北京大学学报(自然科学版) 第 57 卷 第 1 期 2021 年 1 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 57, No. 1 (Jan. 2021)
doi: 10.13209/j.0479-8023.2020.084

基于语义对齐的生成式文本摘要研究

吴世鑫 黄德根† 李玖一

大连理工大学计算机学院, 大连 116023; † 通信作者, E-mail: huangdg@dlut.edu.cn

摘要 针对当前生成式文本摘要模型在解码时对摘要整体语义信息利用不充分的问题, 提出一种基于语义对齐的神经网络文本摘要方法。该方法以带注意力、Pointer 机制和 Coverage 机制的 Sequence-to-Sequence 模型为基础, 在编码器与解码器之间加入语义对齐网络, 实现文本到摘要的语义信息对齐; 将获得的摘要整体语义信息与解码器的词汇预测上下文向量进行拼接, 使解码器在预测当前词汇时不仅利用已预测词汇序列的部分语义, 而且考虑拟预测摘要的整体语义。在中文新闻语料 LCSTS 上的实验表明, 该模型能够有效地提高文本摘要的质量, 在字粒度上的实验显示, 加入语义对齐机制可以使 Rouge_L 值提高 5.4 个百分点。

关键词 生成式文本摘要; Sequence-to-Sequence 模型; 语义对齐网络

Abstractive Text Summarization Based on Semantic Alignment Network

WU Shixin, HUANG Degen†, LI Jiuyi

Dalian University of Technology, Dalian 116023; † Corresponding author, E-mail: huangdg@dlut.edu.cn

Abstract Aiming at the problem of insufficient utilization of the overall semantic information of abstracts in decoding by the currently abstractive summarization model, this paper proposes a neural network automatic abstract model based on semantic alignment. This model is based on the Sequence-to-Sequence model with attention, Pointer mechanism and Coverage mechanism. A semantic alignment network is added between the encoder and the decoder to achieve the semantic information alignment of the text to the abstract. The achieved semantic information is concatenated with the context vector in decoding, so that when the decoder predicts the vocabulary, it not only uses the partial semantics before decoding, but also considers the overall semantics of the digest sequence. Experiments on the Chinese news corpus LCSTS show that the proposed model can effectively improve the quality of abstractive summarization.

Key words abstractive summarization; Sequence-to-Sequence model; semantic alignment network

文本摘要任务指计算机自动生成准确地、全面地反映某一文本中心内容的简洁且连贯短文的过程[1]。文本摘要自动生成技术应用广泛, 尤其在提高用户获取信息效率和实现文本压缩存储方面的作用越来越突出。文本摘要任务的分类方式有很多, 按照摘要生成方法, 可以分为抽取式摘要和生成式摘要。抽取式摘要指从文本中抽取现有的若干句子, 组合成为文本的摘要; 生成式摘要是在综合分析原文信息后, 通过算法自动生成新句子作为文本的摘要。

由于生成式方法更贴近人工生成摘要的过程, 且在语法准确度和语义连贯性方面比抽取式方法更有优势, 因此受到越来越多的重视。目前生成式方法通常以序列到序列(Sequence-to-Sequence)深度神经网络模型[2]为基础, 该模型的作用是在编码器–解码器框架下, 将源序列转化为目标序列。对文本摘要自动生成任务而言, 就是将文本词汇序列转化为摘要词汇序列。

生成式文本摘要技术近年来得到长足发展, 有很多研究者尝试对编码端进行改进, 不断挖掘并细

国家自然科学基金(U1936109, 61672127)资助
收稿日期: 2020–05–15; 修回日期: 2020–08–12

化对文本有效信息的利用程度。Lin 等[3]提出一种基于卷积神经网络(Convolutional Neural Network, CNN)和自注意力机制的全局编码门, 增强了对文本词汇特征和内部联系的挖掘。Nallapati 等[4]以带注意力机制的 Sequence-to-Sequence 模型为基础, 利用语言特征构造词向量, 并将其与普通词向量相结合, 丰富了编码器词嵌入时包含的信息。Chopra 等[5]利用卷积工具, 在原词汇编码上增加位置和上下文信息, 提出条件循环神经网络模型。Wang 等[6]在编码器文本表示部分构建检索、重排序以及 BiSET 三个模型, 通过检索和重排序构建文本表示模板, 再利用 BiSET, 结合模板与文本构建编码器端的文本表示, 提高了文本摘要的质量。

