面向三元组的文本生成方法综述

摘要：文本生成是自然语言处理中最重要的子领域之一，旨在从各种形式的输入数据（包括文本、图像、表格和知识图谱）中生成生成可信的、可读的人类语言文本。数据到文本生成的任务旨在输入结构化的表格或知识图谱三元组生成有关结构化数据的描述性文本，目前的研究工作主要是使用基于规则和模板方法、基于神经网络模型、以及基于预训练语言模型的生成方法生成文本。本文围绕文本生成领域中的三元组到文本的生成任务，系统总结了近几年来三元组到文本生成的相关研究，并且对现有数据到文本的生成方法研究存在的瓶颈和未来工作方向进行讨论。

关键词：数据到文本的生成、RDF 三元组、自然语言处理

Abstract: Text generation is one of the most important subfields in natural language processing. It aims to generate trusted and readable human language text from various forms of input data, including text, images, tables and knowledge graphs. The task of data-to-text generation is to input structured tables or knowledge graph triples to generate descriptive text about structured data. The current research work mainly uses rules and template-based methods, neural network models, and pre-training language models to generate text. This paper focuses on the task of triple-to-text generation in the field of text generation, systematically summarizes the recent research on data-to-text generation, and discusses the existing bottlenecks and future work directions of the research on the generation method of data-to-text.

Keywords: Data-to-text generation, RDF triples, Natural language processing

1 引言

文本生成，也称为自然语言生成（Natural Language Generation，NLG），是自然语言处理（Natural Language Processing，NLP）中最重要的子领域之一。它旨在从各种形式的输入数据（包括文本、图像、表格和知识图谱[1]）中生成合理且易于理解的自然语言文本。近年来，文本生成技术已经广泛应用于各种的应用。例如，对话系统[2]即生成对用户提问的回答，机器翻译[3]即将文本从一种语言翻译成另一种语言，文本摘要[4]即生成一段长文本的简要摘要等应用。文本生成的基本目标是学习从输入数据到输出文本的映射函数。根据输入数据属性和类型的不同，又可细分为无条件文本生成、主题到文本的生成、数据到文本的生成、多

媒体到文本的生成、文本到文本的生成等类别。早期的方法通常采用统计语言模型来建模单词的条件概率[5]，这种统计方法可能会受到数据稀疏问题的影响。随着深度学习技术的出现和发展，神经网络模型已经成为了文本生成的主流技术，并在生成自然语言文本方面取得了巨大的进步。深度神经网络生成模型通常采用序列到序列框架[6]，基于编码器-解码器方案生成目标文本，编码器首先将输入序列映射为固定大小的低维向量（称为输入嵌入），然后解码器根据输入嵌入进行解码生成目标文本。为了提高文本生成模型的性能，注意力机制[7]和复制机制[8]等机制被提出并广泛运用。深度神经网络生成模型的优点是能够端到端地学习从输入数据到输出文本的语义映射，而无需劳动密集型的特征工程进行干预，并且能够有效地缓解数据稀疏性问题[9]。

虽然文本生成的深度神经网络模型取得了成功，但仍然存在瓶颈，大多数文本生成方法需要大量手动标记的高质量数据，这使得在缺乏高质量标记数据的领域无法进行模型的应用。目前大多数用于文本生成任务的标记数据集的数据量级都较小，导致深度神经网络可能会在这些小数据集中出现过度拟合，不能很好地泛化。因此预训练语言模型（Pre-trained Language Model，PLM）的出现改变了这个局面[10]，其基本思想是首先在大规模无监督语料库上预训练模型，然后在下游监督任务中对这些模型进行微调，以达到最先进的效果。随着 Transformer 的出现[11]和计算机计算处理能力的提升，预训练语言模型架构在 NLP 领域飞速发展并在多种应用中获得最先进的效果，例如 Google 提出的 BERT[12]模型和 OpenAI 提出的 GPT[13]模型。大量的实践工作证明了预训练语言模型可以将语料库中的大量语言知识编码为大规模的模型参数，并学习其中的上下文表示和通用语义，因此在下游任务中可以避免从头开始训练新模型，从而提升模型效率和性能。由于预训练语言模型在 NLP 任务中掀起了热潮，研究人员提出了基于预训练语言模型来解决文本生成任务的方法[15]。在大规模语料库上进行预训练，预训练语言模型能够准确理解自然语言并进一步流利地用人类语言特点进行文本表达，这对文本生成任务是非常有益的。因此利用预训练语言模型框架实现自然语言文本的生成在学术界和工业界都有着广阔的研究前景。

本文围绕文本生成领域中的三元组到文本的生成任务，系统总结了近几年来三元组到文本生成的相关研究。首先对数据到文本生成的任务分类和任务定义进行介绍，然后围绕基于规则和模板方法、基于神经网络模型、以及基于预训练语言模型的生成方法进行系统介绍和总结，最后对现有数据到文本的生成方法研究存在的瓶颈和未来工作方向进行讨论。

