doi:10.11835/j.issn.1000-582X.2020.244

结合 Transformer 模型与深度神经网络的数据到文本生成方法

许晓泓[1],何 霆[1],王华珍[1],陈 坚[2]
(1. 华侨大学 计算机科学与技术学院,福建 厦门 361021;2. 智业软件股份有限公司,福建 厦门 361000)

摘要:数据到文本的生成是指从结构化数据生成连贯文本的一种自然语言处理方法。近年来,由于端到端训练的深度神经网络的应用,数据到文本生成的方法显示出了巨大潜力。该方法能够处理大量数据自动生成连贯性文本,常用于新闻写作、报告生成等场景。然而,已有研究中对于数据中具体数值、时间等数据信息的推理存在较大缺陷,无法充分利用数据间的结构信息给出合理的生成指引,并且生成过程容易出现语义与句法分离训练的问题。因此,文中提出一种结合 Transformer 模型与深度神经网络的数据到文本生成方法,并提出一个用于内容规划的 Transformer Text Planning(TTP)算法,有效地解决上述问题。在 Rotowire 公开数据集上进行方法验证,实验结果表明,文中方法性能优于已有数据到文本生成模型,可直接应用于结构化数据到连贯性文本的生成任务中,具有一定的实际应用价值。

关键词:文本生成;Transformer 模型;内容预选;内容规划;深度神经网络
中图分类号:TP311 **文献标志码**:A **文章编号**:1000-582X(2020)07-091-10

Research on data-to-text generation based on transformer model and deep neural network

XU Xiaohong[1], *HE Ting*[1], *WANG Huazhen*[1], *CHEN Jian*[2]
(1. College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian 361021, P. R. China;
2. ZoeSoft Corp. Ltd., Xiamen, Fujian 361000, P. R. China)

Abstract: Data-to-text generation is a natural language processing method that generates coherent text from structured data. In recent years, data-to-text generation have shown great promise of profit due to the popular neural network architectures which are trained end-to-end. This method can automatically process large amounts of data and generate coherent text and is often used in news writing, report generation, etc. However, there are some defects in the reasoning of information such as the data of specific value and time in the existing researches, which make it unable to make full use of the structural information of data to provide reasonable guidance for the generation. Beyond that the generation process is prone to separate semantic from syntactic when training. In this paper, a data-to-text generation method based on transformer model and deep neural network was proposed, and the algorithm of transformer text planning

收稿日期:2019-12-09
基金项目:国家重点研发计划资助项目(2018YFB1402500)。
Supported by the National Key Research and Development Program of China (2018YFB1402500).
作者简介:许晓泓(1996—),女,硕士研究生,主要从事自然语言处理、文本生成技术研究。
通讯作者:何霆,男,教授,博士生导师,主要从事智慧计算、智慧服务、软件服务工程等研究,(E-mail)heting@hqu.edu.cn。

(TTP) was also introduced so as to effectively control the context information of the generated text and remove the deficiencies of the previous model that resulted in semantics and syntax separation. Experiment results on the Rotowire public dataset show that the method proposed outperforms the existing model and it can be directly applied to the generation task of scattered data to coherent text.

Keywords: text generation; Transformer model; content preselecting; content planning; deep neural network

在信息快速迭代发展的今天,人们很难做到不知疲倦地学习,如何让计算机学习海量知识并像人一样表达和创作逐渐演变成一个重要的研究课题[1]。数据到文本生成技术旨在将结构化数据自动转化为流畅且贴近事实的描述性文本,具有广泛的应用前景。该技术可用于智能写作系统,自动撰写高质量的自然语言文本;也可用于智能问答与机器翻译等系统,实现更加智能化的人机交互;还可用于产品描述的生成,解决日益更新的产品信息所需的大量产品描述生成问题,对数据到文本生成技术的探索具有重要的研究价值。

数据到文本生成任务主要包含两类方法:基于规则的方法以及数据驱动的方法[1]。基于规则的方法需要大量人工特征工程或规则干预,耗时耗力。随着机器学习方法的兴起,数据驱动的方法开始盛行,该方法无须过多的人工干预,通过机器的自主训练,能够生成丰富且流畅性较强的文字描述。基于数据驱动的方法已有学者进行了相关研究[2]。Duboue 等[3]将输入的内容选择作为分类任务判定一个数据库条目是否应该出现在输出中;Barzilay 等[4]将内容选择看作协作分类问题考虑数据库条目之间的依赖性;Liang 等[5]提出隐半马尔可夫模型(HSMM, hidden semi-markov model)匹配数据记录与文本描述生成,实现分割文本到话语,并关联话语到每个对应记录的任务;Angeli 等[6]在 Liang 等[5]的基础上加入对数线性模型,将生成过程细化为一系列本地决策,实现内容选择和表层生成统一;Konstas 等[7]使用概率上下文无关语法,全局地描述输入数据的固有结构,使用超图结构来获得最后的推导;Kondadadi 等[8]利用支持向量机构建基于模板的统计框架,将内容选择与表层生成任务联合成一个统计学习过程;Sowdaboina 等[9]与 Gkatzia 等[10],分别提出使用机器学习解决对时序数据总结的内容选择方法、使用多标签分类被选择的内容,并利用强化学习方法总结时序数据的方法;Mahapatra 等[11]、Hongyuan 等[12]以及 Lebret 等[13]分别采用多属性创建多分区,概率性地选择内容实现多分区图的自然语言生成,利用深度学习实现与领域无关的编码-解码框架的 LSTM 网络模型完成数据到文本生成任务,建立在文本生成的条件神经语言模型且利用事实表格生成初始语句;Wiseman 等[2]采集大量 NBA 篮球比赛统计数据以及对应的文字战报,发布了新的数据集 Rotowire;Qin 等[14]利用 HSMM 学习文本中各部分词汇与输入数据项的对应关系模型,对齐结果也可为自然语言生成提供丰富的规则和模板。相比基于规则的方法,数据驱动的方法无须专家参与,并且更容易优化与扩展。已有的基于数据驱动方法对于数据中具体数值、时间等数据信息的推理存在较大缺陷,无法充分利用数据间的结构信息给出合理的生成指引,并且生成过程容易出现语义与句法分离训练的问题。