有些研究者将解码端作为切入点, 通过增强解码器词汇预测的准确性来提高摘要的生成质量。Gehrmann 等[7]针对解码器在文本内容选择上的局限性, 提出一种短语粒度的内容选择器作为自底向上的注意力机制, 使解码器更多地关注文本中可作为摘要的短语。Song 等[8]在词向量基础上扩展词汇结构特征信息, 并将句子语法结构融入解码器的注意力机制中, 使结构化的关键词及其在文本中的语法关系保留到摘要词汇预测序列中, 帮助再现事实细节。See 等[9]为解决解码器无法生成词表外词汇(out of vocabu-lary, OOV)以及预测词汇重复的问题, 分别提出指针机制和覆盖机制, 用来提高生成摘要的质量。Tan 等[10]提出一种基于图的注意力机制, 提高解码器对文本中显性信息的挖掘程度, 并在解码阶段利用一种分层集束搜索算法来生成多句摘要。Cao 等[11]利用相似句子有相似摘要的假设, 在解码器中引入由检索、重排序和重写 3 个部分组成的软模板, 首先通过检索得到相似句, 然后利用重排序进行排序, 最后通过重写进行摘要词汇预测。以上对解码器进行改进的方法均提高了摘要生成的质量, 但在解码时, 解码器往往仅对已预测的一个或多个词汇语义进行片段性利用, 未有效地利用预测摘要的整体语义信息。

针对上述问题, 本文提出一种基于语义对齐的神经网络文本摘要模型, 以带注意力、Coverage 机制和 Pointer 机制的 Sequence-to-Sequence 模型为基础, 在编码端与解码端之间加入语义对齐网络。通过该语义对齐网络, 挖掘预测摘要的整体语义信息, 并在解码端进行词汇预测时, 将该整体语义信息与上下文向量进行拼接, 从而丰富上下文的向量表示, 提高词汇预测时的准确性。

1 基于指针网络和覆盖机制的生成式文本摘要模型

1.1 指针网络(pointer network)

由于词表大小的限制, 生成式文本摘要在解码过程中无法产生 OOV。指针网络[9]是在注意力机制基础上的改进, 允许解码器通过从词典中生成或复制输入词汇两种方式进行词汇预测。

首先, 计算基于注意力机制的词汇预测概率; 然后, 在给定 t 时刻上下文向量 $\boldsymbol{C}_t$、解码器隐状态 $\boldsymbol{s}_t$ 和 t–1 时刻解码器预测序列 $\boldsymbol{y}_{t-1}$ 的条件下计算利用生成方式进行词预测的概率 p_{gen}; 最后, 将 p_{gen} 作为开关, 并结合词汇预测概率和注意力分布, 计算 Pointer 机制下的词汇预测概率。

1.2 覆盖机制(coverage mechanism)

传统的基于注意力机制的 Sequence-to-Sequence 模型中, 预测词汇出现重复是普遍存在的问题。原因是在解码过程中, 注意力机制的重复关注造成信息的冗余, Coverage 机制[9]可以用来解决注意力重复问题。

Coverage 机制首先在模型中构建一个覆盖向量 $\boldsymbol{c}_i^t$, 表示解码器在 t 时刻之前对文本第 i 个词汇注意力分布的总和; 然后, 利用覆盖向量构造新的注意力计算公式, 确保计算当前时刻注意力时将之前所有的注意力分布情况考虑在内; 最后, 将覆盖损失加入最后的损失函数中。

2 基于语义对齐的生成式文本摘要模型

本文以带注意力、Pointer 和 Coverage 机制的 Sequence-to-Sequence 模型为基础, 通过添加语义对齐网络, 构造基于语义对齐的神经网络文本摘要模型。模型的输入为文本词汇序列 x, 输出为摘要词汇序列 y, 整体结构如图 1 所示。

2.1 编码端

编码端采用两层长短期记忆(long short-term memory, LSTM)网络结构, 第一层为双向 LSTM 网络, 第二层为单向 LSTM 网络。构建过程如下:

$$\vec{\boldsymbol{h}}_i = \text{LSTM}(\vec{\boldsymbol{h}}_{i-1}, \boldsymbol{x}_i), \tag{1}$$

$$\overleftarrow{\boldsymbol{h}}_i = \text{LSTM}(\boldsymbol{x}_i, \overleftarrow{\boldsymbol{h}}_{i+1}), \tag{2}$$

$$\boldsymbol{h}_i^e = \overrightarrow{\boldsymbol{h}_i^e} = \text{LSTM}(\boldsymbol{h}_{i-1}^e, [\vec{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i]), \tag{3}$$