2 文本生成任务

2.1 文本生成任务定义

一般来说，一个长度为 n 的文本序列可以表示为序列 $y =< y_1, y_2, \ldots y_n >$，其中$y_1$是从词汇库 V 中抽取出来的。文本生成的任务旨在生成流畅且正确的自然语言文本。在大多数情况下，文本生成任务可以根据输入的数据，例如文本、图像、表格和知识库等等，生成满足某些语言特性的语言文本，例如流畅性、自然性和连贯性。将输出文本所需的特征定义为特征集 K。因此，文本生成的任务可以定义为：

$$y = f_K(x) \tag{2-1}$$

其中f_K表示根据输入 x 生成特定文本属性 K 的文本序列 y 的生成函数。

文本生成任务根据输入数据的属性和类型的不同，又可细分为无条件文本生成即输入随机噪声或不提供输入、主题到文本的生成即输入主题词或情感标签输出相关主题的文本描述、数据到文本的生成即输入结构化的表格或知识图谱三元组生成有关结构化数据的描述性文本、多媒体到文本的生成即输入一段语音或图像生成相关的语音识别或图像字幕、文本到文本的生成即输入一种语言机器翻译输出另一种语言或者输入某各问题机器生成问题的答案等。

2.2 数据到文本生成的任务定义

从概念上讲，知识图谱就是通过“关系”或“属性值”将“实体”节点联结起来，直观的展现事物概念之间的关系和属性，构成庞大的语义网络图结构，实现对现实世界的结构化描述。RDF 三元组是知识图谱的基本组成部分，它是由实体关系或概念属性构成的，其基本展示形式可以形式化的描述为{实体，关系，实体}或者{概念，属性，属性值}形式，即形成三元组的表达方式。因此结构化的三元组数据到文本的生成任务就是选取以三元组$< S, R, O >$为输入，其中 S 表示头实体、R 表示实体间的关系、O 表示尾实体，生成一句以流畅性、丰富性、准确性为特性的自然语言文本序列 $y =< y_1, y_2, \ldots y_n >$。如图 2-1 示例中，给定三元组<Barack Obama, president, United States>，生成符合三元组事实并且流畅的自然语言文本“Barack Obama is the 44th President of the United States.”。

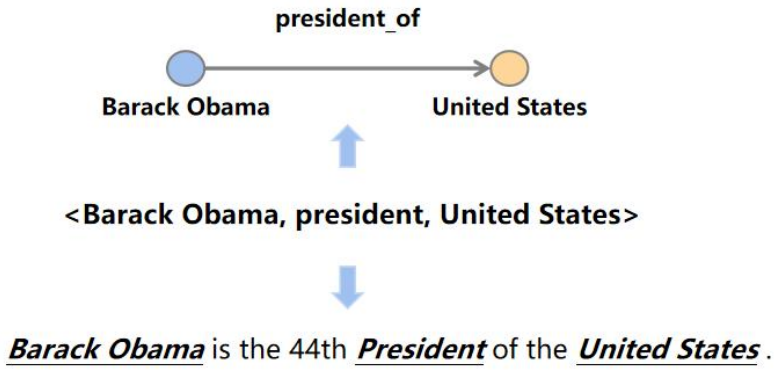

图 2-1 三元组到文本生成样例

2.3 数据到文本生成的研究概述

数据到文本的生成是一个长期被研究者们关注的一个研究领域[16]，旨在生成结构化数据的自然语言描述，包括对表格、ARM 图、知识图谱等结构化输入的自然语言生成任务。RDF 三元组作为结构化知识的基本表示单位，能够更好的展现结构化知识的细粒度组成结构和内容，并且能够嵌入图信息展现为有向图的形式（如图 2-1），更加直观的展示了细粒度节点间的关系。面向三元组的文本生成任务研究一直是研究者们关注的焦点，传统系统主要建立在基于规则和模板的算法上[17][18]。基于规则和模板的方法能够更好地遵循人类指定的文本特性进行文本生成，但是也意味着需要更多的人工参与，从而失去了文本生成的流畅度和多样性，并且这些早期的工作通常集中在较小的特定领域的数据集上，使得文本生成模型的实际应用存在局限。随着深度学习的快速进展，研究人员已经将注意力转移到神经网络结构的生成模型上，尤其是编码-解码（Encoder-Decoder）框架的流行，使得 Seq2Seq（Sequence to Sequence）模式成为了端到端的文本生成技术中的主流方法。

尽管神经模型在文本生成方面取得了成功，但大规模数据集的可用性仍然是主要的性能瓶颈。大多数监督文本生成任务的现有数据集都相当小，深度神经网络通常有大量的参数需要学习，这些参数很可能在这些小数据集上过拟合，并且在实践中不能很好地泛化。因此近年来，预训练语言模型的范式正在蓬勃发展。主要思路是首先在大规模语料库中预训练模型，然后在各种下游任务中对这些模型进行微调，以获得最先进的结果。近两年研究人员发现，扩展通过对模型大小或者数据集大小进行扩展能够有效地提高下游任务的模型性能。许多研究已经通过训练更大的语言模型[19][20]来探索性能极限，这些大语言模型（Large Language Model，LLM）在解决一系列复杂的 NLP 任务时展现出惊人的能力。

3 基于规则和模板的文本生成方法

3.1 基于规则和模板的文本生成过程

传统上基于规则和模板的数据到文本的生成方法需要解决两个主要问题[21]：内容规划和句法实现，即先规划出符合语义的文本结构，然后根据该结构进行最终文本的实现。Reiter 等人[22]又提出了更加细化的数据到文本生成的范式结构，可以归纳为一个四阶段的管道式（pipeline）结构：数据分析、数据解释、句子规划以及句法实现。具体来说，数据分析是指读取并解析输入的数据，识别输入数据的模式；数据解释是指从已识别出的模式和趋势中识别出更加复杂的特定领域信息，并且识别信息之间的因果关系和其他关系；句子规划是指在生成的文本中如何规划前面解析出的信息，并围绕这些信息创建文本和修辞结构；句法实现是指创建并生成自然语言文本来实现前面规划出的句子结构和信息。前两部分可以归结为“说什么”的选择，即产生结构化信息的文本计划，第三步句子规划则可以归结为“如何说”，因此最后一步则是将前面的规划“说出来”，通过运用句法和形态规则，以语法正确的方式生成最后的自然语言文本。

