情报学报　2019 年 10 月　第 38 卷　第 10 期
Journal of the China Society for Scientific and Technical Information, Oct. 2019, 38(10): 1102-1116

DOI: 10.3772/j.issn.1000-0135.2019.10.010

基于序列到序列模型的生成式文本摘要研究综述

石　磊[1]，阮选敏[2]，魏瑞斌[1]，成　颖[2,3]

（1. 安徽财经大学管理科学与工程学院，蚌埠　233030；2. 南京大学信息管理学院，
南京　210023；3. 山东师范大学文学院，济南　250014）

摘　要　相较于早期的生成式摘要方法，基于序列到序列模型的文本摘要方法更接近人工摘要的生成过程，生成摘要的质量也有明显提高，越来越受到学界的关注。本文梳理了近年来基于序列到序列模型的生成式文本摘要的相关研究，根据模型的结构，分别综述了编码、解码、训练等方面的研究工作，并对这些工作进行了比较和讨论，在此基础上总结出该领域未来研究的若干技术路线和发展方向。

关键词　生成式摘要；序列到序列模型；编码器-解码器模型；注意力机制；神经网络

Abstractive Summarization Based on Sequence to Sequence Models: A Review

Shi Lei[1], Ruan Xuanmin[2], Wei Ruibin[1] and Cheng Ying[2,3]

(1. *School of Management Science and Engineering, Anhui University of Finance and Economics, Bengbu*　233030;
2. *School of Information Management, Nanjing University, Nanjing*　210023;
3. *School of Chinese Language and Literature, Shandong Normal University, Jinan*　250014)

Abstract: Compared with the early abstractive summarization method, the text summarization method based on Sequence to Sequence models is much closer to the process of human-written summaries, and the quality of the generated summary has also been significantly improved, which has attracted increasing attention from the academic community. This paper reviews the research related to abstractive summarization based on Sequence to Sequence models in recent years. According to the structure of the model, this paper summarizes the research on the model in terms of encoding, decoding, training, and so on, and it compares and discusses these works. On this basis, some technical routes and development directions for future research in this field are put forward.

Key words: abstractive summarization; Sequence to Sequence model; encoder-decoder model; attention mechanism; neural networks

1　引　言

自 1958 年 Luhn[1] 开启了自动摘要研究以来，该领域已经形成了丰硕的成果。目前，自动摘要方法大体上可以分为两类[2]：抽取式（extractive summarization）和生成式（abstractive summarization）。抽

收稿日期：2019-01-18
基金项目：国家社会科学基金重大项目“中国近现代文学期刊全文数据库建设与研究（1872—1949）”（17ZDA276）。
作者简介：石磊，男，1981 年生，硕士，讲师，主要研究方向为自然语言处理；阮选敏，女，1994 年生，硕士研究生，主要研究方向为信息检索、信息计量；魏瑞斌，男，1973 年生，博士，教授，主要研究方向为科学计量学、数据分析与数据可视化；成颖，男，1971 年生，教授，博士生导师，主要研究方向为信息行为、信息检索，E-mail：chengy@nju.edu.cn。

取式的基本做法是从原文中抽取部分重要的句子形成摘要，研究重点集中在句子的重要性判断、筛选以及排序等。生成式摘要的基本思路是在理解原文语义的基础上，凝练其思想与概念，以实现语义重构。抽取式是先前自动摘要研究的主导方法[3]，不过，在TAC2011等评估中[4]，表现最优的结果仅被认为“勉强可接受”[5]。

Khan等[6]对止于2014年的生成式摘要的相关研究进行了综述，从方法上将其分为基于结构以及基于语义两类。前者生成摘要的主要不足是语言质量相对较差，比如，语句中包含较多的语法错误；后者生成的摘要具备简明、内聚、信息丰富以及低冗余等优点，不足之处在于主要使用浅层自然语言处理技术。近年来，深度学习技术为自动摘要研究提供了新的思路，其中，序列到序列（sequence to sequence，Seq2Seq）模型的研究与应用最为广泛。该模型由Cho等[7]和Sutskever等[8]提出，基本思想是利用输入序列的全局信息推断出与之相对应的输出序列，由编码器（encoder）和解码器（decoder）构成。Rush等[9]首次将该模型应用于生成式摘要，和先前的生成式方法相比，该模型是在“理解”文本语义的基础上生成摘要，更加接近人工摘要的生成过程[10]。随之，学界提出了一系列基于Seq2Seq的生成式摘要模型，对编码器、解码器以及训练方法等开展了卓有成效的研究工作。基于该模型生成的摘要在语言流畅性、连贯性等方面让学界看到自动摘要实用化的希望[11]。

本文以（“abstractive” OR “sequence” OR “seq2seq” OR “neural”）AND “summarization”及其对应的中文为检索词分别检索中国知网、万方、Web of Science（WoS）以及Google Scholar等数据库，在文献阅读过程中再根据参考文献不断扩展文献范围，最后发现切题文献共50篇，其中会议论文30篇，开放存取论文16篇，期刊论文4篇。鉴于该主题的研究已经形成了丰富的文献积累，有必要对其进行梳理，在深入提炼的基础上为后继研究提供参考。本文第2节阐述基础Seq2Seq模型，第3节按照模型的结构分别梳理编码、解码以及训练等方面的研究进展，第4节与第5节分别对本领域工作展开讨论并做出总结。

2　基础模型

在Cho等[7]的工作中，编码器和解码器均采用循环神经网络（recurrent neural network，RNN）。编码器将输入的一个可变长序列$X=(x_1,\cdots,x_T)$编码为一个固定的语义向量；解码器从该向量中提取语义信息，输出另一个可变长序列$Y=(y_1,\cdots,y_{T'})$，序列中的每个词项采用词向量表示。模型的具体计算过程如下：编码器基于输入的词向量x_i以及上一词项的隐层h_{i-1}计算当前词项隐层h_i[公式(1)]，再通过隐层向量计算语义向量c[公式(2)]；解码器在每个时间步t，基于语义向量c、上一时间步隐层s_{t-1}和生成的上一个词项y_{t-1}，计算当前隐层s_t[公式(3)]，再基于语义向量c、当前隐层s_t和生成的上一个词项y_{t-1}，推导当前词项y_t的分布[公式(4)]。

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

$$c = q(\{h_1,\cdots,h_T\}) \tag{2}$$

$$s_t = f(y_{t-1}, s_{t-1}, c) \tag{3}$$

$$p(y_t|y_{<t}, X) = g(y_{t-1}, s_t, c) \tag{4}$$

其中，f和q为非线性激活函数；g通常是softmax函数，用于产生词项在词汇表V中的概率分布，一般用贪婪算法（greedy search）取最大概率对应的词项作为输出。

模型使用有标注的训练集D进行训练，D由大量源文本x和对应的标准摘要y构成。训练的基本目标是优化参数集θ，使输入序列x的输出结果最大似然于序列y，即最大化$\log p(y|x;\theta)$，等同于最小化交叉熵损失，损失函数为

$$L_{\text{MLE}}(\theta) = -\sum_{(x,y)\in D}\log p(y|x;\theta) = -\sum_{(x,y)\in D}\sum_{t}\log p(y_t|y_{<t}, x;\theta) \tag{5}$$

该模型的不足之处是，编码器把源文本中的所有信息表示为一个固定的语义向量，解码器在生成每一个词项时均参考该向量，这为神经网络处理长文本带来了困难。Cho等[7]的实验证实随着文本长度的增加，模型的表现快速下降。对此，Bahdanau等[12]在模型中引入了注意力（attention）机制，目的是使解码器在生成每一个词项时重点关注源文本中的特定部分，即解码过程不再依赖原先固定的语义向量c，而是利用动态的c_t[公式(6)~公式(8)]，c_t是时间步t所有词项隐层的加权。

$$c_t = \sum_{i=1}^{T}\alpha_{ti}h_i \tag{6}$$

$$\alpha_{ti} = \text{softmax}(e_{ti}) = \frac{\exp(e_{ti})}{\sum_{k=1}^{T}\exp(e_{tk})} \tag{7}$$

$$e_{ti} = \text{score}(h_i, s_{t-1}) \tag{8}$$

其中，e_{ti}是注意力得分，用来估计位置i附近的输

入和时间步 t 的输出之间的匹配程度；α_{ti} 是注意力分布，代表在时间步 t 每个词项的隐层 h_i 被解码器关注的程度。每个输出词项的分布相应地由公式(4)更新为公式(9)，

$$p(y_t|y_{<t},X) = g(y_{t-1},s_t,c_t) \tag{9}$$

Bahdanau 等[12]的实验结果表明，带有注意力机制的模型在机器翻译任务上取得了更好的成绩，对于句子长度变化更具鲁棒性。注意力机制的加入使得 Seq2Seq 模型更加完善，之后大量相关研究都建立在该模型的基础上。带注意力机制的 Seq2Seq 模型结构如图 1 所示，图中编码器以双向 RNN 为例，解码器以单向 RNN 为例，以生成词项 y_2 为例对编码、解码过程做出示意。