为解决上述问题,文中提出结合 Transformer 模型与深度神经网络的数据到文本生成方法,并在 Rotowire 数据集(如表 1 与表 2 所示)上展开了方法的验证。该方法包含 3 个阶段:第一阶段是语句内容预选,通过多层感知器(MLP, multilayer perceptron)与注意力机制(attention mechanism)从输入的结构化数据中选择重要信息构建特征表示空间,有效地对数值、时间等类型数据进行推理,增强数据间的结构信息;第二阶段是文本内容规划,采用基于 Transformer 模型的 Transformer Text Planning 算法将上阶段的特征表示进行上下文信息的推导,给出生成指引序列,有效控制输出文本的连贯性,将语义与句法统一分析;第三阶段是文本生成,利用序列到序列结构中编码器-解码器方式,在双向长短期记忆神经网络(BiLSTM, bi-directional long short-term memory)的基础上,采用联合复制与条件复制两种复制机制构建文本生成器生成最终文本,提高生成信息准确率与文本逻辑性。文中的主要贡献包括 3 个方面:一是提出一种新颖的数据到文本生成模型,该模型能够构建丰富的特征表示空间并生成连贯性较强的文本;二是提出 Transformer Text Planning(TTP)算法进行文本内容规划,给出合理的生成指引,并改善先前工作中语义与句法分离的不

足；三是在 BiLSTM 网络的基础上结合复制机制构造文本生成器，提高生成文本连贯性。

表 1　数据记录的样本（NBA 赛事记录）

Table 1　Example of data-records (NBA games)

球队					球员				
队名	得分	丢分	总分	助攻次数	姓名	得分	助攻次数	篮板数	城市
Pacers	4	6	99	17	Jeff Teague	20	4	3	Indiana
					Miles Turner	17	1	8	Indiana
Celtics	4	6	99	22	Isaiah Thomas	23	5	0	Boston
					Kelly Olynyk	16	4	6	Boston
					Amir Johnson	14	3	9	Boston

表 2　人工撰写文本的样本（NBA 赛事报道）

Table 2　Example of human-written text (NBA games reports)

赛事报道
The **boston** Celtics defeated the host **Indiana Pacers105-99** at Bankers Life Field-house on Saturday. In a battle between two injury-riddled teams, the Celtics were able to prevail with a much needed road victory. The key was shooting and defense, as the **Celtics** outshot the **Pacers** from the field, from three-point range and from the free-throw line. Boston also held Indiana to **42 percent** from the field and **22 percent** from long distance. The Celtics also won the rebounding and assisting differentials, while tying the Pacers in turnovers. There were 10 ties and 10 lead changes, as this game went down to the final seconds. Boston (**5-4**) has had to deal with a gluttony of injuries, but they had the fortunate task of playing a team just as injured here. **Isaiah** Thomas led the team in scoring, totaling **23 points and five assists on 4 of 13** shooting. He got most of those points by going 14 of 15 from the free-throw line. **Kelly Olynyk** got a rare start and finished second on the team with his **16 points, six rebounds and four assists.**

注：NBA 表示美国职业篮球联赛（National Basketball Association）。表 1 仅列出 Rotowire 数据集的部分数据信息，表 2 为表 1 数据记录所对应的专业文本。

1　结合 Transformer 模型与深度神经网络的数据到文本生成方法

1.1　问题描述与定义

以 Rotowire 数据集为例，对整个数据到文本生成问题进行公式化建模。每条样本输入数据作为一个无序记录表 R，R 中每条记录包含多维特征，经过降维分析，将其降为 4 个主要特征，如图 1 所示，这 4 个特征分别为 VALUE（即 $r_{j,1}$）、ENTITY（即 $r_{j,2}$）、TYPE（即 $r_{j,3}$）、H/V（即 $r_{j,4}$），其中，ENTITY 表示球队/球员、H/V 表示该队是主场队还是客场队、TYPE 表示表 1 中每个球队/球员对应的记录属性、VALUE 表示 TYPE 对应的属性值（可为字符串、数字等）。文中对该问题做出如下 2 个定义。