3.2 主要方法介绍

对于文本规划阶段，Stent 等人[23]使用自动适应应用领域的通用语言知识，提出了一种可训练的句子规划器 SPaRKy，展示了一种生成连贯句子计划的方法，根据偏好操作概率分布来进行随机选择，为每个文档计划详尽地生成多达 20 个句子计划树，并手工标记它们，然后将不同的论点子句组合起来，因此根据不同的操作，语法树可以生成多种句子计划，然后由统计排名器选择最佳计划。由于数据库中丰富的结构信息可以很容易地用于确定不同数据库条目之间的语义相关性，Barzilay 等人[24]提出了一种从语料库及其相关数据库中自动学习内容选择规则的有效方法，将内容选择视为一个集体分类问题，而不是单独进行某个内容的分类，集体选择能够优化生成文本的连贯性，使得语义相关的内容能够一起被选中，与上下文不可知的方法相比，该方法更好地捕获输入项之间的上下文依赖关系。从生成决策器的出发，Paiva 等人[26]通过将大量文本的自由生成过程中的内部决策与输出结果的文本风格特征相关联，利用文本相关性来控制生成器，这与以前大多数基于经验的生成方法形成对比，提供了一种更有效的语言转换模型，总体方法有两个阶段，首先是计算控制参数，即确定一组相关方程，这些方程捕获了生成文本的表面语言特征与产生这些文本的内部生成器决策之间的关系，然

后是在线应用到生成，利用这种相关性指导生成器生成具有特定语言特征的文本。但是上述方法都需要一个现有的手工句子规划器，因此 Dušek 等人[27]提出了基于 A*搜索的感知排序器，不需要在训练数据中进行细粒度对齐，而是使用深度依赖语法进行句子计划，生成器将对齐学习纳入句子规划器训练，并使用深度语法树和基于规则的表面实现步骤，以确保输出的语法正确性，与之前的方法不同的是该生成器不需要手工制作的基本句子规划器。

对于句法实现阶段，Lapata 等人[25]设计了一个符合人类书写顺序的句子排序概率模型，提出了一种适合于文本到文本生成的信息排序方法，描述了一个从特定领域文本的语料库中学习句子顺序约束的模型。针对不同规则模块和手写的规则生成若干个候选的生成文本，然后利用统计模型对这些候选文本进行排序，并且提出了一种自动评估模型产生的文本质量的方法。Dethlefs 等人[28]将句法实现作为序列标记任务，并将条件随机场（CRF）的使用与语义树相结合。提出了扩展的语境概念，其他句法实现方式容易受到局部特征的限制，CRF 能够考虑到整体的话语语境，适用于在丰富语义的上下文中进行建模，并结合语义树来获取丰富的语言信息，实现了更自然和低重复的句法实现效果。这种组合能够在多个话语中跟踪句法、语义和词汇特征之间的依赖关系。

还有其他工作将文本规划和句法实现结合到一个步骤中，不需要手工制作基础模块。Angeli 等人[29]提出了一个简单健壮的生成系统，它在一个统一的、独立于领域的框架中执行内容选择和文本实现，使用一个端到端的过程，并将其中的子任务分解为一系列局部决策和分层排列，对每个决策都进行了判别性训练，从数据库记录中生成文本，逐步选择数据库记录、字段和相应的文本实现来描述它们，实现了一个具有领域通用性的生成系统。Dušek 等人[30]提出了一个基于序列到序列方法的自然语言生成器，结合集束搜索和 n-best 列表重新排序器来控制输出文本中的相关信息，并且通过实验比较了通过深度语法树的两步生成方法与直接生成字符串的比较，实验结果表明，直接生成文本方法更好，并且基于 n-gram 的分数更高。

3.3 方法总结

	研究内容	方法	方法内容	年份
1		SPaRKy[23]	提出了一种可训练的句法规划树并可控生成多种句式	2004
2		Barzilay 等人[24]	从数据库中自动学习内容选择规则并研究集合分类问题	2005

3	文本规划	Paiva 等人[26]	捕获生成文本的表面语言特征，利用文本相关性来控制生成器	2005
4		Dušek 等人[27]	使用深度依赖语法树并提出基于 A*搜索的感知排序器	2015
5	句法实现	Lapata 等人[25]	提出了一种无监督概率模型，从大型语料库中学习排序约束	2003
6		Dethlefs 等人[28]	将条件随机场与语义树相结合实现语境的扩展	2013
7	文本规划和句法实现联合实现	Angeli 等人[29]	独立于领域的框架中执行内容选择和文本实现	2010
8		Dušek 等人[30]	提出了一个基于序列到序列方法的自然语言生成器	2016

3.4 目前存在的问题

一些早期的工作通常集中在较小的特定领域的数据集上，使得文本生成模型的实际应用存在局限。由于管道式（pipeline）结构特点，当某个模块产生一些误差，误差传播会导致最后的生成结果不尽人意。

基于模板的生成使没有受过语言训练的程序员能够编写一个文本生成器，该生成器可以针对不同的语境有效地生成高质量的输出。然而，特定领域或特定对话上下文的输出可能无法基于一个统一的模板，因此它需要大量的人工干涉来为每个应用程序重新手工创建模板，开发特定于领域的规则或针对特定领域调整一般规则，还需要耗费资源设计和维护不同语境下的规则模板，从而也失去了文本生成的流畅度和多样性。