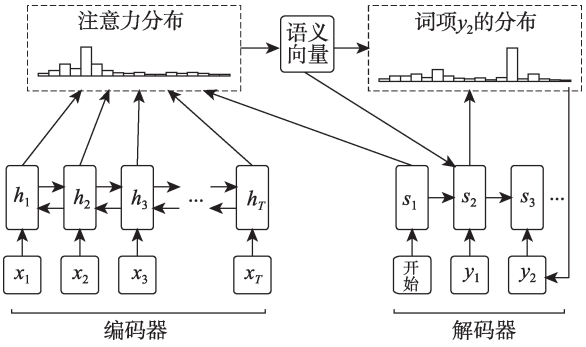

图 1 带注意力机制的 Seq2Seq 模型示意图

3 衍 化

3.1 编 码

Rush 等[9]在论文中提出了三种编码器方案：①词袋编码器，将输入序列中的词向量平均后作为语义向量，并不考虑词的顺序；②卷积编码器，使用时滞神经网络（time-delay neural network，TDNN）对词向量交替进行时间卷积（temporal convolution）和最大池化（max pooling）以计算出语义向量；③基于注意力机制的编码器，即 ABS（attention-based summarization）模型。ABS 模型基于词袋编码器，在计算语义向量时，不仅考虑输入序列中的词向量 x_i，还考虑解码器已输出的最近 R 个词的向量 y_R^{t-1}，模型用 y_R^{t-1} 在输入序列和输出序列之间做对齐。ABS 模型在 DUC2004 和英语 Gigaword 测试集上的表现并不理想（表 1），生成的摘要存在着语法错误、事实歪曲等问题[9]，但该工作作为一次有益的尝试，为后续研究带来了很多启发。编码器的工作相当于阅读一篇文本并理解其语义，这对机器来说无疑是相当具有挑战性的。人在阅读时，可能会采取选择性阅读、多遍阅读、分块阅读，还可能会借助外部资源理解文本，受此启发，学界提出多种编码器改进方案。由于注意力机制的设计和编码方式密切相关，本节在梳理编码方面的工作时将涵盖注意力相关内容。

3.1.1 选择性编码

借鉴人类在阅读时会标注出重点内容的做法，Zhou 等[13]提出了选择性编码模型。该模型通过设置一个门控网络对编码器生成的词项隐层进行权重标注，相当于“选择”出相对重要的内容，使解码器可以有针对性的读取源文本。模型的具体实现使用门控循环单元（gated recurrent unit，GRU）计算词项的隐层 $h_i = \mathrm{GRU}(x_i,h_{i-1})$ 以及文本表示 h_{sent}，然后将二者输入基于多层感知机的门控网络以计算出每个词项的权重向量 weight_i，

$$\mathrm{weight}_i = \sigma(h_i,h_{\mathrm{sent}}) \tag{10}$$

之后用权重向量更新隐层 h_i，得到新词项隐层 h_i^{new}，

$$h_i^{\mathrm{new}} = h_i \otimes \mathrm{weight}_i \tag{11}$$

其中，$\otimes$ 代表向量点乘运算（element-wise multiplication）。

Zeng 等[14]提出的“再读（read-again）”模型与 Zhou 等[13]工作类似，不同点是该模型没有直接用权重向量更新当前词项的隐层，而是用另一个 GRU 对源文本进行二次编码，然后将权重向量用于更新第二次编码生成的词项隐层 $h_i^{(2)}$，

$$h_i^{(2)} = (1 - \mathrm{weight}_i) \otimes h_{i-1}^{(2)} + \mathrm{weight}_i \otimes \mathrm{GRU}^{(2)}(x_i,h_{i-1}^{(2)}) \tag{12}$$

该模型综合考虑了当前词项的权重、当前隐层 $h_i^{(2)}$ 以及上一个词项的隐层 $h_{i-1}^{(2)}$。

Zeng 等[14]还将 GRU 更换为长短时记忆网络（long short-term memory，LSTM）做了对比实验，考虑到 LSTM 使用非线性激活函数来更新隐层，无需单独计算 weight_i，直接利用第一遍编码获得的词项隐层和文本表示即可更新 $h_i^{(2)}$，

$$h_i^{(2)} = \mathrm{LSTM}^{(2)}([x_i; h_i^{(1)}; h_{\mathrm{sent}}^{(1)}],h_{i-1}^{(2)}) \tag{13}$$

实验结果表明，GRU 和 LSTM 的表现不分伯仲。

Lin 等[15]提出的全局编码（global encoding）模型本质上也属于选择性编码，该模型使用门控单元对第一遍编码的输出赋予权重，从而选择出与全局语义相关的信息。模型的创新之处是在第二遍编码时使用了 CNN（convolutional neural network）和自注意力（self-attention）机制。基于卷积核参数共

表 1　短文本摘要模型基本信息及评测结果

作者	模型	框架	DUC2004			英语 Gigaword			LCSTS		
			ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Rush 等[9]（2015）	ABS（基线）	注意力编码 → NNLM	26.55	7.06	22.05	29.55	11.32	26.42			
Rush 等[9]（2015）	ABS+	注意力编码 → NNLM	28.18	8.49	23.81	29.76	11.88	26.96			
Chopra 等[32]（2016）	RSA-Elman	卷积编码 → Elman	28.97	8.26	24.06	33.78	15.97	31.15			
Gulcehre 等[36]（2016）	指针 Softmax	GRU → GRU				**37.29**	**17.75**	**34.70**			
Gu 等[40]（2016）	CopyNet	GRU → GRU							35.00	22.30	32.00
Zeng 等[14]（2016）	再读+拷贝机制	LSTM → LSTM	29.89	9.37	25.93						
Ayana 等[51]（2016）	最小风险训练	GRU → GRU	30.41	**10.87**	26.79	36.54	16.59	33.44	**38.20**	**25.20**	**35.40**
Zhou 等[13]（2017）	选择性编码	GRU → GRU	29.21	9.56	25.51	36.15	17.54	33.63			
Cao 等[31]（2017）	抽取事实	GRU → GRU				**37.27**	17.65	34.24			
Gehring 等[33]（2017）	ConvS2S	多层 CNN → 多层 CNN	**30.44**	**10.84**	**26.90**	35.88	17.48	33.29			
Li 等[44]（2017）	深度循环生成解码器	GRU → GRU + VAE	**31.79**	10.75	**27.48**	36.27	17.57	33.62	36.99	24.15	34.21
Amplayo 等[49]（2018）	解码器融入主题信息	GRU → GRU				**37.04**	16.66	**34.93**			
Zhou 等[39]（2018）	序列拷贝网络	GRU → GRU				35.93	17.51	33.35			
Wang 等[68]（2018）	ConvS2S+主题+强化学习	多层 CNN → 多层 CNN	**31.15**	**10.85**	**27.68**	36.92	**18.29**	**34.58**	**39.93**	21.58	**37.92**
Li 等[54]（2018）	Actor-Critic 训练	GRU → GRU	29.41	9.84	25.85	36.05	17.35	33.49	37.51	**24.68**	35.02
Lin 等[15]（2018）	全局编码 + CNN+ 自注意力	LSTM + CNN → LSTM				36.30	**18.00**	33.80	**39.40**	**26.90**	**36.50**
Song 等[69]（2018）	融入结构的拷贝机制	LSTM → LSTM				35.47	17.66	33.52			

注：DUC2004 是美国国家标准与技术研究院（NIST）文本摘要比赛用的小型数据集，包含 500 篇短文本，每篇文本包含 4 个由专家撰写的标准摘要，该数据集仅用于评测；英语 Gigaword 包含约 950 万篇新闻，取每篇新闻的首句作为源文本，新闻标题作为标准摘要[9]，属于单句摘要数据集；LCSTS 是大规模中文短文本摘要数据集，数据采集自新浪微博，短文本为 80~140 字，摘要为 10~30 字[70]。

享，CNN 可以提取文本局部特征，如 *n*-gram；自注意力是 Vaswani 等[16]提出的一种注意力机制，句子中的每个词项都和该句中的所有词项进行注意力计算，其目的是学习句子内部的词依赖关系，捕获句子中的内部结构。经过第二遍编码，编码器输出的词项隐层包含了更多的短语结构信息和句子结构信息。

3.1.2　*层级编码*

实验证明，对于 Seq2Seq 模型来说，源文本越长处理难度越大，主要原因在于神经网络的记忆能力有限，即使有注意力机制，也很难联合较远的输

入做出判断。因此处理长文本的一个思路是将其拆分成区块（如句子、段落、语篇等），对区块和全文分别进行编码，再用层级注意力（hierarchical attention）计算语义向量，从而缓解记忆压力，并尝试在语义向量中融入文本的结构特征。