定义 1. 每条输入的样本数据为无序记录表 $R=\{r_j\}_{j=1}^{|r|}$，每个无序记录表中的单个记录表示为 $r_j=\varphi(\{r_{j,k}\}_{k=1}^{4})$，其中，$\varphi(x)$为记录编码函数。每条输入样本对应的输出文本为 $y=y_1y_2\cdots y_{|y|}$（y_i 表示生成文本中的第 i 个单词，$|y|$为文本长度）。

定义 2. 整个数据到文本生成方法的目标函数为

$$\theta^{*}=\arg\max_{\theta}\sum_{(R,y)}\log p(y\mid R;\theta)。\tag{1}$$

结合 2 个定义，将问题分解为 2 个任务，并作为级联目标如下：

1) $\max p(z_q = r_j \mid z_{<q}, r)$;

2) $\max p(y \mid z, r)$。

目标1)中，r 为每条样本的所有记录表示，$z_{<q}$ 为段落内容规划输出的前 q 个规划，z_q 表示段落内容规划输出的第 q 个规划；目标2)中，z 表示所有段落内容规划的集合，y 表示最终输出文本。

文中所提出的结合 Transformer 模型与深度神经网络的数据到文本生成方法流程如图1所示，主要包括语句内容预选、文本内容规划和文本生成3个阶段。

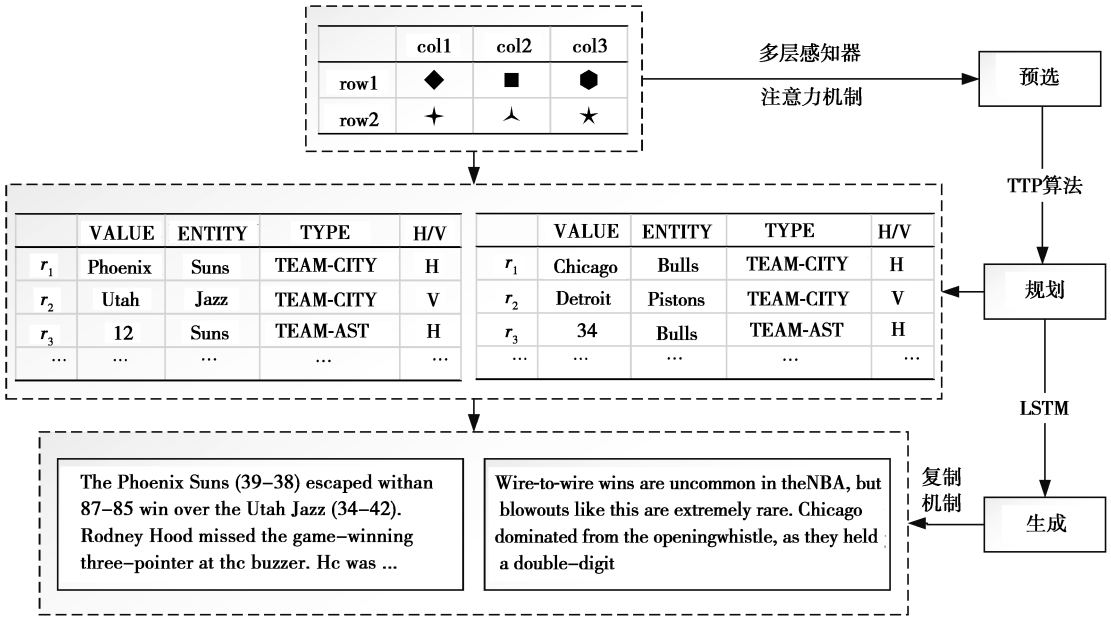

图1 结合 Transformer 模型与深度神经网络的数据到文本生成方法概述

Fig. 1 Overview of data－to－text generation methods with Transformer model and deep neural network

1.2 语句内容预选

在本小节采用多层感知器与注意力机制结合的方法实现语句内容预选。注意力机制最早在视觉图像领域提出，Dzmitry 等[15]使用注意力机制在机器翻译任务上将翻译和对齐同时进行，首次提出将注意力机制应用到自然语言处理领域中，随后基于注意力机制的 RNN 与 CNN 的扩展模型开始应用到各种 NLP 任务中，不仅仅是序列到序列模型，各种分类问题都可以使用注意力机制进行模型构建，解决 NLP 各类文本词汇权重分配问题。

将 VALUE、ENTITY、TYPE、H/V 这4个特征进行向量拼接并输入到多层感知器，从而得到每一条记录的特征向量：

$$r_j = f_r(\boldsymbol{W}_r[r_{j,1}; r_{j,2}; r_{j,3}; r_{j,4}] + \boldsymbol{b}_r)。 \tag{2}$$

式(2)为定义1中 $r_j = \varphi(x)$ 的实现，其中，$\boldsymbol{W}_r \in \mathbb{R}^{n \times 4n}$，$\boldsymbol{b}_r \in \mathbb{R}^n$，$\boldsymbol{W}_r$ 与 $\boldsymbol{b}_r$ 为参数矩阵；[;]为拼接向量；f_r 代表线性整流函数(ReLU, rectified linear unit)，ReLU 是一个神经元激活函数，能够在线性变化($\boldsymbol{W}_r[;]+\boldsymbol{b}_r$)之后进行 $f_r(x) = \max(0, x)$，从而输出非线性结果。