4 基于神经网络模型的文本生成方法

4.1 主要结构

神经网络方法只要有足够的训练数据，就有可能学习从输入到输出的直接映射。因此神经网络模型的文本生成方法大都基于端到端的数据-文本模型，直接学习输入-输出映射，而较少的依赖于显式的中间环节表示。基于编码器-解码器方案生成目标文本，编码器首先将输入序列映射为固定大小的低维向量（称为输入嵌入），然后解码器根据输入嵌入进行解码生成目标文本。优点是能够端到端

地学习从输入数据到输出文本的语义映射，而无需劳动密集型的特征工程进行干预。但是神经网络端到端方法被证明并不一定比基于神经网络的管道模型更好[31]，因为为特定任务开发专用的神经模块可以在每个连续的任务上获得更好的性能，并且将它们组合起来可能会产生更好、更可重用的输出结果。

4.2 神经网络端到端模型的方法介绍

神经网络的重新兴起的部分原因是硬件的进步，可以支持资源密集型问题的学习，并且神经网络利用反向传播来学习特征表示，具有密集的、低维的和分布式的特点，使得它们非常适合捕捉语法和语义的特征[32]。循环神经网络（Recurrent Neural Network，RNN）可以在端到端训练过程中自动捕获高度复杂的模式，因此在自然语言生成中扮演着重要的角色；同时具有长短期记忆单元的神经网络（Long Short-Term Memory，LSTM）的序列在先验建模方面也取得了显著的成功。与标准语言模型相比，它们的主要优点是它们处理不同长度的序列，同时通过将历史投影到低维空间，避免了数据稀疏性和参数数量的爆炸。Sutskever 等人[33]探究了循环神经网络对自然语言生成的潜在效用，他们使用了字符级的 LSTM 模型来生成语法英语句子，还引入了一种新的 RNN 变体实现字符级语言建模，它使用乘法（或“门控”）连接，允许当前输入字符确定从一个隐藏状态向量到下一个隐藏状态向量的转换矩阵。

编码器-解码器框架的流行，使得序列到序列（Seq2Seq）模式成为了端到端的文本生成技术中的主流方法，即将源语言中的可变长度输入序列映射到可变长度序列的目标语言中。编码和解码之间的这种解耦使得模型可以在多任务学习环境中跨多个 NLP 任务共享编码向量。基于注意力机制的模型结构的提出加速了编码器-解码器结构的发展，即在解码期间预测输出的某些部分时，对输入编码的部分有选择的进行关注。Bahdanau 等人[7]引入了对编码器-解码器模型的扩展，该模型可以学习对齐和翻译，每次提出的模型在翻译中生成一个单词时，它会软搜索源句子中最相关的信息集中的一组位置。然后，该模型根据与这些源位置和之前生成的所有目标单词相关的上下文向量预测目标单词。这种方法与基本编码器-解码器最重要的区别在于，它不试图将整个输入句子编码为单个固定长度的向量。而是将输入的句子编码成一个向量序列，并在解码翻译时自适应地选择这些向量的子集。Liu 等人[34]提出了一种结构感知的 seq2seq 结构，该结构由领域门控编码器和双注意描述生成器组成。在编码阶段，我们通过字段门及其相应的字段值更新 LSTM 单元的单元内存，以便将字段信息合并到表示中。在解码阶段，提出了包含词级注意和字段级注意的双注意机制，对生成的描述与表之间的语义关联进行建模。

除了注意力机制，复制机制也是端到端的数据到文本生成任务的重要技术之一，复制机制首先被引入到循环神经网络中，以产生完全由输入序列元素组成的输出序列，Gu 等人[35]将复制引入到基于神经网络的 Seq2Seq 学习中，并提出了一个具有编码器-解码器结构的新模型 COPYNET，利用复制机制实现了输入序列中的某些片段被选择性地复制到输出序列中，不仅能够正常生成单词，而且能够复制输入序列的适当片段。See 等人[8]基于复制机制提出了一种指针生成模型，以两种正交的方式增强了标准的序列到序列的注意力模型。首先使用了一个混合的指针生成器网络，它可以通过指向源文本中的单词来进行内容的复制，这有助于准确地复制输入序列的信息，同时保留了通过生成器产生新单词的能力。其次使用覆盖来跟踪已经总结的内容，这阻止了输出内容的重复。

由于三元组之间并不存在时序关系，而更多的是图的结构和节点之间的边，因而基于图结构的 GTR-LSTM[36]被提出，Distiawan 等人为了尽可能多地保留 RDF 三元组中的信息，提出了一种新的基于图的三元组编码器，三元组编码器不仅对三元组的元素进行编码，而且还对三元组内部和三元组之间的关系进行编码。因此充分考虑了元素间、三元组之间的联系，更好地实现 RDF 三元组到文本的生成任务。但由于编码器组件仍然是基于循环神经网络，会忽视异构数据的结构化信息，往往不能充分捕捉到实体和关系之间丰富且复杂结构信息。Marcheggiani 等人[37]提出了一种基于图卷积神经网络（Graph Convolutional Network，GCN）的编码器更好地利用了输入的三元组之间的结构信息，允许将图形数据显式编码到神经网络中，作者在实验中比较了 LSTM 顺序编码器和基于 GCNs 的结构化数据编码器，实验结果表明使用 GCNs 显式编码结构信息对顺序编码是有益的。同时考虑到图结构中相邻节点的影响大小，解决由于信息过长导致信息丢失的问题，放大节点中重要部分的影响，Koncel-Kedziorski 等人[38]使用了图注意力网络（Graph Attention Network, GAT）来实现图到文本的生成，由于 GCN、GAT 这类模型仅考虑图谱中已出现相邻节点的信息而忽略全局信息，所以作者添加了全局结点使得模型能够利用更为全面的全局信息，充分利用这些知识图的关系结构而不强加线性化或分层约束。

