i]$ 。 s_0 表示解码端的初始状态向量。这时, 我们可以通过得分 e_{i1} 来计算权重 a_{i1} , 如 (12) 所示:

$$a_{i1} = \frac{\exp(e_{i1})}{\sum_k \exp(e_{k1})} \#(12)$$

$k \leq m$, m 是源端序列的最大长度, 在图 1 中 m 为 5。所以, 上下文向量 c_1 通过公式 (13) 计算得到:

$$c_1 = \sum_i a_{i1} h_i \#(13)$$

这时, s_0, w_1, c_1 通过解码层 LSTM 单元计算, 如公式 (14), 得到第一个时序的状态向量 s_1 和预测标签 \bar{y}_1 :

$$\bar{y}_1, s_1 = f(s_0, w_1, c_1) \#(14)$$

同时, 第一个时序的预测标签作为第二个时序的输入标签:

$$y_2 = \bar{y}_1 \#(15)$$

在图 1 的例子中, 第一步解码预测标签 \bar{y}_1 通过词典映射后就是符号“你”。

至此, 第一步解码计算完成, 之后解码步骤类似, 直到输出结束符号“<EOS>”为止。最后可以得到编码端输出标签 $y = (y_1, y_2, \dots, y_t)$, 在图 1 中经过字典映射后就是序列 {“你”, “吃”, “饭”, “了”, “吗”, “<EOS>”}。

在训练模式中本文采用公式 (16) negative log-likelihood 来计算损失:

$$loss = \frac{1}{t} \sum_{j=1}^t -\log(\mathbf{p}_j^*) \quad (16)$$

\mathbf{p}_j^* 表示j时刻概率分布 \mathbf{p}_j 在目标词处的概率。

在推理模式下, 本文模型采用的标签选择策略是 beam search 算法, 在下面的章节中本文会详细介绍。

2.3.3 Beam Search

在推理过程中, 常用的方法使用的是贪心算法搜索。这种方法实现简单, 但是结果往往不是最优解。因此, 为了增加候选预测序列集合, 提高模型的性能, 我们采用了一种 beam search 的解码方法。假设 beam size 大小为 K , 词典维度为 V , 它的思想是每次从概率分布中选择最高的 K 个值所对应的标签作为预测结果。设第 i 时刻的最高的 K 个得分结果按照公式 (17) 计算为:

$$Score_{1 \sim K}^i = \text{top}_K \left(\bigcup_{k=1}^K \bigcup_{j=1}^V (Score_k^{i-1} + (-\log(\mathbf{p}_{ij}))) \right) \quad (17)$$

其中 \mathbf{p}_{ij} 是第 i 时刻词典上第 j 个词的概率, 那么 $Score_{1 \sim K}^i$ 表示第 i 时刻下从 $K * V$ 个概率中选择最大的 K 个的结果, 预测的标签就是 $Score_{1 \sim K}^i$ 对应的词典中的序号。

在下面的实验章节中我们会选取不同的 beam size 进行实验, 并分析选取不同的 beam size 后对系统的性能影响情况。

3 实验

3.1 数据集

本文使用的实验数据集是通过网络以及其他各种途径搜集的问答和一些真实场景中的单轮短文本对话, 以此为基础人工标注形成的。最终的语料包括 14000 多个短文本问题/回答对, 经过人工标注后得到的省略情况如表 1 所示。

表 1. 短文本语料中省略分布情况统计

	问题	答案	占比 (%)
包含省略	7855	10467	61.7
无省略	6989	4377	38.3

总数	14844	14844	100
----	-------	-------	-----

从表 1 中可以看到, 包含省略的实例在全部语料中的总占比为 61.7%, 无省略的实例在全部语料中总占比为 38.3%, 包含省略和无省略样例的比例约为 1.6:1, 正负样例比例较为均衡。

表 2 给出了摘自标注语料库的部分示例, 其中省略成份以“(*)”表示。

表 2. 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	不需要了 谢谢
答	好吧
问补	我(*)不需要了 谢谢
答补	好吧

3.2 实验设置

在语料的预处理工作中, 为了方便模型训练, 本文在一个 batch 中将长度不足最大长度的短句填充零元素标识符“<PAD>”到 batch 中最大句子长度。为了能够让模型识别句子的起始位置和结束位置的信息, 本文分别在句子的开头处添加开始标志“<GO>”, 并且在句子的结尾添加句子结束标志“<EOS>”。模型在解码层预测可能的标签序列时, 如果预测结果为“<EOS>”就停止预测。

关于实验中超参数的设置, 实验中设置隐层神经元个数为 1536 个, 编码端神经网络层数为 2 层, batch size 大小为 50, 学习速率为初始值 0.001 的指数衰减, 每 50 步进行一次衰减, 衰减速率为 0.98。每个神经单元的 dropout 为 0.2, 训练集、验证集、测试集以 8:1:1 进行分割, 迭代次数为 200 次。