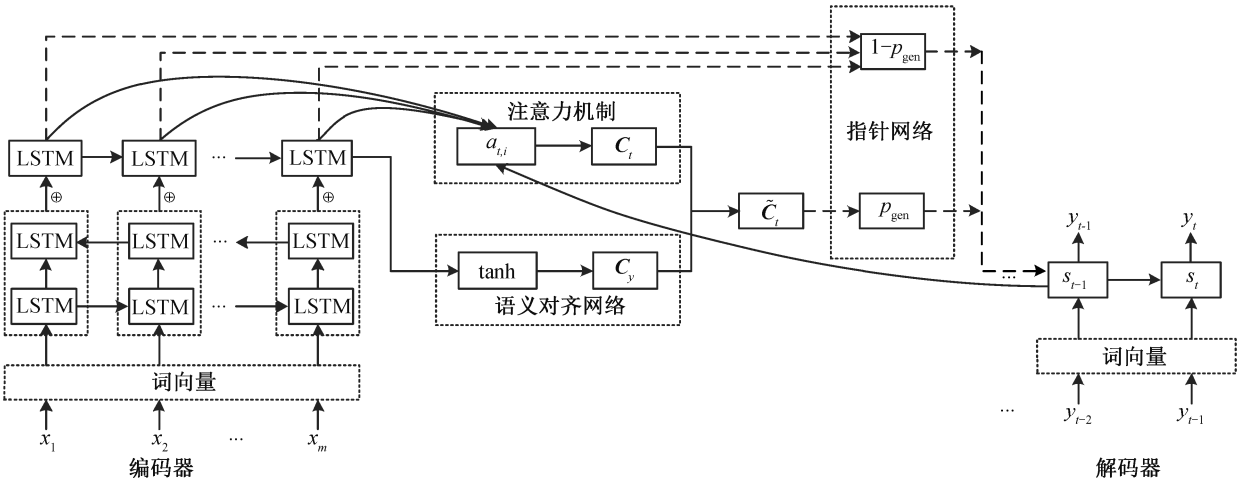

图 1　基于语义对齐网络的生成式文本摘要模型

Fig. 1　Abstractive text summarization model based on semantic alignment Network

其中，$\vec{\boldsymbol{h}}_i$ 和 $\overleftarrow{\boldsymbol{h}}_i$ 分别表示输入序列在第一层双向 LSTM 网络得到的前向和反向隐状态，$\boldsymbol{h}_i^e$ 表示第二层 LSTM 网络的隐状态。

2.2 解码端

解码端采用单层单向 LSTM 网络结构，构建过程如下:

$$\boldsymbol{s}_t = \mathrm{LSTM}(\boldsymbol{s}_{t-1}, \mathbf{y}_{t-1}), \tag{4}$$

其中，S_t 表示 t 时刻的解码隐状态。

1) 注意力机制[9]:

$$e_{t,i} = \boldsymbol{V}^{\mathrm{T}} \tanh(\boldsymbol{W}^e[s_{t-1}; \boldsymbol{h}_i^e] + \boldsymbol{b}^e), \tag{5}$$

$$\alpha_{t,i} = \frac{\exp(\boldsymbol{e}_{t,i})}{\sum_{j=1}^{m} \exp(\boldsymbol{e}_{t,\mathrm{j}})}, \tag{6}$$

其中，m 为输入序列的长度；$\boldsymbol{V}$，$\boldsymbol{W}^e$ 和 $\boldsymbol{b}^e$ 均为可训练的参数。

2) Coverage 机制[9]:

$$c_i^t = \sum_{k=0}^{t-1} \alpha_{k,i}, \tag{7}$$

$$e_i^t = \boldsymbol{v}^{\mathrm{T}} \tanh(\boldsymbol{W}_h \boldsymbol{h}_i + \boldsymbol{W}_s \boldsymbol{s}_t + \boldsymbol{W}_c \boldsymbol{c}_i^t + \boldsymbol{b}_{\mathrm{attn}}), \tag{8}$$

$$a_{t,i} = \frac{\exp(\boldsymbol{e}_i^t)}{\sum_{j=1}^{m} \exp(\boldsymbol{e}_i^t)}, \tag{9}$$

$$\boldsymbol{C}_t = \sum_{i=1}^{m} a_{t,i} \boldsymbol{h}_i^e, \tag{10}$$

其中，$\boldsymbol{C}_t$ 表示解码器在 t 时刻进行词汇预测的上下文向量；$\boldsymbol{v}^{\mathrm{T}}, \boldsymbol{W}_c, \boldsymbol{W}_s, \boldsymbol{W}_c$ 和 $\boldsymbol{b}_{\mathrm{attn}}$ 均为可训练参数。