4.3 神经网络和管道模型相结合的方法介绍

许多研究人员试图在神经数据到文本生成模型中对内容规划进行明确的建模，将管道式的生成方法与神经网络模型的优点相结合，进一步提升数据到文本的生成质量。由于神经网络的方法通常不会对文本生成过程中的内容顺序进行建模，但是参考人类的书写风格，在书写前需要对内容的顺序进行规划，因此 Sha

等人[39]提出了一个顺序规划的文本生成模型，其中文本的顺序信息通过基于链接的注意力显式捕获，然后，自适应门将基于链接的注意与传统的基于内容的注意结合起来，实现文本内容的规划和输出，并且使用了复制机制来解决生僻字的问题。Puduppully 等人[40]提出了两阶段的管道式生成方法，该管道使用基于注意力机制的指针网络从结构化数据中选择和规划重要信息，然后使用另一种编码器-解码器模型生成最终文本。首先生成一个内容计划，突出显示应该提到哪些信息以及以何种顺序进行书写，然后在考虑内容计划的同时生成文档。Moryossef 等人[41]将生成过程划分为一个依赖于输入的符号文本规划阶段和一个只关注句子实现的神经生成阶段，并且提出了一种将参考文本格式化为相应文本计划的方法，实现了为输入选择高质量文本计划的方法。文本规划与神经实现解耦确实提高了系统的可靠性和充分性，同时保持了流畅的输出。图神经网络可以更好地对输入图进行编码，但会扩大编码器和解码器之间的结构差距，使忠于事实的生成变得困难。Zhao 等人[42]为了缩小这一差距，提出了 DUALENC，这是一种双重编码模型，不仅可以结合图结构，还可以满足输出文本的线性结构，即使用两个 GNN 编码器分别负责提取语义和文本规划，从而更好地利用输入结构和信息，双编码结构可以显著提高生成文本的质量。

Chen 等人[43]提出了一个具有动态内容规划的神经数据到文本生成模型，在计划应该提到哪些数据时考虑先前生成的文本，从给定的结构化数据中利用先前生成的文本来动态地选择适当的内容。模型的动态内容规划机制易于与编码器-解码器框架集成，并具有自己的目标函数。此外 Puduppully 等人[44]也从同样的角度出发提出了一个神经模型，该模型具有宏观规划阶段和生成阶段的两段式计划，宏观计划表示重要内容（如实体、事件及其相互作用）的高层组织，这些信息从输入数据中学习然后输入到文本生成器中，实现了对输出文本的更好控制的同时发挥了编码器-解码器架构的优势。

4.4 方法总结

	研究内容	方法	方法内容	年份
1	端到端模型	Sutskever 等人[33]	引入了一种新的 RNN 变体实现字符级语言建模	2011
2		Bahdanau 等人[7]	基于注意的模型来学习基于输入和输出文本的松散耦合对齐	2015
3		Liu 等人[34]	结构感知的端到端模型由领域	2018

			门控编码器和双注意描述生成器组成	
4		COPYNET[35]	将复制机制应用于具有编码器-解码器结构的新模型	2016
5		pointer-generator[8]	利用混合的指针生成器网络和覆盖追踪的方式生成文本	2017
6		GTR-LSTM[36]	基于图结构信息的三元组 LSTM 编码器	2018
7		Marcheggiani 等人[37]	基于图卷积神经网络的编码器显式编码结构信息	2018
8		GraphWriter[38]	使用了图注意力网络并添加全局结点来实现图到文本的生成	2019
9	神经网络和管道模型相结合	Sha 等人[39]	提出了一种表到文本生成的顺序规划方法	2018
10		Puduppully 等人[40]	提出了两阶段的管道式生成方法并保留端到端的训练	2019
11		Moryossef 等人[41]	生成过程划分为符号文本规划阶段和神经生成阶段	2019
12		DUALENC[42]	使用两个 GNN 编码器分别负责提取语义和文本规划	2020
13		NDP[43]	具有动态内容规划的神经数据到文本生成模型	2021
14		Puduppully 等人[44]	将宏观规划阶段和生成阶段的两段式计划与端到端模型结合	2021

4.5 目前存在的问题

虽然数据到文本生成的深度神经网络模型取得了成功，但仍然存在瓶颈，大多数文本生成方法需要大量手动标记的高质量数据，这使得在缺乏高质量标记数据的领域无法进行模型的应用。目前大多数用于文本生成任务的标记数据集的数据量级都较小，导致深度神经网络可能会在这些小数据集中出现过度拟合，不能很好地泛化。端到端的文本生成方法虽然具有流畅、正确的语法结构，但生成的文本内容偏离输入数据，产生重复、省略和不忠实等问题，这些问题在传统的规则和模板框架中不太可能发生，因此基于神经网络方法的文本生成仍然欠缺实用

性，存在不足和改进空间。

5 基于大模型的文本生成方法

5.1 主要过程

预训练语言模型使用大量未标记的文本数据进行预训练，可以在下游生成任务中进行微调。预训练语言模型在大规模语料库上进行预训练，将大量的语言特点和语境知识编码成大量的参数存储起来，增强了对语言的理解，从而提高了生成质量。预训练的想法是从人类书写文本的角度得到启发的，也就是说，我们迁移和利用过去学过的旧知识来理解新知识并处理各种新任务。通过这种方式，预训练语言模型可以利用模型学习到的旧经验和知识来执行新任务。