不同记录之间的关联信息可以用来确定该记录对于其余记录的重要性程度，从而给出所有记录对于语句的重要性排序。例如：就 Rotowire 数据集而言，一个球员得分较多，这些得分很可能有很多有价值的相关记录，如投篮得分、三分球、罚球等。为了更好地捕获记录之间的这种依赖关系，提出利用注意力机制进数据间结构关系的信息捕捉。

计算每个输入记录的注意力分数 $\xi_{j,k}$，利用 $\xi_{j,k}$ 为每个 r_j 生成一个注意力向量表示 r_j^{att}：

$$\xi_{j,k} \propto \exp(r_j^{\mathrm{T}} \boldsymbol{W}_a r_k)， \tag{3}$$

$$r_j^{\text{att}} = \boldsymbol{W}_g[r_j; \sum_{k\neq j}\xi_{j,k} r_k], \tag{4}$$

式(3)～式(4)中，$\boldsymbol{W}_a \in \mathbb{R}^{n\times n}$ 与 $\boldsymbol{W}_g \in \mathbb{R}^{n\times 2n}$ 表示参数矩阵，并且 $\sum_{k\neq j}\xi_{j,k}=1$。

将式(4)通过一个 Sigmoid 激活函数，可获得记每条记录新的特征向量 r_j^{cp}：

$$r_j^{\text{cp}} = f_{si}(r_j^{\text{att}}) \odot r_j, \tag{5}$$

式中：$\odot$ 表示对应特征向量中元素的相乘；f_{si} 表示 Sigmoid 神经元激活函数，$\text{Sigmoid}(r_j^{\text{att}}) \in [0,1]^n$，Sigmoid 激活函数通过 $f_{si}(x)=1/1+\text{e}^{-x}$ 来实现。

语句内容预选阶段给出所有样本数据以 VALUE、ENTITY、TYPE、H/V 这 4 个特征拼接转换后的特征表示向量空间。该特征空间包含了不同类型数据(例如：比分、时间等)之间的对应关系以及各记录相对于语句的重要性排序信息。

1.3　文本内容规划

在内容规划阶段提出一个 Transformer Text Planning(TTP)算法，该算法基于 Transformer 模型。Transformer 是一种处理序列到序列问题的新模型，该模型仍然沿用了经典的编码器-解码器(Encoder-Decoder)结构，但不再使用循环神经网络(RNN, recurrent neural network)或者卷积神经网络(CNN, convolutional neural networks)进行序列建模，而是完全使用自注意力机制(self-attention)，该机制可对句子中所有单词之间的关系直接进行建模，而无须考虑各自的位置。Google 研究结果[16]表明，与 RNN 或者 CNN 等结果相比，Transformer 模型可以在减少计算量和提高并行效率的同时获得更好的学习效果。

基于 Transformer 模型的内容规划算法能够将所输入的每条样本的记录特征向量表示集合进行转化，从而输出生成文本所需的内容规划集合。文本内容规划能够令端到端训练的神经网络结构的文本生成器获取到一个明确的生成指引，该指引规定了文本将生成什么并以何种顺序生成。

假定内容规划序列 $z=z_1\cdots z_{|z|}$，每个 z_q 指向一个输入记录 r_j。Transformer 模型由多头注意力构成，每头注意力设为 $z_{q,v}(v\in\{1,2,\cdots,h\})$，则：

$$T(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = f_{so}\left(\frac{\boldsymbol{Q}(\boldsymbol{K})^{\text{T}}}{\sqrt{|d_k|}}\right)\boldsymbol{V}, \tag{6}$$

$$z_{q,v} = T(r_j\boldsymbol{W}_v^{Q}, r_j^{\text{cp}}\boldsymbol{W}_v^{K}, r_{i\neq j}\boldsymbol{W}_v^{V}), \tag{7}$$

$$z_q = \boldsymbol{W}_o[z_{q,1}; z_{q,2}; \cdots; z_{q,h}], \tag{8}$$

其中，f_{so} 代表 Softmax 神经元激活函数，适用于多分类任务，Softmax 通过 $f_{so}(x)_p = \text{e}^{x_p}/\sum_{m=1}^{M}\text{e}^{x_p}$ 来实现；$T(x)$ 表示单头注意力的计算函数。结合式(6)～式(8)与 Transformer 模型，提出一个文本内容规划算法 Transformer Text Planning，如算法 1 所示。

算法 1.　Transformer Text Planning(TTP).