1）句子

Nallapati 等[17]首次用 Seq2Seq 模型处理长文本摘要，提出层级注意力模型，其基本思想是词的重要性会受到其所在句子的重要性的影响，因此在计算词项注意力分布时要考虑句子的注意力分布。该模型在编码器端使用了两个双向 RNN，一个是词级，一个是句子级。词级 RNN 对词项进行编码，生成词项隐层，每遇到句末标志，就将当前隐层和相应句子的位置信息相结合，作为句子的向量表示赋给句子级 RNN；句子级 RNN 对多个句子向量进行编码，生成句子隐层。分别计算词项注意力分布 α_i^w 和句子注意力分布 α_s^s，之后用句子的注意力分布将词项的注意力分布更新为 α_i，

$$\alpha_i = \frac{\alpha_i^w \alpha_{s(i)}^s}{\sum_{k=1}^{T} \alpha_k^w \alpha_{s(k)}^s} \tag{14}$$

最后用公式(6)计算出语义向量 c_t。

2）段落

和句子相比，段落可以更好地体现文本的结构信息。Celikyilmaz 等[18]提出深度通讯代理（deep communicating agents）模型，该模型设置了多个“代理”，每个代理负责编码一个段落。假设文本由 M 个段落组成，代理 a 负责编码段落 a，a=1,…,M。每个代理使用两个 LSTM 对段落进行编码：先用双向单层 LSTM 对段落中的每个词项进行编码，获得词项隐层 $h_i^{(1)}$；将 $h_i^{(1)}$ 输入双向多层 LSTM，作为第一层的初始值；第 l 层的词项隐层 $h_{m,l}^{(l)}$ 的计算公式为

$$h_i^{(l+1)} = b\mathrm{LSTM}(f(h_i^{(l)}, z^{(l)}), \overrightarrow{h}_{i-1}^{(l+1)}, \overleftarrow{h}_{i+1}^{(l+1)}) \tag{15}$$

其中，$z^{(l)}$ 代表来自其他代理的第 l 层输出的最后一个隐层 $h_{m,i}^{(L)}$ 的均值，

$$z^{(l)} = \frac{1}{M-1} \sum_{m \neq a} h_{m,l}^{(l)} \tag{16}$$

代理之间通过 $z^{(l)}$ 实现了“通讯”，这些信息可以让模型了解段落之间的关系，进而理解文本的全局结构。

用公式(15)计算出的双向多层 LSTM 的最后一层的词项隐层 $h_{a,i}^{(L)}$ 作为每个段落中词项的隐层向量，之后计算出每个段落中的词注意力分布 $\alpha_{a,i}^t$，将其和词项隐层 $h_{a,i}^{(L)}$ 加权后得到每个段落的语义向量 c_a^t。和层级注意力不同的是，该模型利用 c_a^t 计算代理注意力分布 β_a^t，

$$\beta_a^t = \mathrm{softmax}(\mathrm{score}(c_a^t, s_t)) \tag{17}$$

最后将 c_a^t 和 β_a^t 加权计算出语义向量 c_t。

3）语篇

论文摘要在很大程度上依赖于语篇（discourse）结构，即通过总结各语篇的要点以形成摘要。例如，学术论文的典型语篇包括问题描述、研究方法、结论等。受此启发，Cohan 等[19]提出语篇感知注意力（discourse-aware attention）模型，用学术论文作为语料，在预处理阶段利用论文的一级标题将其拆分为 M 个语篇，每个语篇包含 N 个词。用双向单层 LSTM 分别对每个语篇中的词项进行编码，得到第 j 个语篇中第 i 个词项的隐层 $h_{j,i}$，各个语篇参数共享。连接每个语篇的正向和反向隐层作为语篇隐层 $h_j = f([\overrightarrow{h}_j; \overleftarrow{h}_j])$。该模型在计算词注意力分布时，用语篇注意力分布 β_j^t 对词项注意力得分进行加权，

$$a_{j,i}^t = \mathrm{softmax}(\beta_j^t \mathrm{score}(h_{j,i}, s_{t-1})) \tag{18}$$

再将 $a_{j,i}^t$ 和 $h_{j,i}$ 加权计算出语义向量 c_t。

鉴于层级注意力模型的时间复杂度偏高，Ling 等[20]提出了“由粗到精（coarse to fine，C2F）”注意力模型。模型对区块注意力分布进行“硬注意（hard attention）”，选出注意力权重最大的区块 J，利用该区块中的词注意力分布直接计算语义向量 $c_t = \sum_i \alpha_{J,i} h_{J,i}$。C2F 模型虽然有效降低了层级注意力计算的时间复杂度，但“硬注意”会丢失其他区块的信息，致使该模型在 CNN/Daily Mail 测试集上的表现并不理想（表 1）。

3.1.3 抽取辅助

抽取式摘要研究中形成了包括基于词频统计、机器学习、主题以及图等在内的多种抽取方法，此外还利用外部资源帮助识别文本中的命名实体、词性等[3]。尽管抽取式摘要本身存在局限，但依然具有借鉴价值。一些学者利用抽取方法辅助编码，试图使机器注意到源文本的重点，这一点上和选择性编码类似，区别在于前者有显式抽取过程，后者没有。

1）句子抽取

一些学者在生成式模型中加入了句子抽取技术，具体做法大致可以分为两类。

一类是将抽取和生成独立为两个阶段，即先抽取句子，再将这些句子输入生成式模型。王帅等[21]用 TextRank 算法[22]从源文本中抽取出重要的句子，

再用基于RNN的Seq2Seq模型生成摘要，这种做法显然会丢失源文本中的信息。Xie等[23]利用WordNet识别和抽取重要的句子，在编码阶段用两个双向LSTM对源文本和抽取出的句子分别进行编码，得到它们的语义向量c_t^{doc}和c_t^{extr}，利用二者计算出语义向量c_t，

$$c_t = g_t c_t^{\text{doc}} + (1 - g_t) c_t^{\text{extr}} \text{ where } g_t = \text{MLP}([c_t^{\text{doc}}; c_t^{\text{extr}}]) \tag{19}$$

该模型的优点在于既关注重要性高的句子，也兼顾重要性相对较低的句子。

Chen等[24]构造了两个Seq2Seq模型，一个是基于CNN-LSTM的句子抽取器（extractor），另一个是基于RNN的生成器（abstractor）。模型的创新之处在于利用强化学习方法训练句子抽取器，即用生成器生成的摘要和标准摘要进行比较得到的ROUGE值作为回报，反馈给抽取器，修正其参数，以期下一轮的抽取更准确。

另一类是将抽取和生成融合在一个Seq2Seq模型中。Chen等[25]用两个双向GRU分别计算每个词项和句子的隐层，用sigmoid函数计算句子的重要性得分，基于给定阈值确定句子保留与否。对于所有保留下来的句子，用层级注意力计算语义向量。Hsu等[26]构建了一个统一模型，可同时用于抽取式和生成式摘要，并构造了不一致损失函数（inconsistency loss），用于惩罚词注意力分布和句子注意力分布的不一致性，力图达到抽取和生成的一致性。

Tan等[27]以句子隐层h_j和解码器隐层s_t作为图的结点，利用TextRank算法计算句子的重要性得分f_j^t。该工作的创新之处是在计算句子注意力分布时不再利用h_j，而是利用句子得分f_j^t，

$$\alpha_j^t = \frac{\max(f_j^t - f_j^{t-1}, 0)}{\sum_m (\max(f_m^t - f_m^{t-1}, 0))} \tag{20}$$

可以看出，一个句子只有在当前时间步的重要性大于上一时间步的重要性时才会被关注，这一算法兼顾了句子的重要性和新颖性。

2）关键词抽取

一些学者用TextRank算法抽取源文本中的关键词，得到关键词序列$\{k_1, \cdots, k_n\}$。Li等[28]用双向LSTM对关键词序列进行编码，计算出关键词的隐层向量，通过连接正向隐层和反向隐层得到关键词序列的隐层向量$kw = [\overrightarrow{h}_n; \overleftarrow{h}_1]$；Jiang等[29]考虑到关键词是相对独立的，不需要考虑上下文关系，直接将关键词序列中的词向量相加得到关键词向量kw = $\sum_{i=1}^{n} k_i$。两篇工作均将kw添加到注意力计算中，

$$e_{ti} = \text{score}(h_i, s_t, \text{kw}) \tag{21}$$

可以让模型更多地关注源文本中的关键信息。

侯丽微等[30]用注意力机制计算出关键词语义向量c_t^k，将其和编码器语义向量c_t^e、解码器语义向量c_t^d结合后共同推导下一个词项。该模型在NLPCC2017的中文单文档摘要评测数据集上取得了领先成绩。

3）事实抽取

Cao等[31]研究发现基于神经网络生成的摘要有30%存在事实捏造现象，例如，谓词与其主语或宾语之间不匹配，因此提出将源文本中的事实描述显式地编码到模型中。该模型利用开源信息抽取（OpenIE）和依存句法分析（dependency parser）工具抽取源文本中的事实关系，并表示为三元组的形式，如(主语+谓语+宾语)；再利用两个双向GRU对源文本以及抽取出的事实分别编码，得到它们的语义向量c_t^{doc}和c_t^{fact}，之后的工作和Xie等[23]类似。