5.2 基于预训练语言模型的生成方法介绍

5.2.1 生成模型

由于 Transformer 模型[11]取得的巨大成就，几乎所有的预训练语言模型都采用了 Transformer 的骨干。对于文本生成任务，一些预训练语言模型遵循标准 Transformer 架构实现编码器-解码器框架，还有一些预训练语言模型只应用 Transformer 的解码器架构。由于 Transformer 编码器的预训练任务与下游生成功能之间的差异，因此很少用于文本生成任务，而是广泛用于自然语言理解。

标准 Transformer 使用编码器-解码器架构，该架构由两个 Transformer 块堆栈组成。编码器以输入序列为源输入，而解码器则基于编码器-解码器的自注意机制生成输出序列。基于上述架构，MASS[45]、T5[46]和 BART[47]等模型的提出显著提高了文本生成的质量。Song 等人[45]提出了基于编码-解码器结构的掩码序列到序列预训练语言生成模型 MASS，MASS 采用编码器-解码器框架，在给定句子剩余部分的情况下，重构一个句子片段，它的编码器以一个随机屏蔽片段（几个连续的 token）的句子作为输入，它的解码器尝试预测这个被屏蔽的片段内容。通过这种方式，MASS 可以联合训练编码器和解码器来开发特征提取和语言建模的能力。Raffel 等人[46]在迁移学习的启发下，引入一个将所有基于文本的语言问题转换为文本到文本格式的统一框架，基本思想是将每个文本处理问题视为“文本到文本”问题，即将文本作为输入并生成新文本作为输出，同时通过扩展模型和数据集来探索 NLP 迁移学习的前景和局限性。Lewis 等人[47]提出了用于序列到序列预训练模型的去噪自编码器，训练过程中首先用任意的噪声函数破坏文本，使

用了新颖的填充方案来学习一个模型对原始文本进行重建，BART 在文本生成任务和文本理解任务上都能获得不错的效果。

基于 Transformer 的解码器架构，GPT 系列模型被提出，它们使用单向的自注意力掩码模型，每个 token 只能关注前一个 token 信息。GPT-1[48]探索了一种结合无监督预训练和监督微调的半监督方法来完成语言理解任务，学习了一种通用的表征，这种表征可以在很少的为微调情况下转移到广泛的下游任务中。同样采用两阶段的训练过程，首先在未标记数据上使用语言建模来学习神经网络模型的初始参数，然后使用相应的监督目标将这些参数适应于目标任务。GPT-2[13]探索了语言模型对零样本生成任务的迁移能力，突出了数据体量的重要性，演示了语言模型可以在零样本设置中执行下游任务的潜力，且不进行任何参数或架构修改。GPT-3[49]表明海量模型参数可以显着改善下游生成任务，扩展语言模型的参数量可以极大地提高任务不可知的、少样本情况下的模型性能，有时甚至可以与先前最先进的微调方法相竞争。对于所有任务，GPT-3 的应用没有任何梯度更新或微调，但在许多 NLP 数据集上取得了出色的表现，包括翻译、问答和填空任务。

Kale 等人[51]还研究了 T5 形式的文本到文本预训练与管道式的神经架构以及其他预训练模型的效果对比，实验结果表明 T5 形式的文本到文本预训练使简单的端到端转换器模型优于为数据到文本生成量身定制的流水线神经架构，也优于基于掩码的语言模型预训练技术的 BERT 和 GPT-2 等模型。

5.2.2 优化方法

由于预训练语言模型已经在多个 NLP 领域中取得了非常显著的成功，因此学者们开始利用预训练语言模型和迁移学习来解决数据到文本的生成问题。目前有两种主流的迁移方法，对整个预训练语言模型进行微调[50]或使用前缀学习的轻量级微调方式[52]。

对于整个预训练语言模型进行微调方面，Chen 等人[50]提出了一种基于知识的预训练方法 KGPT，这是一种远程监督学习算法，利用大规模未标记的网络文本来预训练数据到文本模型。该方法由两部分组成，首先是基于知识的通用生成模型，用于生成包含丰富知识的文本，然后从网络上抓取大量基于知识的文本语料库的进行模型的预训练。预先训练的模型可以在各种数据到文本生成任务上进行微调，以生成特定于任务的文本。Deng 等人[54]研究了在少样本场景下的具有逻辑形式的结构化数据到文本的生成，提出了一个统一的逻辑知识条件文本生成框架 LOGEN，利用容易获得的领域内语料库来进行自我训练，使用少量种子数据进行训练以生成伪逻辑形式，并逐步扩大生成逻辑形式的训练语料库。LOGEN 还采用了两个关键组件来保证内容的一致性和结构的一致性。