输入：每条样本中所有记录在初始状态下的特征向量表示集合 $r=\{r_1,r_2,\cdots,r_{|r|}\}$；每条样本中所有记录在计算记录间相关关系状态下的特征向量表示集合 $r^{cp}=\{r_1^{cp},r_2^{cp},\cdots,r^{cp}_{|r|}\}$。

输出：文本内容规划序列 $z=\{z_1,\cdots z_q\cdots,z_{|z|}\}$。

过程：

①初始化内容规划集合：$z=\varnothing$；

②设置头注意力的头数为 h；

③初始化每头注意力的值为 $z_{q,v}$ 且 $v\in\{1,2,\cdots,h\}$；

④设置文本输出维度为 d_{model}，$\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 3 个向量维度为 $d_k=d_{\text{model}}/h$；

⑤随机初始化 4 个矩阵 $\boldsymbol{W}_v^{Q}$，$\boldsymbol{W}_v^{K}$，$\boldsymbol{W}_v^{V}$ 以及 W_o，矩阵维度分别为 $d_{\text{model}}\times d_k$，$d_{\text{model}}\times d_k$，$d_{\text{model}}\times d_k$ 以及 $d_{\text{model}}\times hd_k$，矩阵随着模型训练迭代更新；

⑥令 $z_q\in z$ 且 z_q 指向一个输入记录 r_j；

⑦for $q=1,2,\cdots,|z|$ do；

⑧计算每个记录的 $r_j\boldsymbol{W}_v^{Q}$，$r_j^{cp}\boldsymbol{W}_v^{K}$，$r_{i\neq j}\boldsymbol{W}_v^{V}$；

⑨根据式(6)与式(7)计算 $z_{q,v}$；

⑩根据式(8)对所有 $z_{q,v}$ 进行向量拼接,并与 $\boldsymbol{W}_o$ 进行矩阵相乘获得 z_q;

⑪end for。

由算法 1 可知,文本内容规划概率计算为:

$$p(z_q = r_j \mid z_{<q}, r) \propto \exp(z_{<q}^{\mathrm{T}} \boldsymbol{W}_c r_j^{\mathrm{cp}}), \tag{9}$$

式中:$z_{<q}$ 为 z_q 前的内容规划输出序列;$\boldsymbol{W}_c$ 表示训练过程的所有参数矩阵。

1.4 文本生成

使用双向长短期记忆神经网络结合上文的内容规划构建文本生成器,如图 2 所示,该生成器为一个编码器–解码器模型。

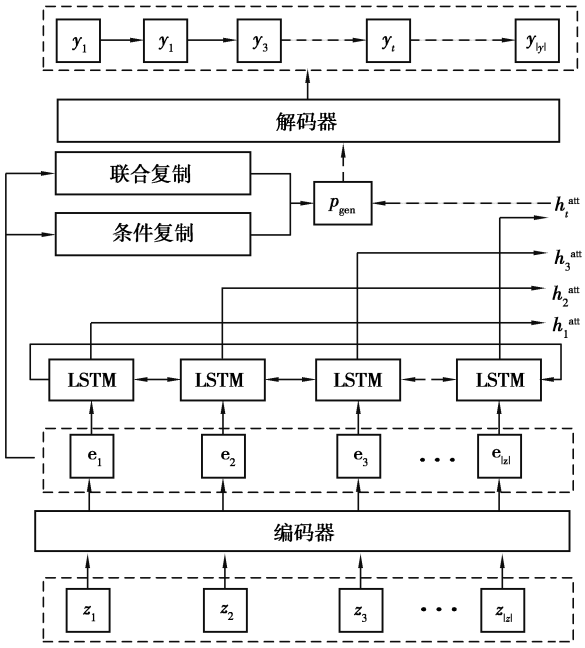

图 2 文本生成器的流程

Fig. 2 Processing of text generator

将内容规划 z 编码为 $\{\mathrm{e}_q\}_{q=1}^{|z|}$,由于内容规划是一个输入标记序列,文中直接将其对应的内容选择的记录向量表示 $\{r_j^{\mathrm{cp}}\}_{j=1}^{|r|}$ 作为 LSTM 神经网络的输入层,该过程作为文本生成器的编码器部分。

解码器部分主要由 LSTM 神经网络构成,使用编码器最后一步的隐藏状态初始化解码器。解码到第 t 步时,将已预测的单词 y_{t-1} 的向量表示输入到该步的 LSTM 神经元中,假设 h_t 为第 t 个 LSTM 神经元的隐藏状态,则该步输出的词的计算公式为

$$\gamma_{t,k} \propto \exp(h_t^{\mathrm{T}} \boldsymbol{W}_b \mathrm{e}_k), \tag{10}$$

$$h_t^{\mathrm{att}} = f_t(\boldsymbol{W}_d[h_t; \sum\nolimits_k \gamma_{t,k} \mathrm{e}_k]), \tag{11}$$

$$p_{\mathrm{gen}}(y_t \mid y_{1:t}, z, r) = f_{so}(\boldsymbol{W}_y h_t^{\mathrm{att}} + \boldsymbol{b}_y)_{y_t}, \tag{12}$$

式中:$\sum_k \gamma_{t,k} = 1$;$\boldsymbol{W}_b \in \mathbb{R}^{n \times n}$,$\boldsymbol{W}_d \in \mathbb{R}^{n \times 2n}$,$\boldsymbol{W}_y \in \mathbb{R}^{n \times d}$,$\boldsymbol{b}_y \in \mathbb{R}^d$ 均为参数矩阵;d 表示输出词典长度;f_t 代表 Tanh 神经元激活函数,Tanh 通过 $f_t(x) = \dfrac{\mathrm{e}^z - \mathrm{e}^{-z}}{\mathrm{e}^z + \mathrm{e}^{-z}}$ 实现。