3.1.4　卷积编码

卷积编码即对输入序列进行卷积操作以得出文本表示。Rush等[9]曾使用基于TDNN的卷积编码器计算语义向量，鉴于TDNN不擅长处理时间序列，而且模型缺少注意力机制，实验效果并不理想；来自同一团队的Chopra等[32]改进了Rush等[9]的工作，将输入序列中词项的位置信息嵌入到词向量中，并加入注意力机制，在一定程度上提升了模型的表现。之后的相关工作大多采用更擅长处理时间序列的RNN进行建模。2017年，Gehring等[33]提出基于CNN的卷积序列到序列（convolutional sequence to sequence，ConvS2S）模型，在机器翻译和文本摘要任务中均表现出色，引起学界的关注。

ConvS2S模型的编码器和解码器均是多层CNN，编码器对输入序列做多层卷积，解码器在每一层都做注意力计算，即多步注意力（multi-step attention）。模型首先对输入序列中的词项做位置嵌入，将每个词向量x_i和其绝对位置向量p_i相加作为模型的输入，即$e=(x_1+p_1, \cdots, x_T+p_T)$，位置嵌入给原本不擅长处理时间序列的CNN带来一些“位置感”。对于第l层，用大小为k的卷积核$v^l \in \mathbb{R}^{kd}$对上一层的输出做一维卷积，得到每一层的输出$g^l \in \mathbb{R}^{2d}$，其中d代表词向量的维度；将g^l转换为矩阵$G^l = [G_1 G_2]$，其中$G_1, G_2 \epsilon \mathbb{R}^d$，用门控线性单元（gated linear unit，GLU）对G^l做非线性变换；为支

持深度卷积网络，在每一层的输出中还增加了残差连接，最后得到第 l+1 层的隐层 h_i^{l+1}，

$$h_i^{l+1} = G_1^l \otimes \sigma(G_2^l) + h_i^l \tag{22}$$

多步注意力的具体实现为：首先将解码器第 l 层的隐层 s_t^l 和上一步输出的词项 y_{t-1} 结合得到 d_t^l，之后利用 d_t^l 和编码器最后一层的隐层 h_i^L 计算解码器第 l 层的注意力 α_{ti}^l，

$$\alpha_{ti}^l = \frac{\exp(d_t^l \cdot h_i^L)}{\sum_{k=1}^{T} exp(d_t^l \cdot h_k^L)} \tag{23}$$

最后用 α_{ti}^l 加权 h_i^L 和 e_i 得到第 l 层的语义向量 c_t^l，

$$c_t^l = \sum_{i=1}^{T} \alpha_{ti}^l (h_i^L + e_i) \tag{24}$$

3.1.5 其 他

在大部分模型中，输入序列中的每个词都是词向量形式（如 one-hot、word2vec 等），仅含有少量语义信息，一些学者[17,34]将富特征（feature-rich）嵌入词向量中，试图通过增加输入词项的特征来提高模型的学习能力。这些特征包括命名实体标签（NER）、词性（POS）、词频（TF）和逆文档频率（IDF）等。实验结果显示，这些特征并没有提升模型的表现。Song 等[35]认为短语比词更能表达完整的意思，因此提出基于短语的 LSTM-CNN 编码方案，将模型输入和输出的基本单位都替换成短语。在预处理阶段用短语抽取方法将文本分解成短语序列，具体包括主语短语、关系短语和宾语短语等。编码阶段先用 CNN 将短语编码成向量，再用 LSTM 计算每个短语向量的隐层。该方法保持了生成摘要的语法完整性，但短语的组合可能会限制摘要的新颖性。

3.2 解 码

解码器读取语义向量并输出目标序列，相当于人在理解文本后开始编写摘要。解码过程主要存在以下问题：①当某个词不存在于词汇表中时，便无法生成；②解码器可能会重复关注到源文本的某些部分，导致摘要也产生重复；③解码器变量有限，无法将高级语法或结构信息模型化。围绕上述问题，学界提出多种改进方法。

3.2.1 拷贝机制（copy mechanism）

在基础模型中，解码器基于一个预设的词汇表 V 生成词项，V 一般由训练集中的高频词汇组成。对于未登录词（out of vocabulary，OOV）使用 <UNK> 标签来替代。当源文本中存在 <UNK> 时，生成的摘要中也可能会出现 <UNK>，这显然是无法接受的。针对该问题，Gulcehre 等[36]提出指针 softmax（pointer softmax）模型，基本思路是将源文本中的 OOV 在必要时直接拷贝到摘要中。该模型要解决两个问题：在解码的每个时间步是选择从词汇表中生成一个词还是从源文本中拷贝一个词；若选择拷贝，从源文本的什么位置拷贝。模型在解码器中使用了两个 softmax 输出层：词汇表 softmax 和位置 softmax。前者输出的是要生成的词项在词汇表中的分布 P_{vocab}；后者是一个指针网络[37]，输出要拷贝的词项在输入序列中的位置分布。由于词项的注意力分布和位置分布基本相同，模型直接将注意力分布 α_{ti} 用作位置分布，通过参数复用，降低了模型的复杂度。模型设置了一个基于多层感知机的开关网络，将语义向量 c_t 和解码器隐层 s_t 输入该网络，得到一个开关变量 switch_t，

$$\text{switch}_t = \sigma(c_t, s_t) \tag{25}$$

在解码的每个时间步，若 switch_t=1，用词汇表 softmax 从词汇表中生成一个词；若 switch_t=0，则用位置 softmax 输出一个位置，并将输入序列中和该位置对应的词项拷贝到输出序列。Nallapati 等[17]改进了 Gulcehre 等[36]的工作，在计算 switch_t 时加入了解码器上一时间步生成或拷贝的词项 y_{t-1}，在一定程度上提高了开关网络的准确性。

然而 switch_t 是“硬开关”，非 0 即 1，不利于提升机器学习的精度。针对这一问题，See 等[38]提出指针-生成器网络（pointer-generator network），计算出生成概率 $p_{\text{gen}} \in (0,1)$，

$$p_{\text{gen}} = \sigma(c_t, s_t, y_{t-1}) \tag{26}$$

拷贝概率则为 $1-p_{\text{gen}}$。对于每篇源文本，模型在解码时将该文本中的词动态地加入词汇表中得到扩展词汇表，扩展后的词汇分布为

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i = w} \alpha_{ti} \tag{27}$$

当词项 w 不在词汇表中时，$P_{\text{vocab}}(w)$=0；当 w 不在源文本中时，$\sum_{i:w_i = w} \alpha_{ti} = 0$。$p_{\text{gen}}$ 相当于“软开关”，不仅可以让模型学习是从词汇分布 P_{vocab} 中生成一个词还是从输入序列中拷贝一个词，而且可以提高词汇表中罕见词项的训练精度。

Zhou 等[39]在 Gigaword 训练集上统计发现，文本摘要中生成的词占 42.5%，其余的均由拷贝所得，且连续两个词以上的拷贝约占 1/3。在之前的拷贝机制中，每次拷贝都有一个决策过程——拷贝还是

生成，如果要连续拷贝 3 个词，机器就要做 3 次决策。因此，Zhou 等[39]提出序列拷贝网络，其基本思想是如果机器决定要拷贝，则直接拷贝一个子序列，如 3 个词，从而减少决策次数，同时降低了拷贝机制在连续拷贝过程中出错的概率。

Gu 等[40]提出的 CopyNet 在模型结构上有别于先前的拷贝机制[36,38]，没有使用开关网络和指针网络，在解码时基于生成模式和拷贝模式的混合概率预测单词。模型构造了一个词汇表 X，只收录存在于输入序列中的词，扩展后的词汇表为 $V \cup X \cup <\text{UNK}>$，由于 X 中可能包含不存在于 V 中的词，这部分词将用于拷贝。在输出每个目标单词时分别计算生成模式和拷贝模式的概率，并相加得到混合概率，

$$p(y_t|s_t,y_{t-1},c_t,H) = p_{\text{gen}}(y_t|s_t,y_{t-1},c_t,H) + p_{\text{copy}}(y_t|s_t,y_{t-1},c_t,H) \tag{28}$$

其中，H 表示由编码器生成的词项隐层 $\{h'_1,\cdots,h'_T\}$ 构成的矩阵，h'_t 既包含词项语义信息又包含位置信息。CopyNet 在生成模式下读取语义信息，在拷贝模式下则读取位置信息。由于 H 包含了位置信息，CopyNet 在网络结构上更加简单，不需要开关和指针；但也正是因为 H 的特殊性，限制了 CopyNet 的通用性。

3.2.2　重复控制

由于注意力机制忽略了输入序列和输出序列之间的对齐关系[41]，因此解码器可能会重复关注到输入序列的某些部分，导致输出序列也产生重复。针对这一问题，有以下几种解决方案。

1）覆盖机制（coverage mechanism）

覆盖机制由 Tu 等[41]提出，最初用于解决神经机器翻译模型中存在的过度翻译和翻译不足的问题。See 等[38]将覆盖机制应用于生成式摘要模型，其基本思路是在计算注意力分布时不再关注先前已经关注过的部分。具体做法是在计算时间步 t 的注意力分布时考虑一个覆盖向量 coverage_t，

$$\alpha_{ti} = \text{softmax}(\text{score}(h_i,s_t,\text{coverage}_{ti})) \tag{29}$$