3) 语义对齐网络(semantic alignment network, SAN): 通过文本与摘要之间的语义对齐，丰富解码器在预测词汇时上下文向量的语义信息，构造过程如下:

$$\boldsymbol{C}_y = \tanh(\boldsymbol{W}_a \boldsymbol{h}_m^e + \boldsymbol{b}_a), \tag{11}$$

$$\widetilde{\boldsymbol{C}_t} = \left[\boldsymbol{C}_t; \boldsymbol{C}_y\right], \tag{12}$$

其中，$\boldsymbol{h}_m^e$ 为 encoder 第二层 LSTM 网络的末尾隐状态，$\boldsymbol{W}_a$ 和 $\boldsymbol{b}_a$ 为可训练参数。

4) Pointer 机制[9]:

$$P_{\mathrm{vocab}}(y) = \mathrm{softmax}(\boldsymbol{W}^y[\widetilde{\boldsymbol{C}_t}; \boldsymbol{s}_t] + \boldsymbol{b}^y), \tag{13}$$

$$p_{\mathrm{gen}} = \sigma(\boldsymbol{W}_h^{\mathrm{T}} \widetilde{\boldsymbol{C}}_t + \boldsymbol{W}_s^{\mathrm{T}} \boldsymbol{s}_t + \boldsymbol{W}_y^{\mathrm{T}} \boldsymbol{y}_{t-1} + \boldsymbol{b}_{\mathrm{ptr}}), \tag{14}$$

$$p(y) = p_{\mathrm{gen}} p_{\mathrm{vocab}}(y) + (1 - p_{\mathrm{gen}}) \sum_{i:w_i=w} a_{t,i}, \tag{15}$$

其中，$\boldsymbol{W}^y, \boldsymbol{b}^y, \boldsymbol{W}_h^{\mathrm{T}}, \boldsymbol{W}_s^{\mathrm{T}}, \boldsymbol{W}_y^{\mathrm{T}}$ 和 $\boldsymbol{b}_{\mathrm{ptr}}$ 均为可训练参数。

2.3 损失函数

$$L = -\frac{1}{n} \sum_{t=1}^{n+1} \log P\left(\boldsymbol{y}_t^* \mid \boldsymbol{y}_1^*, ..., \boldsymbol{y}_{t-1}^*, \boldsymbol{x}; \theta\right) + \sum_{t=1}^{n+1} \mathrm{covloss}_t, \tag{16}$$

其中，L 表示损失函数，$\boldsymbol{y}_t^*$ 表示预测词汇，n 为摘要序列长度，x 表示输入序列，θ 代表整个模型中的可训练参数。

3 实验结果与分析

3.1 语料

本研究使用 LCSTS[12](A Large Scale Chinese Short Text Summarization Dataset)语料，内容来自新

表 1 基于词粒度和字粒度的实验结果对比

Table 1 Comparison of experimental results based on word and charactor granularity

模型	词粒度			字粒度		
	Rouge_1	Rouge_2	Rouge_L	Rouge_1	Rouge_2	Rouge_L
RNN (baseline)	17.7	8.5	15.8	21.5	8.9	18.6
RNN+Attention+Pointer+Coverage (RAPC)	35.2	18.0	**27.5**	33.0	16.1	24.1
RAPC+SAN	**35.3**	**18.0**	26.2	**34.7**	**17.8**	**29.5**

说明: RAPC 为 RNN+Attention+Pointer+Coverage; 粗体数字表示最优结果, 下同。

浪微博。语料包括 3 个部分: 第 1 部分为 2400591 个文本–摘要序列, 第 2 部分为 10666 个带人工打分标签的文本–摘要序列, 第 3 部分为 1106 个人工交叉打分一致的文本–摘要序列。本文选取第 1 部分为训练集, 第 2 部分为验证集, 第 3 部分为测试集。

3.2 实验结果

从基于字和基于词两个粒度展开实验, 模型训练的相关参数设定如下: 输入和输出词(字)维度均为 128, LSTM 网络隐藏层维度为 256, 学习率为 0.1, batchsize 批次大小为 100, beamsize 集束搜索宽度设定为 4, 词表大小为 50000; 字典大小为 10723, epoch 为 30, 结果评价工具采用 Rouge[13]。实验结果如表 1 所示。