关于评价指标, 由于在省略恢复任务中, 没有一个统一的评价标准, 我们借鉴了阅读理解 SQuAD 中完全匹配的方法, 即预测结果必须和标准答案完全一致才算一个正确预测, 并采用准确率对模型性能进行衡量。具体方法如下:

$$\text{准确率} = \frac{\text{有省略预测正确} + \text{无省略预测正确}}{\text{预测总数}} \quad (18)$$

其中,有省略预测正确是指模型补全后的实例和标准补全后的实例一致,无省略预测正确是指模型未补全的实例和标准未补全的实例一致,具体会在下一小节进行详细的解释。

3.3 实验结果

首先,为了研究不同 beam size 对模型实验结果的影响,本文选取了 beam size 从 1 到 20 并分别进行实验,实验结果如图 4 所示,横坐标表示 beam size 大小,纵坐标表示准确率。

从图 4 中可以看出,在解码层加入了 beam search 后系统性能是有从 0.451 到 0.550 的一个百分点的明显提升,这也证明了 beam search 确实可以增加模型预测结果集合,提高模型的性能。但是,从 beam size 为 2 开始,准确率的提升就逐渐放缓,当 beam size 到 17 时准确率几乎没有

变化,这反映已经达到模型的性能上限,继续增加 beam size 只会增加模型训练时间。对于图 4 中出现的准确率抖动现象,一方面这是由于模型中可学习的变量参数都是随机初始化的,不同的初始化参数在通过梯度下降可能会得到不同的局部最优值;另一方面为了节省模型训练时间,本文采用了一种神经网络训练中常用的方法 early stop,即在模型训练时同时观察验证集的损失变化,当验证集损失到达最小时会提前结束训练,这也就可能会导致当提前结束的时候,模型可能没有得到充分学习,从而带来准确率抖动的情况。

另外,本文除了对 beam search 进行实验,为了能够客观地反映本文中的省略恢复模型的性能,我们将 beam size 设置为 1,即采用简单的贪心搜索策略,采用语料集上十折交叉验证的评测策略。十折交叉验证的结果分布如表 3 所示,准确率如表 4 所示。

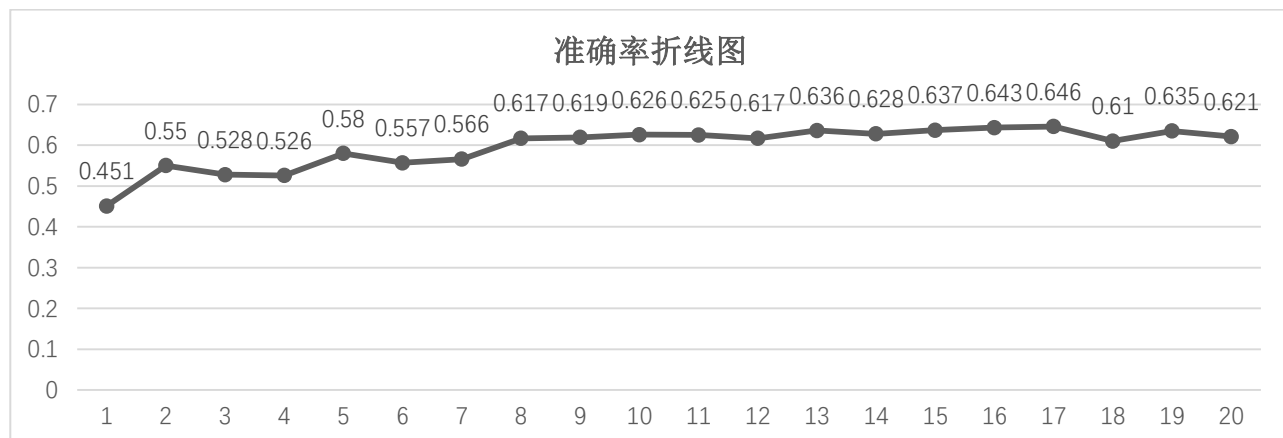


图 4 不同 beam size 对模型准确率的影响

表 3 十折交叉验证结果分布表

序号	1	2	3	4	5
有省略预测正确	244	234	278	242	346
无省略预测正确	353	369	306	321	318
有省略没有补	229	277	107	206	69
有省略补错	426	388	514	451	484
无省略预测错	198	182	245	230	233
序号	6	7	8	9	10
有省略预测正确	249	282	230	304	337
无省略预测正确	322	350	361	350	345
有省略没有补	177	197	262	146	106
有省略补错	473	420	407	449	456
无省略预测错	229	201	190	201	206

表4 十折交叉验证准确率

序号	1	2	3	4	5
Acc	0.412	0.416	0.403	0.388	0.458
序号	6	7	8	9	10
Acc	0.394	0.436	0.408	0.451	0.470