该向量是时间步 t 之前所有时间步计算出的注意力分布的累加和，

$$\text{coverage}_t = \sum_{t'=0}^{t-1} \alpha_{t'} \tag{30}$$

记录了源文本中解码器已经关注过的部分。为确保覆盖机制的有效性，模型还定义了一个覆盖损失，

$$\text{covloss}_t = \sum_i \min(\alpha_{ti},\text{coverage}_{ti}) \tag{31}$$

用于惩罚重复关注到同一位置的情况，并将该损失添加到损失函数[公式(5)]中。

2）时间注意力（temporal attention）和解码器内注意力（intra-decoder attention）

时间注意力由 Sankaran 等[42]提出，最初用于应对机器翻译中注意力缺乏问题。Nallapati 等[17]发现它也能应对重复摘要问题。时间注意力机制在思想上和覆盖机制类似，都是利用先前时间步的注意力来影响当前注意力的计算，以防止生成重复内容。具体计算[43]是将公式(8)计算出的注意力得分 e_{ti} 在时间维度上做归一化处理，得到时间注意力得分 e_{ti}^{temporal}，

$$e_{ti}^{\text{temporal}} = \begin{cases} \exp(e_{ti}) & t = 1 \\ \dfrac{\exp(e_{ti})}{\sum_{j=1}^{t-1}\exp(e_{ji})} & \text{其他} \end{cases} \tag{32}$$

可以看出，先前关注过的词项的时间注意力得分会降低，未关注过的会提高，这样解码器就不会重复关注到相同的词项。

和覆盖机制相比，时间注意力直接用先前注意力得分调整当前注意力得分，不需要损失函数，使模型结构更加简单。虽然时间注意力确保了输入序列不会被重复关注，但解码器仍可能会根据自身已经生成的词项来生成重复内容。为避免这种情况，Paulus 等[43]提出解码器内注意力，用解码器已生成的词项隐层计算出解码器语义向量 c_t^d，让解码器在生成每个词项时同时考虑 c_t^e 和 c_t^d，该工作在重复控制上取得了不错的效果。

3.2.3　融入额外信息

Li 等[44]通过对解码过程的研究发现，用于生成摘要的变量很有限，无法将高级语法或结构信息模型化。针对该问题，一些学者将额外信息融入解码过程，试图提高生成句子的质量。Li 等[44]对 CNN 新闻中的人工摘要进行了语言学分析，发现人工摘要存在着某些固定的句子结构，如“什么”、“发生了什么”、“谁做了什么”等。为将这些潜在的句子结构信息融入解码过程中，Li 等[44]提出深度循环生成解码器（deep recurrent generative decoder，DRGD），在解码器端加入变分自编码器（variational auto-encoder，VAE）。VAE 最初由 Kingma 等[45]提出，对潜在随机变量具有较强的建模能力，在句子生成和图像生成方面取得了一定的成效[46-47]。VAE 自身也是一个基于神经网络的编码器-解码器模型，将源文本和对应的摘要输入 VAE 模型进行训练（实验阶段和基础模型共同训练），可以推导出含有潜

在结构信息的向量q_t。DRGD使用了双层GRU，并将VAE作为解码器的一个子模块，通过第二层隐层$s_t^{(2)}$和潜在结构向量q_t计算出解码器隐层$s_t = f(s_t^{(2)}, q_t)$，用含有潜在结构信息的解码器隐层s_t来生成句子，会使句子也具有一定的结构。

Kryściński等[48]在解码器端融入了语言模型，试图在提高摘要流畅性的同时融入特定领域的语言风格。该工作基于Paulus等[43]的工作，创新之处是通过特定语料集预训练语言模型，从而获得特定领域的语言风格。在解码时，将上一时间步生成的词项y_{t-1}分别输入语言模型和解码器中，得到语言模型的隐层h_t^{lm}和语义向量c_t，将两者结合后共同推导下一个词项。

Amplayo等[49]在解码器端加入实体到主题（Entity2Topic）模块，将源文本的主题信息融入解码过程。具体做法是，利用维基百科抽取源文本中的实体，将抽取的实体向量和编码器生成的文本向量进行比较，选出最重要的E个实体，并利用其计算主题向量t。连接主题向量t和解码器隐层s_t得到新的解码器隐层$s'_t = [s_t; t]$。

3.2.4 束搜索（beam search）

在训练阶段，解码器生成的每个词项都有标准摘要作参考，并将误差反向传播以修正模型参数，因此一般采用贪婪算法取词汇分布的概率最大值作为输出词项。但在测试阶段，没有标准摘要作参考，这时概率最大的词项未必是最好的选择，研究表明，用贪婪算法生成的句子可读性较差[9]。束搜索算法是解决上述问题的一种手段，已被广泛运用于多个摘要模型[9,38]，具体算法是，给定一个束宽（beam width）B，在解码的每个时间步都保留词汇分布中概率最大的B个词项作为候选词项，从第二个时间步开始，会产生$B \times B$个候选分支，依然只保留概率最大的前B个分支，依次进行下去，直至遇到终止条件。最后得到B个候选序列，选择概率最大者作为最终结果。束搜索的问题是缺乏多样性，即B个候选序列区别不大，因此Cibils等[50]提出多样性束搜索（diverse beam search），将束宽B等分为若干个组，在解码时每个组依次进行束搜索，并构造一个差异函数来度量当前组的候选序列和先前组生成的序列之间的差异，通过惩罚差异小的分支以增加组之间生成序列的多样性。

3.3 训 练

从公式(5)可以看出，基础模型的训练过程属于词级训练，即逐个最大化每个词项的条件概率$p(y_t|y_{<t}, X)$，可能会丢失全局信息。对此，Ayana等[51]提出最小风险训练（minimum risk training，MRT）策略，属于序列级训练[52]，即通过最小化生成摘要y'和标准摘要y的距离$\Delta(y', y)$来估计模型参数，距离$\Delta(y', y)$利用ROUGE值计算而来（如负ROUGE值），损失函数为

$$L_{\mathrm{MRT}}(\theta) = \sum_{(x,y)\in D} \sum_{y'\in Y(x;\theta)} p(y'|x;\theta)\Delta(y', y) \quad (33)$$

其中，$Y(x;\theta)$代表训练集中的每个x可能生成的摘要的集合。可以看出，最小化L_{MRT}可以使模型生成的摘要更加接近标准摘要。

MRT策略依然属于有监督学习，主要存在两个不足[43]：一是曝光偏差（exposure bias）[52]，由于在训练过程中有真值（ground truth）参考，而测试过程中没有，因此测试时会产生误差累积；二是最大似然于真值并非摘要质量评价的唯一标准。针对以上问题，一些学者使用自我评判（self-critical）[53]——一种强化学习（reinforcement learning，RL）中的策略梯度训练算法来训练模型。模型的训练目标不再似然于真值，而是优化用户定义的度量标准（如ROUGE）。Paulus等[43]让模型在每次训练迭代时分别产生两个输出序列：用贪婪算法得到的$\hat{y}$和经随机采样得到的y^s。用回报函数$r(\cdot)$返回参数序列和标准摘要y相比较得到的ROUGE分数。基于强化学习的损失函数为

$$L_{\mathrm{RL}}(\theta) = \sum_{(x,y)\in D} (r(\hat{y}) - r(y^s))\log p(y^s|x;\theta) \quad (34)$$

可以看出，最小化L_{RL}相当于最大化y^s的条件似然，从而增加模型的预期回报。然而，Paulus等[43]发现强化学习方法虽然可以提高模型的ROUGE得分，但生成的摘要在可读性上不如最大似然方法，因此将公式(34)和公式(5)结合，得到混合目标函数，

$$L_{\mathrm{MIXED}} = \gamma L_{\mathrm{RL}} + (1-\gamma)L_{\mathrm{MLE}} \quad (35)$$

其中，γ为超参数，用于权衡两个目标的比重。

Li等[54]借鉴了深度强化学习中的Actor-Critic算法[55]，用摘要生成模型作为actor，将最大似然估计器和全局摘要质量估计器相结合作为critic。全局摘要质量估计器是一个回报函数，可以将生成的摘要和标准摘要区分开。actor生成摘要，critic对其打分，actor根据打分调整策略，通过迭代实现回报最大化，以此训练出模型的参数。

Shi等[56]发现摘要任务在训练过程中存在样本不均衡问题，如训练集中各个样本的训练难度不同，每个句子中各个词的训练难度也有区别。这些区别可以通过输出的词项概率$P(w)$体现出来：$P(w)$

接近 1，代表系统可以较为肯定地预测该词；$P(w)$ 接近 0，代表该词的训练难度较大。focal loss 是 Lin 等[57]提出的一种损失函数，基本思想是希望困难样本对损失的贡献变大，使网络更倾向于从这些样本上学习。因此，Shi 等[56]利用 focal loss 为每个词的损失分配权重 $\lambda(1-P(w))^{\gamma}$（λ 和 γ 为超参数），让模型可以更多地关注 $P(w)$ 较小的词，从而提升模型的整体表现。