基于神经的端到端方法从结构化数据或知识中生成自然语言，这需要大量数据，因此在数据有限的情况下难以将其应用于实际应用。一些研究者探索在小样本和零样本设置下数据到文本的生成任务。Chen 等人[55]利用预训练语言模型中的知识，对目标域中具有有限训练实例的数据到文本生成模型进行微调。模型体系结构的设计基于两个方面，分别从输入数据中选择内容和语言建模来组成连贯的句子，这些句子可以利用外部资源作为先验知识，显著减少人工标注的工作量。Chang 等人[56]采取了不同的途径，首先基于用同一类别的替代值替换特定值来生成新的文本样本，然后基于 GPT-2 生成新的文本样本，并且提出一种将新文本样本与数据样本配对的自动方法来自动增加可用于训练的数据。由于文本增强会给训练数据引入噪声，作者使用循环一致性作为目标，以确保给定的数据样本在被表述为文本后可以被正确地重构。Kasner 等人[57]的工作灵感来自于传统的管道式数据到文本系统，包括排序、聚合和段落压缩的阶段，并使用预训练的语言模型来实现这些阶段，避免在数据到文本生成的数据集上微调预训练语言模型，同时仍然利用预训练语言模型的文本实现能力。

关于前缀调优[52]，它为利用预训练的语言模型提供了一种有效的训练范式。具体来说，它冻结了预训练语言模型的参数，只调整了一组预先准备的可训练连续向量到语言模型，该模型由更少的参数组成。这些向量可以被认为是任务需求的“软”描述，以使语言模型适应给定的任务。在此基础上，Chen 等[70]提出在提示学习和适配学习之间架起桥梁，提出了 inducer-tuning 来扩展前缀调优的优势，将提示符（包括硬提示和软提示）视为内核方法中的“诱导变量”，将预训练的语言模型中的注意力模块与核估计器作类比，解释并激发了前缀调优的潜在机制。

其他优化视角方面，有的工作从模型可控性角度出发，由于预训练语言模型控制生成文本的属性变得很困难，Dathathri 等人[53]提出了用于可控语言生成的即插即用语言模型 PPLM，不修改模型架构或对特定属性数据进行微调，将预训练的语言模型与一个或多个简单的属性分类器相结合，这些分类器指导文本生成，而无需对语言模型进行任何进一步的训练。PPLM 模型十分灵活因为任何可微分属性模型的组合都可以用来引导文本生成，是指可以生成在提示之外的丰富性语言。同时现有的解决方案通常学习从输入中复制实体或三元组，缺乏对知识的选择和安排的整体考虑，容易造成语篇的不连贯。Li 等人[58]提出了一种基于知识图谱的连贯增强文本规划模型 CETP，以提高评论生成的全局和局部一致性。该模型学习了一个两级文本计划来生成文档，首先将文档计划建模为句子计划的顺序序列，然后将句子计划建模为来自知识图谱的基于实体的子图。子图可以通过实体之间的句内关联来自然地增强局部一致性，同时设计了一种具有子图和节点级关注的分层自关注架构以增强全局一致性。

5.3 方法总结

	研究内容	方法	方法内容	年份
1	基于Transformer的编码器-解码器架构的PLM	MASS[45]	基于编码-解码器结构的掩码序列到序列预训练语言生成模型	2019
2		T5[46]	将所有基于文本的语言问题转换为文本到文本格式的统一框架中	2020
3		BART[47]	实现了序列到序列的去噪自编码器预训练语言模型	2020
4	基于Transformer的解码器架构的PLM	GPT-1[48]	结合无监督预训练和监督微调的半监督方法来完成语言理解任务	2018
5		GPT-2[13]	探索了语言模型对零样本生成任务的迁移能力	2019
6		GPT-3[49]	实现了海量模型参数可以显着改善下游生成任务	2020
7	基于预训练语言模型的微调	KGPT[50]	利用大规模未标记的网络文本来预训练数据到文本模型	2020
8		LOGEN[54]	利用自我训练和基于内容和结构一致性的伪逻辑形式样本	2023
9	基于小样本和零样本设置	Chen 等人[55]	利用外部资源作为先验知识减少人工标注的工作量	2020
10		Chang 等人[56]	使用语言模型来生成新的文本来增加有限的语料库	2021
11		Kasner 等人[57]	通过使用排序、聚合和段落压缩一系列的训练模块来转换文本	2022
12	基于前缀调优	inducer-tuning[70]	将提示符（包括硬提示和软提示）视为内核方法中的“诱导变量”	2022
13	生成文本的可控性	PPLM[53]	提出了用于可控语言生成的即插即用语言模型	2020
14	生成文本的一致性	CETP[58]	提出了一种基于知识图谱的连贯增强文本规划模型	2021

5.5 目前存在的问题

首先在与数据相关的挑战中会缺乏足够的训练数据，在许多文本生成任务中，很难获得足够的注释数据。同时在研究领域的转移中，预训练语料库的数据容易产生偏差，预训练语料库是从网络中收集的，它可能包含来自不同领域的数据集，例如生物医学和法律语料库，因此将预训练语言模型应用于新领域时，尤其是与预训练的分布差异很大时，这些特定领域的数据很可能包含有偏差的信息，这会给新任务带来很大的挑战。此外一些特定的种族、性别的偏见，减轻词嵌入中性别偏见的一种简单方法是在生成词嵌入时“交换”训练数据中的性别术语，或者简单地掩盖名称和代词也可以减少偏见并提高某些语言任务的性能，目前并没有通用的统一方法。

从模型角度来看预训练语言模型架构也面临挑，尽管预训练语言模型在文本生成任务上取得了巨大成功，但主干 Transformer 模型的体量较大，导致高内存消耗，在资源匮乏的情况下计算开销和能源成本也成为问题。由于有研究表明[49]，预训练语言模型的性能与预训练语言模型参数规模有很大关系。因此研究者开始研究模型的扩展，对预训练语言模型的底层设计进行改进以提高文本生成的性能，最具代表性的工作是 GPT-3，它采用了 1750 亿个参数，在各种文本生成任务中实现强大的性能，而无需任何梯度更新或微调。但是在大规模文本数据上预训练并构建一个高效的预训练语言模型也十分非常昂贵。