从表 1 可以看出, 在词粒度上, 基于注意力机制、Pointer 机制和 Coverage 机制的 RAPC 模型实验结果总体上比 baseline 好, 在 RAPC 上加入语义对齐网络后 Rouge_1 提高 0.1 个百分点, Rouge_2 持平, Rouge_L 稍降低 1.3 个百分点。在字粒度上, RAPC 在 3 个评价指标上表现均比 baseline 好, 在 RAPC 上加入语义对齐网络后, Rouge_1 提高 1.7 个百分点, Rouge_2 提高 1.7 个百分点, Rouge_L 提高 5.4 个百分点。

为检验语义对齐网络对 UNK(未登录词标识)和词汇重复问题的影响程度, 选取 RAPC 和 RAPC+SAN 两个模型, 对测试集 1106 个文本生成摘要中 UNK 以及出现重复词汇句子的数量进行统计, 结果如表 2 所示。

为检验语义对齐网络对摘要生成质量的影响, 对 RAPC 和 RAPC+SAN 两个模型生成的摘要进行人工评价。首先, 寻找 5 名评价人员(3 名研究生学历, 2 名本科学历); 然后, 分别从两个模型的生成结果中随机选取 200 条摘要内容进行人工评价, 评价内容包括生成摘要与参考摘要的信息吻合度、生成摘要的语言简洁性和可读性, 并从 1 到 5 进行打分, 分数越高代表相应的性能越好; 最后, 对打分结果取均值。人工评价结果如表 3 所示。通过两个例句对摘要生成结果进行对比, 结果如表 4 所示。

表 2 加入语义对齐网络前后出现 UNK 和重复词汇情况

Table 2 UNK and repeated words before and after joining the semantic alignment network

模型	UNK/个	出现重复词汇的句子/个
RAPC (词粒度)	555	43
RAPC+SAN (词粒度)	**526**	**40**
RAPC (字粒度)	0	4
RAPC+SAN (字粒度)	**0**	**3**

3.3 实验分析

从表 1 可以看出, 基于词粒度的实验结果提升不够明显, 而基于字粒度的实验结果整体上有较大的改进。

词粒度实验方面, 从人工评价结果(表 3)可以看出, 加入语义对齐网络后, 生成摘要在信息吻合度、简洁性和可读性上均有一定程度的提升, 说明语义对齐网络能够改善生成摘要的质量。但是, 从 UNK 和词汇重复统计结果(表 2)可以发现, 加入语义对齐网络后, UNK 和词汇重复问题未得到明显的改善。词汇重复问题在模型中主要依靠 Coverage 机制进行规避, 而 UNK 问题与词表大小和 Pointer 机制有很大关系。表 2 中的结果说明语义对齐网络并不能很好地解决以上两个问题, 这在一定程度上影响 Rouge 值的提升, 因此加入语义对齐网络前后

表 3 人工评价结果

Table 3 Results of human evaluation

模型	信息吻合度	简洁性	可读性
RAPC (词粒度)	3.21	3.57	3.61
RAPC+SAN (词粒度)	3.25	3.58	3.69
RAPC (字粒度)	3.23	3.74	3.72
RAPC+SAN (字粒度)	**3.57**	**3.89**	**3.92**

表 4　摘要生成结果对比

Table 4　Summary generation result comparison

原文文本	模型	摘要结果
例句1　市场的阴霾开始扩散到投行从业人员的头上, 那些怀揣着投行梦的年轻人, 此时不得不重新检视自己当初的职业选择。裁员规模加大, 津贴减少, 女性需要在家庭和事业中取舍, 竞争压力剧增。投行在年轻人心中一直像个梦般美好, 而今, 它越来越现实。	参考摘要	市场阴霾扩散投行梦难圆
	RAPC (词粒度)	投行的“<unk>”时代到来了吗？？？
	RAPC+SAN (词粒度)	**投行梦**的“<unk>”: 投行梦的<unk>之路?
	RAPC (字粒度)	投行梦的投行梦与投行梦的故事——投行
	RAPC+SAN (字粒度)	投行梦的年轻人不得不重新检视自己裁员
例句2　继去年底被曝出两位企业主负债跑路半年后, 江阴再现老板跑路事件。5月中旬, 江苏融泰石油科技股份有限公司实际控制人许才良出逃美国, 使当地银行面临近8亿坏账风险。所涉贷款主要集中在工行、华夏银行、广发银行和江阴农商行。	参考摘要	江阴再现老板跑路牵涉近 8 亿银行资金
	RAPC (词粒度)	<unk>石油实际控制人许才良出逃美国面临 8 亿坏账风险风险！！！！江阴
	RAPC+SAN (词粒度)	**江阴再现** 8 亿坏账风险银行面临近 8 亿坏账风险？！！！！江阴
	RAPC (字粒度)	江阴再现老板跑路事件江苏融泰公司面临
	RAPC+SAN (字粒度)	江阴再现老板跑路事件**牵涉 8 亿**坏账贷款