3.4　其　他

基于 Seq2Seq 模型的生成式摘要的研究还包括以下方面：Hua 等[58]研究了摘要模型的跨领域适应性，发现利用某个领域的语料训练出来的模型可以捕捉另一领域中文本的重要信息，但无法生成和目标领域相符的语言风格，模型的可移植性有待进一步研究；Fan 等[59]研究了摘要模型的用户可控性，包括自定义摘要的长度、聚焦特定的实体、选择喜好的风格，以及仅摘取未读内容；Zhang 等[60]和 Chu 等[61]设计了面向多文档的摘要模型；Nema 等[62]、Hasselqvist 等[63]和 Baumel 等[64]研究了基于查询的摘要任务；Pasunuru 等[65-66]和 Guo 等[67]提出了多任务学习框架，即用一个 Seq2Seq 模型处理多种不同的任务，包括问题生成、蕴涵（entailment）生成和文本摘要，并利用问题生成和蕴含生成的相关信息提升文本摘要的质量。

4　讨　论

表 1 以 Rush 等[9]的工作为基线整理了短文本摘要模型的基本信息和评测结果，表 2 以 Nallapati 等[17]的工作为基线整理了长文本摘要模型的基本信息和评测结果。从整体上看，在基于 Seq2Seq 模型的生成式摘要研究中，2015—2016 年的工作主要集中于短文本摘要，2017—2018 年逐渐转向长文本，大部分研究的评测结果相较于基线模型都有明显提升。结合第 3 节的梳理以及表中的数据可以进行以下主题的讨论。

1）模型框架

由于基础模型中的 RNN 存在长期依赖问题（long term dependency），难以学习到相隔较远的信息，因此绝大部分模型都使用 LSTM 或 GRU 取代之。LSTM 引入了门控概念，可以学习记忆单元的记忆/遗忘、输入程度和输出程度，从而让机器知道何时应记住或抛弃某些信息。GRU 则使用门控网络来学习新的输入与先前记忆的组合方式以及先前记忆的保留程度。尽管 LSTM 和 GRU 原理不同，但都可以学习长期依赖信息，在很多任务上的表现都优于 RNN。和 LSTM 相比，GRU 参数较少，更容易训练，但在训练数据较多的情况下，表达能力更强的 LSTM 或许会有更好的表现。基于该考虑，超过半数的短文本摘要任务使用了 GRU 建模（见表 1），长文本摘要任务中超过 80% 的研究者使用了 LSTM 建模（见表 2）。

在使用 LSTM/GRU 的模型中，绝大部分都采用了单层网络结构，只有 Celikyilmaz 等[18]提出的深度通讯代理模型在编码器端使用了多层 LSTM，该模型在 CNN/Daily Mail 测试集上综合表现第一。表 1 中有两项研究[33,68]使用了 ConvS2S 模型，ConvS2S 模型默认是多层结构，两个模型在相应测试集上的表现均位于前列。Gehring 等[33]对其实验结果的解释是，相较于单层 RNN，ConvS2S 模型对原文本的层级表示，使其较易于发现序列中的复合结构。综合表 1 与表 2 可见，多层神经网络在摘要任务中的表现整体上优于单层。究其机理，以编码为例，单层网络仅将词向量编码为包含上下文关系的隐层；而多层网络可以将低层隐层编码为更加抽象的高层隐层，相当于源文本的抽象表示。多层网络的层层抽象，可以构建出源文本的分布式特征，还可以提供更短的路径来捕捉长期依赖关系[33]。但是，多层网络（尤其是多层循环神经网络）需要更大量的训练数据才能达到较好的学习精度，而且需要更大的计算开销，这可能是绝大部分使用 LSTM/GRU 的摘要模型采用单层结构的原因之一。鉴于 CNN 可以并行处理并利用 GPU 加速[71]，在训练速度上优于 RNN，在多层网络模型中具有良好的应用前景。

2）注意力机制

注意力机制是 Seq2Seq 模型的核心组件，是连接编码器和解码器的桥梁，被广泛运用于各种自然语言处理任务。早期的注意力机制只用于计算输入序列和输出序列的匹配程度，随着研究的深入，多种注意力被提出，并可实现更多的功能。例如，在拷贝机制中用注意力分布表示输入序列中词项的位置信息[36,38]，在覆盖机制中用于避免重复关注[38]，时间注意力用于重复控制[17,43]，解码器内注意力可以关注已生成的序列[43]，层级注意力可以捕捉词项和区块的关系[17-19]，自注意力可以学习句子的内部结构[15]，多步注意力使解码器在每一层都能关注到先前的信息[33,68]，等等。其中使用了内注意力、多步注意力和自注意力的模型在评测结果中均有突出

表 2 长文本摘要模型基本信息及评测结果

作者	模型	框架	CNN/Daily Mail		
			ROUGE-1	ROUGE-2	ROUGE-L
Nallapati 等[17](2016)	基线	RNN → RNN	32.49	11.84	29.47
Nallapati 等[17](2016)	层级注意力	RNN → RNN	32.75	12.21	29.01
Nallapati 等[17](2016)	时间注意力	RNN → RNN	35.46	13.30	32.65
Ling 等[20](2017)	由粗到精	LSTM → LSTM	31.10	15.40	28.80
See 等[38](2017)	指针生成器网络+覆盖机制	LSTM → LSTM	39.53	17.28	36.38
Paulus 等[43](2017)	内注意力+强化学习	LSTM → LSTM	**41.16**	15.75	**39.08**
Tan 等[27](2017)	基于图的注意力机制	LSTM → LSTM	38.10	13.90	34.00
Chen 等[25](2018)	抽取句子+层级注意力	GRU → GRU	35.80	13.60	33.40
Li 等[28](2018)	抽取关键词	LSTM → LSTM	38.95	17.12	35.68
Chen 等[24](2018)	抽取器+生成器+强化学习	抽取器 → 生成器	**40.88**	17.80	**38.54**
Hsu 等[26](2018)	抽取生成统一模型+不一致损失	抽取器 → 生成器	40.86	**17.97**	37.13
Xie 等[23](2018)	抽取句子+指针生成器网络+覆盖机制	LSTM → LSTM	39.32	17.15	36.02
Cibils 等[50](2018)	指针生成器网络+多样性束搜索	LSTM → LSTM	40.19	17.09	36.63
Celikyilmaz 等[18](2018)	深度通讯代理+指针生成器网络+强化学习	多层 LSTM → LSTM	**41.69**	**19.47**	**37.92**
Song 等[35](2019)	基于短语的 LSTM-CNN	CNN + LSTM → LSTM	34.90	17.80	-
Pasunuru 等[66](2018)	多回报(ROUGE+显著+蕴含)	LSTM → LSTM	40.43	**18.00**	37.10
Guo 等[67](2018)	多任务(问题生成+蕴含生成)	LSTM → LSTM	39.81	17.64	36.54
Kryściński 等[48](2018)	内部注意力+语言模型+强化学习	LSTM → LSTM + LM	40.19	17.38	37.52

注：CNN/Daily Mail 包含了来自美国有线电视新闻网和英国《每日邮报》的约 30 万篇新闻语料，每篇源文本包含平均 766 个词和 29.74 个句子，对应的标准摘要包含平均 53 个词和 3.72 个句子[17]。

表现。可以看出，注意力机制在自然语言处理任务中正扮演着重要的角色，模型的表现和注意力机制的设计密切相关。与选择性编码、“再读”等几乎没有提高摘要质量的技术相比，注意力机制在多项研究中已被证实可以切实提高模型的表现[72]。作为摘要任务本质上是从长文本到短文本的映射，为模型设计一个契合的注意力机制以学习映射关系，也许比改进编/解码器更有效率。此外，在 Vaswani 等[16]的机器翻译工作中，注意力机制甚至替代 RNN 或 CNN 完成了编/解码工作，进一步证明注意力机制的应用前景。

3）长文本处理

由于摘要长度往往是固定的，那么源文本越长，重要信息就越难把握，因此长文本摘要模型在设计上和短文本模型有显著的区别。从表 2 可以看出，长文本摘要模型大多利用抽取辅助或分区块等予以处理，两种方法在操作上截然不同，但基本思想都是分而治之。其中层级注意力专为处理长文本而设计，但相关模型的评测结果并不理想。例如，Nallapati 等[17]将层级注意力模型和基线模型做了对比实验，评测结果没有显著差异，甚至层级注意力模型在 ROUGE-L 指标上还略有降低。在使用层级注意力的模型中只有深度通讯代理表现优异，该模型注重区块间的“通讯”，这些信息所反映的文本的全局结构对长文本摘要来说无疑是十分重要的。Celikyilmaz 等[18]在实验阶段对“代理”之间是否通讯分别做了测试，实验结果表明“通讯代理”在评测结果上具有明显优势。此外，在长文本模型中有超过 1/3 使用了抽取辅助，这些工作的评测结果同样参差不齐，说明抽取和生成的结合方式对于能否发挥出 1 加 1 大于 2 的效用至关重要。其中 Chen 等[24]和 Hsu 等[26]的工作在评测结果中取得了不错的表现，两篇工作的思路均是协调抽取器和生成器的关系，值得后续工作借鉴。