表3中，有省略预测正确是指训练样例中目标端序列是存在省略现象的，并且模型预测后的补全结果和目标端完全一致，算一次正确的预测；无省略预测正确是指训练样例中目标端没有省略现象，并且模型的预测结果并没有对源端序列进行省略补全，和目标端序列完全一致，算一次正确的预测；有省略没有补是指训练样例中目标端序列存在省略，但是模型的预测结果并没有进行补全操作，和目标端结果不一致，算错误预测；有省略补错是指训练样例中目标端序列是存在省略的，模型对源端序列进行补全操作后与目标端序列不一致，即补错的情况，算错误预测；无省略预测错误是指目标端序列不存在省略现象，而模型却进行了省略恢复操作，算错误预测。下面本文通过从表5到表9中分别用一个例子来对这五种类别进行更直观的解释。

表5 有省略预测正确的样例

模型预测结果	我 之前 就 看见 你 了
原始回答	之前 就 看见 你 了
标准回答(补)	我(*)之前 就 看见 你 了

表6 无省略预测正确的样例

模型预测结果	我 也 不 知 道
原始回答	我 也 不 知 道
标准回答(补)	我 也 不 知 道

表7 有省略没有补的样例

模型预测结果	谁 的 微 信 号
原始回答	谁 的 微 信 号
标准回答(补)	那 是 (*) 谁 的 微 信 号

表8 有省略补错

模型预测结果	我 生 意 没 有 生 意
原始回答	没 有 生 意
标准回答(补)	我(*)没 有 生 意

表9 无省略预测错误

模型预测结果	今 天 下 雨 巨 大
原始回答	下 雨 特 别 大
标准回答(补)	下 雨 特 别 大

从表3可以看到，在十次实验结果中，有省略补错的现象占有很高的比例，表8的有省略补错的样例中可以看出模型在预测时并没有考虑句子的语义信息，会存在“病句”的情况，这也在很大程度上制约了模型的性能。另外，从省略恢复结果中可以看出，省略补全的内容大多以主语等语法结构单元为主，这是符合中文表达的特征，但是考虑到省略恢复是为其他任务服务的，对于语义信息的补全可能会比语法结构的补全更加重要。这些问题在接下来的工作中会重点考虑。

4 总结

本文首次提出了一种基于序列到序列的中文短文本省略恢复模型。该模型在编码层采用Bi-LSTM学习源端序列的抽象表征，在解码层采用beam search算法进行解码，并引入注意力机制让模型自动学习焦点信息。最后，在中文短文本问答和对话语料上的实验表明序列到序列的模型在处理中文短文本省略恢复问题上有较好的表现。

之后的工作会重点处理预测结果中出现“病句”的情况，一方面考虑在模型训练中增加规则限制，另一方面在beam size一定的情况下借助语言生成的相关策略通过后处理对答案进行二次评估，选择更符合要求的结果。

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] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.
- [3] Mary Dalrymple, Stuart M. Shieber, and Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. Linguistics and Philosophy, 14(4):399-452.
- [4] Stuart M. Shieber, Fernando C. N. Pereira, and Mary Dalrymple. 1996. Interactions of scope and ellipsis. Linguistics and Philosophy, 19(5):527-552.
- [5] Leif Arda Nielsen. 2005. A corpus-based study of Verb Phrase Ellipsis Identification and Resolution. Doctor of philosophy, King's College London.
- [6] Liu Z, Pellicer E G, Gillick D. Exploring the

- steps of verb phrase ellipsis[C]//Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016). 2016: 32-40.
- [7] Kong F, Zhou G. A tree kernel-based unified framework for Chinese zero anaphora resolution[C]//Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2010: 882-891.
- [8] Fang K, Guodong Z. Chinese Zero Pronoun Resolution: A Chain to Chain Approach[C]//National CCF Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2017: 393-405.
- [9] Chen C, Ng V. Chinese zero pronoun resolution with deep neural networks[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016, 1: 778-788.
- [10] Chen C, Ng V. Chinese zero pronoun resolution: An unsupervised probabilistic model rivaling supervised resolvers[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 763-774.
- [11] Chen C, Ng V. Chinese Zero Pronoun Resolution: An Unsupervised Approach Combining Ranking and Integer Linear Programming[C]//AAAI. 2014: 1622-1628.
- [12] Cai S, Chiang D, Goldberg Y. Language-independent parsing with empty elements[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2. Association for Computational Linguistics, 2011: 212-216.
- [13] Kong F, Zhou G. A Clause-Level Hybrid Approach to Chinese Empty Element Recovery[C]//IJCAI. 2013: 2113-2119.
- [14] Xue N, Yang Y. Dependency-based empty category detection via phrase structure trees[C]//Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013: 1051-1060.
- [15] Huang Y, Zheng F, Su Y, et al. A theme structure method for the ellipsis resolution[C]//Seventh European Conference on Speech Communication and Technology. 2001.
- [16] Yin Q. A joint model for ellipsis identification and recovery[J]. J. Comput. Res. Dev, 2015, 52(11): 2460-2467.
- [17] 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.
- [18] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
- [19] M. Schuster and K. Paliwal, "Bidirectional re-current neural networks," IEEE Trans. on Signal Processing, vol. 45, no. 11, pp. 2673 - 2681, 1997.
- [20] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.



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



孔芳 (1977-), 通讯作者, 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



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