文中在解码器部分改进了 Wiseman 等[2]提出的复制机制,该复制机制能够将文本内容规划序列 $z = z_1 \cdots z_{|z|}$ 中规划的内容进行复制,主要包括联合复制与条件复制 2 个部分。具体来说,对每一个时间步 t 引入变量 $\varepsilon_t \in \{0,1\}$,其中,$\varepsilon_t = 0$ 表示该记录或单词不必复制,而 $\varepsilon_t = 1$,则表示该记录或单词须复制。因此,第 t 步生成的词 y_t 的概率分布计算为

$$p(y_t \mid y_{1:t-1}, z, r) = \sum_{\varepsilon_t \in \{0,1\}} p(y_t, \varepsilon_t \mid y_{1:t-1}, z, r)。 \tag{13}$$

1.4.1　联合复制机制

联合复制(JC, joint copy)表示从内容规划后的规划序列 $z=z_1\cdots z_{|z|}$ 中复制记录值或复制生成器生成的词的联合复制，复制规则如下：

$$p(y_t,\varepsilon_t \mid y_{1:t-1},z,r)\propto\begin{cases}\sum_{y_t\leftarrow z_q}\exp(h_t^{\mathrm{T}}\boldsymbol{W}_b\mathrm{e}_k),\varepsilon_t=1,\\ \exp(\boldsymbol{W}_y h_t^{\mathrm{att}}+b_y),\varepsilon_t=0。\end{cases}\tag{14}$$

式(14)中，$y_t\leftarrow z_q$ 表示将 z_q 复制给 y_t；其余参数参考式(10)～式(12)所述。

1.4.2　条件复制机制

条件复制(CC, conditional copy)在计算生成和复制的概率时，引入判断某个词来自生成模型或来自复制模型的概率作为条件，具体计算为

$$p(y_t,\varepsilon_t \mid y_{1:t-1},z,r)=\begin{cases}p(\varepsilon_t \mid y_{1:t-1},z,r)\sum_{y_t\leftarrow z_q}\gamma_{t,k},\varepsilon_t=1,\\ p(\varepsilon_t \mid y_{1:t-1},z,r)p_{\mathrm{gen}}(y_t \mid y_{<t},z,r),\varepsilon_t=0。\end{cases}\tag{15}$$

式中，$\gamma_{t,k}$ 与 $p_{\mathrm{gen}}(y_t\mid y_{<t},z,r)$ 分别由式(10)～式(12)计算而得。

联合复制和条件复制的区别在于条件复制能够分解联合复制的情况，在某些情况下更加高效准确。文中方法的文本生成器模块带有条件复制和联合复制 2 种复制机制，能够有效地提高生成文本数据的准确率，使得生成文本的信息更具有可靠性，实验部分也将对比这 2 种复制方式的效果。

1.5　损失与优化

为了实现模型训练目标 θ^*，采用损失函数：

$$L=-\sum_{(r,z,y)\in E}\log p(z \mid r)p(y \mid r,z),\tag{16}$$

进行优化训练，式中，E 代表所有训练样本集(包括记录 r、内容规划 z 以及对应的文本 y)。

由式(1)与式(16)推导得：

$$\max\sum_{(r,z,y)\in E}\log p(z \mid r)+\log p(y \mid r,z)。\tag{17}$$

式(17)表示模型的 2 个级联目标，采用最大似然估计法来降低训练误差，实现模型优化。

2　实　验

2.1　实验数据

在国际公开数据集 Rotowire[2] 上对结合 Transformer 模型与深度神经网络的数据到文本生成方法性能进行了评估。国际公开数据集 Rotowire 是一个包含赛事记录与专业人员撰写的 NBA 赛事报道的英文数据集，如表 1 和表 2 所示。与其他数据到文本生成数据集(例如：E2E 餐厅描述生成数据集、WikiBio 人物传记生成数据集)不同的是，该数据集的文本(NBA 赛事报道)平均长度在 337 词以上，显著长于其余数据集，并且输入数据更丰富(通常每篇报道包含 5～7 个语句)；此外，Rotowire 数据集的文本中直接提及或通过不同措辞蕴含的信息，需从具体的数值、时间等信息推断得出，无法从输入的表格数据中直接获取。例如，输入数据中并没有直接标示比赛的获胜球队，只列出了交战双方比分，而文字内诸如“The Atlanta Hawks beat the Miami Heat 103-95”中的“beat”一词，需要数据到文本生成的模型能够准确捕捉“A team with a higher score wins”这样的对应关系，对当前高度依赖连续向量表示的神经网络方法而言颇具挑战。

Rotowire 包含 2014 年 1 月 1 日至 2017 年 3 月 29 日期间的 NBA 赛事记录数据与专业的赛事报道。该数据集共有 4 853 个样本数据，每个样本对应的赛事报道文本结构与篇幅长度(平均 337 词)，样本词汇量约为 11.3 K 个，带标记词汇量为 1.6 M 个，赛事记录类型共 39 种，平均记录数为 628 条。文中遵循 Wiseman 等将数据集划分为 3 398 条样本数据用于训练集，727 条样本数据用于验证集，728 条样本数据用于测试集。