结果基本上持平(表 1)。摘要生成结果(表 4)的示例也可以印证上述分析, 加入语义对齐网络后, 例句 1 中原来的“投行”变成“投行梦”, 例句 2 预测出“江阴再现”, 与原文意思更加贴近, 说明词汇生成的准确性得到提高。但是, 依然存在大量词汇重复以及 UNK 的问题, 如例句 1 中加入语义对齐网络后, 由“投行的“<unk>”时代到来了吗???”变成“投行梦的“<unk>”: 投行梦的<unk>之路?”, 词汇重复和 UNK 问题并未得到有效的改善。

字粒度实验方面, 从人工评价结果(表 3)可以看出, 加入语义对齐网络后, 预测摘要在信息吻合度、简洁性和可读性上均有较大的提升, 尤其是生成摘要与参考摘要的信息吻合度提升 0.34。究其原因, 基于字粒度的模型在生成摘要序列时使用的是字典, 与固定大小的词表相比, 能够显著地提升词汇覆盖度, 有效地规避 UNK 问题。从 UNK 和词汇重复统计结果(表 2)可以看出, 基于字粒度的模型未出现 UNK, 在这种情况下, 解码器预测词汇时使用的上下文语义信息在句子生成时的作用会更加突出。表 1 和 3 的结果也表明, 加入语义对齐网络有效地改善了生成摘要的整体质量。在摘要生成结果(表 5)的示例中, 加入语义对齐网络后, 例句 1 中“投行梦的年轻人不得不重新检视自己裁员”比原来的“投行梦的投行梦与投行梦的故事——投行”在可读性和简洁性上都有很大的提升, 例句 2 预测出“牵涉 8 亿”, 与参考摘要的信息吻合度更高。可见加入语义对齐网络后确实丰富了解码过程的整体语义, 对提升摘要效果有较大的帮助。

4 结论

本文针对生成式文本摘要模型中解码器对摘要整体语义利用不充分的问题, 提出一种基于语义对齐的神经网络文本摘要模型。该模型通过构造语义对齐网络, 提高了解码器在解码时上下文向量的语义信息表示能力, 使解码器在预测当前词汇时, 不仅利用已预测词汇序列的部分语义, 而且考虑拟预测摘要的整体语义。实验结果表明, 基于语义对齐的神经网络文本摘要模型能够提升生成摘要的准确性和连贯性。

参考文献

[1] Gomaa W H, Fahmy A A. A survey of text similarity approaches. International Journal of Computer Applications, 2014, 68(13): 13–18

[2] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks // Advances in Neural Information Processing Systems. Montreal, 2014: 3104–3112

[3] Lin J, Xu S, Ma S, et al. Global encoding for abstractive summarization // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Melbourne, 2018: 163–169

[4] Nallapati R, Zhai F, Zhou B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents // Thirty-First AAAI Conference on Artificial Intelligence. San

Francisco, 2017: 3075–3081

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

[6] Wang K, Quan X, Wang R. BiSET: bi-directional selective encoding with template for abstractive summarization // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, 2019: 2153–2162

[7] Gehrmann S, Deng Y, Rush A. Bottom-up abstractive summarization // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, 2018: 4098–4109

[8] Song K, Zhao L, Liu F. Structure-infused copy mechanisms for abstractive summarization // Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, 2018: 1717–1729

[9] See A, Liu P J, Manning C D. Get to the point: summarization with pointer-generator networks. Association for Computational Linguistics, 2017, 17: 1073–1083

[10] Tan J, Wan X, Xiao J. Abstractive document summarization with a graph-based attentional neural model // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, 2017: 1171–1181

[11] Cao Z, Li W, Li S, et al. Retrieve, rerank and rewrite: soft template based neural summarization // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, 2018: 152–161

[12] Hu B, Chen Q, Zhu F. LCSTS: a large scale chinese short text summarization dataset // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015: 1967–1972

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