值得注意的是，Paulus 等[43]的工作没有采用抽取辅助或分区块处理，却在评测结果中领先于绝大部分长文本模型，其中 ROUGE-L 得分位居第一。Paulus 等[43]对此的解释是，内注意力和混合训练目标可以兼顾摘要的可读性和 ROUGE 评分，而且非常适合处理长文本任务。该工作为长文本摘要的后续研究提供了新的思路。

4）句子质量

句子的质量通常取决于句子内部词项间的依赖关系，具体来说是短语结构和句子结构。深度循环

生成解码器在解码时融入由 VAE 提取的句子结构信息[44]；CNN 编码器可以利用固定大小的卷积核提取词项的 *n*-gram 特征[15]，相当于短语结构；基于短语的 LSTM-CNN 模型直接用短语代替词项作为输入和输出的基本单位[35]。以上工作从不同角度出发来解决词项依赖问题，在评测结果中均有不错的表现。其中，使用自注意力机制的全局编码模型[15]在 LCSTS 和英语 Gigaword 两个测试集上的 ROUGE-2 得分均位于领先水平（见表 1），ROUGE-2 反映了自动摘要和标准摘要间连续两个词项的重叠度，从一定程度上可以反映词项间的依赖程度。该机制在机器翻译任务上也取得了令人瞩目的成绩[16]，后续工作可以继续挖掘该机制的潜力。

拷贝机制借鉴了人在处理难理解的语言片段时的解决方式——直接照搬，这是保证句子质量的有效途径。该机制将生成和拷贝相结合，一方面保持了模型的生成能力，在输出端生成一些和源文本措辞不同的摘要；另一方面可以从源文本中拷贝适合的片段到输出序列中，保证了字面的完整。多项研究证实拷贝机制在处理 OOV 问题上（尤其是命名实体和新生词汇）是非常有效的，其中指针-生成器的网络应用最为广泛[18,23,38,50]。

5）评测与训练

从表 1 和表 2 可以看出，有多个模型使用面向 ROUGE 的方法进行训练，这些模型在相应测试集上的 ROUGE 得分都不出意料地排在前列。然而 ROUGE 评价方法本身存在一定的局限性。ROUGE 的提出者 Lin[73]认为该评价方法和人工评价的一致性仍有待提高，Paulus 等[43]也认为 ROUGE 指标不应该成为摘要（尤其是长文本摘要）评价的唯一标准。多篇工作[18,43,48]在模型评估阶段分别采用 ROUGE 和人工评价方法，对比结果证实了上述观点。尽管面向 ROUGE 的训练方法带有投机嫌疑，却也给出一个重要启示：一个好的摘要评价方法，对自动摘要模型的构建是有积极意义的。

6）先验知识

Amplayo 等[49]利用维基百科识别文本主题，Xie 等[23]利用 WordNet 识别重要的句子，证明先验知识可以帮助模型更好地理解文本。类似的，Kryściński 等[48]在解码器中增加了预训练的语言模型，在提高语言流畅性的同时使生成的摘要具有特定的语言风格，证明在解码时融入先验知识，可以提高生成的摘要质量。尽管数据驱动是人工智能的发展趋势，但“黑盒子”式的学习模型存在难以适应环境变化，结果难以解释等问题[74]。一些研究开始重视在机器学习过程中引入先验知识，建立数据驱动和知识驱动相结合的学习模型。这一思路在自动摘要甚至整个自然语言处理领域都值得借鉴。

综上所述，基于 Seq2Seq 模型的生成式摘要可以在网络深度、注意力机制等方面展开更多研究，着重提高模型对长文本的表示能力以及生成句子的质量，并适当引入先验知识以弥补模型学习能力上的不足。此外，摘要质量评价方法还需进一步完善。

5　总　结

Seq2Seq 模型源于机器翻译，文本摘要和机器翻译虽然都属于序列到序列转换问题，但文本摘要聚焦输入序列的关键信息，而且输入和输出之间没有明显的对齐关系[13]，从这一角度来说文本摘要任务更加复杂。尽管 Seq2Seq 模型在机器翻译领域已进入实用阶段[75]，但对于文本摘要来说显然还有很长的路要走。随着模型的不断衍化，Seq2Seq 模型生成摘要的方式跟人类思维越来越接近，与此同时生成摘要的质量也越来越好。尽管该模型依然存在很多不足，如难以处理几千词以上的长文本、模型时间复杂度高、样本标注开销大等，但其可以引领生成式文本摘要未来的研究方向。

参　考　文　献

[1] Luhn H P. The automatic creation of literature abstracts[J]. IBM Journal of Research and Development, 1958, 2(2): 159-165.

[2] Gambhir M, Gupta V. Recent automatic text summarization techniques: A survey[J]. Artificial Intelligence Review, 2017, 47(1): 1-66.

[3] 刘家益, 邹益民 . 近 70 年文本自动摘要研究综述[J]. 情报科学, 2017, 35(7): 154-161.

[4] Owczarzak K, Dang H T. Overview of the TAC 2011 summarization track: Guided task and aesop task[C]//Proceedings of the Fourth Text Analysis Conference, Gaithersburg, Maryland, USA, 2011.

[5] Genest P E, Lapalme G. Fully abstractive approach to guided summarization[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2012, 2: 354-358.

[6] Khan A, Salim N. A review on abstractive summarization methods[J]. Journal of Theoretical and Applied Information Technology, 2014, 59(1): 64-72.

[7] Cho K, van Merriënboer B, Gulcehre C, et al. Learning phrase

representations using RNN encoder-decoder for statistical machine translation[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2014: 1724-1734.

[8] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2014: 3104-3112.

[9] Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2015: 379-389.

[10] Jing H Y. Using hidden Markov modeling to decompose human-written summaries[J]. Computational Linguistics, 2002, 28(4): 527-543.

[11] Shi T, Keneshloo Y, Ramakrishnan N, et al. Neural abstractive text summarization with sequence-to-sequence models[OL]. (2018-12-7) [2018-12-15]. https://arxiv.org/abs/1812.02303.

[12] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[OL]. (2015-4-24) [2018-10-15]. https://arxiv.org/abs/1409.0473v6.

[13] Zhou Q Y, Yang N, Wei F R, et al. Selective encoding for abstractive sentence summarization[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2017: 1095-1104.

[14] Zeng W Y, Luo W J, Fidler S, et al. Efficient summarization with read-again and copy mechanism[OL]. (2016-11-10) [2018-9-15]. https://arxiv.org/abs/1611.03382.

[15] Lin J Y, Sun X, Ma S M, et al. Global encoding for abstractive summarization[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2018: 163-169.

[16] Vaswani A , Shazeer N , Parmar N , et al. Attention is all you need [C]// Proceedings of the Conference on Advances in Neural Information Processing Systems, Long Beach, USA, 2017: 6000-6010.

[17] Nallapati R, Zhou B W, dos Santos C, et al. Abstractive text summarization using sequence-to-sequence RNNs and beyond[C]// Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning. Stroudsburg: Association for Computational Linguistics, 2016: 280-290.

[18] Celikyilmaz A, Bosselut A, He X D, et al. Deep communicating agents for abstractive summarization[C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association for Computational Linguistics, 2018, 1: 1662-1675.

[19] Cohan A, Dernoncourt F, Kim D S, et al. A discourse-aware attention model for abstractive summarization of long documents[C]// Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association for Computational Linguistics, 2018, 2: 615-621.

[20] Ling J, Rush A. Coarse-to-fine attention models for document summarization[C]//Proceedings of the Workshop on New Frontiers in Summarization. Stroudsburg: Association for Computational Linguistics, 2017: 33-42.

[21] 王帅, 赵翔, 李博, 等. TP-AS: 一种面向长文本的两阶段自动摘要方法[J]. 中文信息学报, 2018, 32(6): 71-79.

[22] Mihalcea R, Tarau P. Textrank: Bringing order into text[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004: 404-411.

[23] Xie N T, Li S J, Ren H L, et al. Abstractive summarization improved by WordNet-based extractive sentences[C]//Proceedings of the 7th CCF International Conference on Natural Language Processing and Chinese Computing. Cham: Springer, 2018, 11108: 404-415.

[24] Chen Y C, Bansal M. Fast abstractive summarization with reinforce-selected sentence rewriting[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2018: 675-686.

[25] Chen Y B, Ma Y, Mao X D, et al. Abstractive summarization with the aid of extractive summarization[C]//Proceedings of Asia-Pacific Web and Web-Age Information Management Joint International Conference on Web and Big Data. Cham: Springer, 2018, 10987: 3-15.

[26] Hsu W T, Lin C K, Lee M Y, et al. A unified model for extractive and abstractive summarization using inconsistency loss[OL]. (2018-7-5) [2018-9-15]. https://arxiv.org/abs/1805.06266.