2.2　实验设置

关于实验参数的设置，在语句内容预选部分，注意力层数设置为 64 层，采用 Glove 预训练模型进行预训练并结合 mean 函数进行编码操作，词向量维度设置为 600；在 Transformer 的文本内容规划部分，多头注意力的平行层数 h 设置为 8，文本输出维度 d_{model} 设置为 344，$\boldsymbol{Q}$ 与 $\boldsymbol{K}$ 向量维度 d_k 设置为 43。在文本生成器部

分，使用了2层LSTM网络，dropout设置为0.3，使用Adagrad优化器对模型进行优化训练，epoch设置为25，初始学习率为0.15，从{0.5，0.97}中选择学习率衰减，batch设置为10，所生成的赛事报道文本语句数量最小值与最大值分别设置为35与80。文中模型在Klein等[17]提出的OpenNMT-py模型上进行构建。

2.3 实验指标

采用CS、CO、RG、BLEU以及PPL这5个评价指标对文中方法进行验证。假设 y_{gold} 为专业参考文本，y_{gen} 为模型输出文本，$P\%$ 表示精确率，$R\%$ 表示召回率，则评测指标含义如下：

1) CS[2](Content Selection)：从 y_{gen} 抽取出的记录数据与 y_{gold} 中匹配的概率来评价内容选择层的性能，采用精确率与召回率的方式计算；

2) CO[2](content ordering)：通过计算 y_{gold} 与 y_{gen} 中记录序列之间的字符串编辑距离(Damerau-Levenshtein距离，即下文所述 $D\%$)，评价内容规划层所给出的记录规划顺序的好坏程度；

3) RG[2](relation generation)：通过从 y_{gold} 抽取出的记录数据同时存在于原始输入记录表的比例来评价模型生成数据的真实性，采用精确率和匹配记录数(即#)来计算；

4) BLEU[18](bilingual evaluation understudy)：由IBM提出一种基于精确度的相似性度量，用于分析 y_{gold} 与 y_{gen} 中 n 元组共同出现的程度；

5) PPL(perplexity)：用于评价整体生成模型的困惑度，衡量语言模型好坏的指标。当语言模型面对一个句子困惑的程度越低，期望的语句出现概率越高，模型越好。

2.4 实验结果

提出的结合Transformer模型与深度神经网络的数据到文本生成方法(ATL)，在联合复制(JC)和条件复制(CC)2种复制机制下与Wiseman等[2]提出基线模型WS-2017和Puduppully等[19]提出的NCP模型进行性能对比，实验结果如表3所示。

表3 CS、RG、CO和BLEU指标下的模型评测结果

Table 3 Evaluation results by model on CS, RG, CO and BLEU metrics

模型	CS		RG		CO	BLEU
	$P\%$	$R\%$	#	$P\%$	$D\%$	
验证集						
WS-2017	28.1	35.9	24.0	75.1	15.3	14.6
NCP+JC	32.2	18.6	33.4	87.4	18.0	14.9
NCP+CC	33.5	51.2	33.9	87.5	18.6	16.2
ATL+JC	32.3	19.0	33.2	88.5	19.2	16.1
ATL+CC	33.7	52.3	33.8	89.6	19.8	16.3
测试集						
WS-2017	29.5	36.2	23.7	74.8	15.4	14.2
NCP+JC	32.0	47.3	34.1	87.2	17.2	14.9
NCP+CC	34.2	51.2	34.3	87.5	18.6	16.5
ATL+JC	32.3	47.2	34.2	87.4	19.2	15.9
ATL+CC	34.3	51.1	34.4	87.9	21.5	16.9

从表3可看出，在验证集与测试集的模型评测结果中，ATL模型对比基线模型WS-2017与NCP模型相比在CO指标得分提升最高，CS、RG、BLEU指标与NCP模型几乎达到一致性能甚至在CS的 $P\%$ 与RG的 $P\%$ 能够保持验证集与测试集指标得分均增长，有力地证明了文中提出的文本内容规划TTP算法能够带来性能提升。BLEU指标也有所提升，可以验证文中算法的文本流畅性能力有所提高。除此之外，可以发现，

相比 JC，CC 下模型的性能更好。因此，文中基于条件复制机制进行进一步分析。

在测试集下，利用条件复制机制展开了模型的困惑度（PPL，perplexity）评估。如表 4 所示，文中构建的 ATL 模型在 PPL 指标上均低于 2 个基线模型，文中方法困惑度有效下降，模型更好地收敛，验证了模型性能较优。

表 4　PPL 指标下的模型评测结果

Table 4　Evaluation results by model on PPL metrics

模型	PPL
WS-2017	7.67
NCP	7.59
ATL	7.38

为了验证生成文本长度对文本连贯性的影响，在测试集下进行了 BLEU-n 指标的评测，结果如表 5 所示。

表 5　BLEU(B-1，2，3，4)指标下的模型评测结果

Table 5　Evaluation results by model on BLEU(B-1～B-4) metrics

模型	BLEU			
	B-1	B-2	B-3	B-4
ATL	47.87	31.04	21.60	16.19

BLEU-n 中的 n 表示 n-gram，即 n 个单词长度的词组集合。当 $n=1$ 时，主要衡量模型输出文本与专业参考文本的相似度，而 $n>1$ 时，BLEU 指标可以用来衡量句子的流畅性。由表 5 可知，文中所构建的 ATL 模型在 B-2 的评测中，语句流畅性较好；当以 B-4 进行评测时，文本流畅性有所降低，这是由于数据到文本生成领域中，由于数据结构化的特点导致生成短句的优势一般远大于长句，因此，当评测的词组集合越大，BLEU 指标也将呈现下降趋势。