[27] Tan J W, Wan X J, Xiao J G, et al. Abstractive document summarization with a graph-based attentional neural model[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2017: 1171-1181.

[28] Li C L, Xu W R, Li S, et al. Guiding generation for abstractive text summarization based on key information guide network[C]// Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association for Computational Linguistics, 2018: 55-60.

[29] Jiang X P, Hu P, Hou L W, et al. Improving pointer-generator network with keywords information for Chinese abstractive summarization[C]//Proceedings of the 7th CCF International Conference on Natural Language Processing and Chinese Computing. Cham:

Springer, 2018: 464-474.

[30] 侯丽微, 胡珀, 曹雯琳. 主题关键词信息融合的中文生成式自动摘要研究[J]. 自动化学报, 2019, 45(3): 530-539.

[31] Cao Z Q, Wei F R, Li W J, et al. Faithful to the original: Fact aware neural abstractive summarization[L]. (2017-11-13) [2018-10-15]. https://arxiv.org/abs/1711.04434v1.

[32] Chopra S, Auli M, Rush A M. Abstractive sentence summarization with attentive recurrent neural networks[C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association for Computational Linguistics, 2016: 93-98.

[33] Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning[C]//Proceedings of the 34th International Conference on Machine Learning. PMLR, 2017, 70: 1243-1252.

[34] 周健, 田萱, 崔晓晖. 基于改进 Sequence-to-Sequence 模型的文本摘要生成方法[J]. 计算机工程与应用, 2019, 55(1): 128-134.

[35] Song S L, Huang H T, Ruan T X. Abstractive text summarization using LSTM-CNN based deep learning[J]. Multimedia Tools and Applications, 2019, 78(1): 857-875.

[36] Gulcehre C, Ahn S, Nallapati R, et al. Pointing the unknown words[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2016: 140-149.

[37] Vinyals O, Fortunato M, Jaitly N. Pointer networks[C]//Proceedings of the Conference on Advances in Neural Information Processing Systems, Montreal, Canada, 2015: 2692-2700.

[38] See A, Liu P J, Manning C D. Get to the point: Summarization with pointer-generator networks[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2017: 1073-1083.

[39] Zhou Q, Yang N, Wei F, et al. Sequential copying networks[C]//Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2018: 4987-4995.

[40] Gu J T, Lu Z D, Li H, et al. Incorporating copying mechanism in sequence-to-sequence learning[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2016: 1631-1640.

[41] Tu Z P, Lu Z D, Liu Y, et al. Modeling coverage for neural machine translation[OL]. (2016-8-6) [2018-10-15]. https://arxiv.org/abs/1601.04811.

[42] Sankaran B, Mi H T, Al-Onaizan Y, et al. Temporal attention model for neural machine translation[OL]. (2016-8-9) [2018-9-15]. https://arxiv.org/abs/1608.02927.

[43] Paulus R, Xiong C M, Socher R. A deep reinforced model for abstractive summarization[OL]. (2017-11-13) [2018-9-15]. https://arxiv.org/abs/1705.04304.

[44] Li P J, Lam W, Bing L D, et al. Deep recurrent generative decoder for abstractive text summarization[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2017: 2091-2100.

[45] Kingma D P, Welling M. Auto encoding variational bayes[OL]. (2014-5-1) [2018-9-15]. https://arxiv.org/abs/1312.6114.

[46] Bowman S R, Vilnis L, Vinyals O, et al. Generating sentences from a continuous space[C]//Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning. Stroudsburg: Association for Computational Linguistics, 2016: 10-21.

[47] Gregor K, Danihelka I, Graves A, et al. Draw: A recurrent neural network for image generation[C]//Proceedings of the 32nd International Conference on Machine Learning. PMLR, 2015, 37: 1462-1471.

[48] Kryściński W, Paulus R, Xiong C M, et al. Improving abstraction in text summarization[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2018: 1808-1817.

[49] Amplayo R K, Lim S, Hwang S. Entity commonsense representation for neural abstractive summarization[OL]. (2018-6-14) [2018-10-15]. https://arxiv.org/abs/1806.05504.

[50] Cibils A, Musat C, Hossmann A, et al. Diverse beam search for increased novelty in abstractive summarization[OL]. (2018-2-5) [2018-9-15]. https://arxiv.org/abs/1802.01457.

[51] Ayana, Shen S Q, Zhao Y, et al. Neural headline generation with sentence-wise optimization[OL]. (2016-10-9) [2018-9-15]. https://arxiv.org/abs/1604.01904.

[52] Ranzato M, Chopra S, Auli M, et al. Sequence level training with recurrent neural networks[OL]. (2015-11-20) [2018-10-15]. https://arxiv.org/abs/1511.06732.

[53] Rennie S J, Marcheret E, Mroueh Y, et al. Self-critical sequence training for image captioning[OL]. (2016-12-2) [2018-9-15]. https://arxiv.org/abs/1612.00563.

[54] Li P J, Bing L D, Lam W. Actor-critic based training framework for abstractive summarization[OL]. (2018-8-15) [2018-9-15]. https://arxiv.org/abs/1803.11070.

[55] Konda V R, Tsitsiklis J N. Actor-critic algorithms[C]//Proceedings of the Conference and Workshop on Neural Information Processing Systems, Denver, USA, 2000: 1008-1014.

[56] Shi Y S, Meng J, Wang J, et al. A normalized encoder-decoder model for abstractive summarization using focal loss[C]//Proceedings of the 7th CCF International Conference on Natural Language Processing and Chinese Computing. Cham: Springer, 2018, 11109s: 383-392.

[57] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of IEEE International Conference on

Computer Vision. IEEE, 2017: 2999-3007.

[58] Hua X Y, Wang L. A pilot study of domain adaptation effect for neural abstractive summarization[OL]. (2017-7-21) [2018-10-15]. https://arxiv.org/abs/1707.07062.

[59] Fan A, Grangier D, Auli M. Controllable abstractive summarization[OL]. (2018-5-18) [2018-9-15]. https://arxiv. org/abs/1711.05217.

[60] Zhang J M, Tan J W, Wan X J. Towards a neural network approach to abstractive multi-document summarization[OL]. (2018-4-24) [2018-9-15]. https://arxiv.org/abs/1804.09010.

[61] Chu E, Liu P J. Unsupervised neural multi-document abstractive summarization[OL]. (2018-10-12) [2018-10-15]. https://arxiv.org/abs/1810.05739.

[62] Nema P, Khapra M M, Laha A, et al. Diversity driven attention model for query-based abstractive summarization[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2017: 1063-1072.

[63] Hasselqvist J, Helmertz N, Kageback, M. Query-based abstractive summarization using neural networks[OL]. (2017-12-17) [2018-9-15]. https://arxiv.org/abs/1712.06100.

[64] Baumel T, Eyal M, Elhadad M. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models[OL]. (2018-1-23) [2018-9-15]. https://arxiv.org/abs/1801.07704.

[65] Pasunuru R, Guo H, Bansal M. Towards improving abstractive summarization via entailment generation[C]//Proceedings of the Workshop on New Frontiers in Summarization. Stroudsburg: Association for Computational Linguistics, 2017: 27-32.

[66] Pasunuru R, Bansal M. Multi-reward reinforced summarization with saliency and entailment[C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg: Association for Computational Linguistics, 2018, 2: 646-653.

[67] Guo H, Pasunuru R, Bansal M. Soft layer-specific multi-task summarization with entailment and question generation[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2018: 687-697.

[68] Wang L, Yao J L, Tao Y Z, et al. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization[OL]. (2018-6-2) [2018-10-15]. https://arxiv.org/abs/1805.03616.

[69] Song K Q, Zhao L, Liu F. Structure-infused copy mechanisms for abstractive summarization[C]//Proceedings of the 27th International Conference on Computational Linguistics. Stroudsburg: Association for Computational Linguistic, 2018: 1717-1729.

[70] Hu B T, Chen Q C, Zhu F Z. LCSTS: A large scale Chinese short text summarization dataset[OL]. (2016-9-19) [2018-9-15]. https://arxiv.org/abs/1506.05865.

[71] 王毅, 谢娟, 成颖 . 结合 LSTM 和 CNN 混合架构的深度神经网络语言模型[J]. 情报学报, 2018, 37(2): 194-205.

[72] Hu D C. An introductory survey on attention mechanisms in nlp problems[OL]. (2018-11-12) [2018-12-15]. https://arxiv. org/abs/1811.05544.

[73] Lin C Y, Hovy E. Automatic evaluation of summaries using n-gram co-occurrence statistics[C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Stroudsburg: Association for Computational Linguistics, 2003, 1: 71-78.

[74] 吴飞, 阳春华, 兰旭光, 等 . 人工智能的回顾与展望[J]. 中国科学基金, 2018, 32(3): 243-250.

[75] 李亚超, 熊德意, 张民 . 神经机器翻译综述[J]. 计算机学报, 2018, 41(12): 2734-2755.

（责任编辑　王克平）