3　结束语

提出了一种结合 Transformer 模型与深度神经网络的数据到文本生成方法，该方法包括基于多层感知器和注意力机制的语句内容预选、基于 Transformer 模型的文本内容规划以及基于双向长短期记忆神经网络的文本生成器模块。在文本内容规划模块，还提出了一种用于内容规划的 Transformer Text Planning (TTP)算法。实验结果表明，在 CS、RG、CO、BLEU、PPL 等指标评测下，相比已有的模型，文中方法整体性能有所提高，尤其在 CO 指标得分上展现出了优势。所改进的内容规划是对生成文本的语义与句法的一致性的改进，由实验分析可知，文中方法增强了模型的推理能力，同时提升了文本的连贯性。团队会继续研究关于自然语言生成的相关任务，并期望在数据到文本的生成任务中对于长句的生成也展现其更好的性能。

参考文献：

[1] 曹娟，龚隽鹏，张鹏洲. 数据到文本生成研究综述[J]. 计算机技术与发展，2019，29(1)：80-84，89.
CAO Juan，GONG Junpeng，ZHANG Pengzhou. Review of data-to-text generation[J]. Computer Technology and Development，2019，29(1)：80-84，89.(in Chinese)

[2] Wiseman S，Shieber S，Rush A. Challenges in data-to-document generation[C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Stroudsburg，PA，USA：Association for Computational Linguistics，2017：2253-2263.

[3] Duboue P A，McKeown K R. Statistical acquisition of content selection rules for natural language generation[C]//

Proceedings of the 2003 conference on Empirical methods in natural language processing . Morristown, NJ, USA: Association for Computational Linguistics, 2003:121-128.

[4] Barzilay R, Lapata M. Collective content selection for concept-to-text generation[C]// Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing—HLT '05. Morristown, NJ, USA: Association for Computational Linguistics, 2005:331-338.

[5] Liang P, Jordan M I, Klein D. Learning semantic correspondences with less supervision[C]// Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-ACL-IJCNLP '09. Morristown, NJ, USA: Association for Computational Linguistics, 2009:91-99.

[6] Angeli G, Liang P, Klein D. A simple domain-independent probabilistic approach to generation[C]// Proceedings of the 2010 conference on empirical methods in natural language processing. Cambridge, Massachusetts: Association for Computational Linguistics, 2010: 502-512.

[7] Konstas I, Lapata M. Unsupervised concept-to-text generation with hypergraphs[C]// Conference of the North American chapter of the association for computational linguistics: human language technologies. Montreal, Canada: Association for Computational Linguistics, 2012: 752-761.

[8] Kondadadi R, Howald B, Schilder F. A statistical NLG framework for aggregated planning and realization[C]// Proceedings of the 51st annual meeting of the association for computational linguistics. Sofia, Bulgaria: Association for Computational Linguistics, 2013: 1406-1415.

[9] Sowdaboina P K V, Chakraborti S, Sripada S. Learning to summarize time series data[M].Computational Linguistics and Intelligent Text Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014: 515-528.

[10] Gkatzia D, Hastie H, Lemon O. Comparing multi-label classification with reinforcement learning for summarisation of time-series data[C]// Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2014:1231-1240.

[11] Mahapatra J, Naskar S K, Bandyopadhyay S. Statistical natural language generation from tabular non-textual data[C]// Proceedings of the 9th International Natural Language Generation Conference. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016:143-152.

[12] Hongyuan M, Mohit B, Matthew R W. What to talk about and how? Selective generation using LSTMs with coarse-to-Fine alignment[J/OL]. arXiv: Computation and Language, 2015(2015-09-02)[2019-09-25]. http://www.oalib.com/paper/4051216#.

[13] Lebret R, Grangier D, Auli M. Neural text generation from structured data with application to the biography domain[C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2016:1203-1213.

[14] Qin G H, Yao J G, Wang X N, et al. Learning latent semantic annotations for grounding natural language to structured data[C]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018:3761-3771.

[15] Dzmitry B, Kyunghyun C, Yoshua B. Neural machine translation by jointly learning to align and translate[J/OL]. arXiv: Computation and Language, 2014[2019-09-25]. https://arxiv.org/abs/1409.0473.

[16] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]// Advances in neural information processing systems, Montreal:NIPS, 2017:5998-6008.

[17] Klein G, Kim Y, Deng Y T, et al. OpenNMT: open-source toolkit for neural machine translation[C]// Proceedings of ACL 2017, System Demonstrations. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017:62-72.

[18] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]// ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, USA : Association for Computational Linguistics 2002: 311-318.

[19] Puduppully R, Dong L, Lapata M. Data-to-text generation with content selection and planning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33: 6908-6915.

(编辑 陈移峰)