6 数据集与评价方式

6.1 数据集

WikiBio（2016）[59]：WikiBio 是一个包含超过 70k 个例子的个人传记数据集，旨在为维基百科表生成第一句传记描述。输入是来自维基百科的信息框，输出是传记的第一句话。输出文本的平均长度为 26.1 个单词。

E2E（2017）[60]：E2E 数据集是根据餐馆的属性生成一些列餐馆描述文本。此数据集的数据以键值格式存储，结构化数据可以被看作是<属性、值>对。它由 42,061,547 和 630 个用于训练、验证和测试集的实例组成。平均输入长度为 28.5，平均输出长度为 27.8。

WebNLG（2017）[61]：WebNLG 挑战的意图是在于将 RDF 三元组生成流畅的自然

语言文本，由描述事实实体关系的 RDF 三元组和一个或多个人工生成的参考文本组成。WebNLG 数据集包含 9,674 组 RDF 三元组和 25,298 个参考文本，输入三元组个数不一，最多包含七个三元组输入集合，这些三元组是从 DBPedia 抽取出来的，每个参考文本还与它实现的三元组顺序配对。输入 x 是一个(主题，属性，对象)三元组的序列。平均输出长度为 22.5。测试数据跨越 15 个领域，其中 10 个是在训练中看到的。测试集由两部分组成，前半部分包含训练数据中看到的 DB 类别，后半部分包含 5 个未见过的类别。

AGENDA（2019）[62]：在这个数据集中，将知识图谱与从 ai 会议论文集中提取的科学摘要配对。每个样本包含论文标题、一个知识图谱和相应的摘要。知识图谱包含与科学术语相对应的实体，边缘表示这些实体之间的关系。由于图形是自动生成的，因此该数据集在图形和相应文本之间具有松散的对齐。模型的输入是包含标题、所有知识图谱实体序列和三元组的文本，目标文本为论文摘要。

ToTTo（2020）[63]：ToTTo 是一个开放域英语表到文本数据集，由维基百科表与人类编写的自然语言描述配对组成，含有超过 120,000 个训练样例，每个输入都是一个维基百科表和一组突出高亮显示的表单元格，需要模型生成一个完整的句子描述。

DART（2021）[64]：DART 是一个大型的开放域结构化数据记录到文本生成语料库集合，集成了来自 WikiSQL、WikiTableQuestions、WebNLG 2017 和 CleanedE2E 的数据。DART 数据集由 62659 对训练对、6980 对验证对和 12552 对测试对组成，并且涵盖了更多的类别，因为 WikiSQL 和 WikiTableQuestions 数据集来自开放域维基百科。与 WebNLG 数据集不同，这个 DART 数据集没有被分类为“可见”和“未见”，因此不会执行实体屏蔽。

6.2 评价方式

对于一个文本生成模型的生成文本质量的评估方式主要有人工评价和机器评价两种方式，通过一个客观、高效的评估手段不仅能够让研究者对于模型效果得到直观的认知，也能更有利于模型后续的改进和数据拟合，从而进一步推动文本生成的发展。但是人工对文本生成的评估是十分昂贵的，需要耗费大量的时间，甚至涉及到语言专家的资源需求。因此机器自动评价方式成为了更加快捷高效的方法，比较常见的有如下几种方法：

BLEU[65]：BLEU 是较早出现的、目前应用较为广泛的一种机器翻译自动评价标准，它能够快速得到评分、运行成本小、无需任何语言学知识，却与人工评估结论高度相关。它的核心思想是对机器生成的自然文本和人类提供的参考文本之间重复出现的词汇频率进行计算，采用一种 N-gram 的匹配规则，即根据不同的维度对相应的词汇进行匹配，相同词汇出现的频率越高则认为文本翻译的准确性越高。其计算公式如式 6-1 所示，首先计算修正的 N -gram 精度 p_n，然后通过 w_n 进行几何平均，根据机器生成文本长度 c 和参考译文长度 r 来计算长度惩罚因子 BP，最终得到 BLEU 得分。改进的 n-gram 精度可以更加充分全面的评估文本翻译质量，使用长度惩罚对生成的过长的语句进行惩罚，从而更客观的得到准确的翻译效果。

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{6-1}$$

$$\text{其中，}\ \mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \le r \end{cases}$$

$$w_n = 1/N$$

NIST[66]：虽然它仍然是一个类似于 BLEU 的基于 n-gram 的度量，但它引入了信息加权的 n 图精度倾向于那些信息量更大的 n 图。NIST 的计算方法如公式 6-2、6-3 所示，式 6-3 的第一个分数是生成文本相对于参考文本的信息加权(式 6-2)的 n-gram 精度。它给那些出现频率较低的 n 个图赋予了更多的权重，这些图被认为信息量更大。L_{gen} 为生成文本的长度，$\bar{L}_{ref}$ 为参考文本的平均长度。$\frac{L_{gen}}{\bar{L}_{ref}}$用于对模型生成的短文本进行处罚。当 $\frac{L_{gen}}{\bar{L}_{ref}} = \frac{2}{3}$ 时，令 N = 5，β 惩罚因子为 0.5。

$$\mathrm{Info}(w_1 \ldots w_n) = \log_2\left(\frac{\text{the number of occurrences of } w_1 \ldots w_{n-1}}{\text{the number of occurrences of } w_1 \ldots w_n}\right) \tag{6-2}$$

$$\mathrm{NIST} = \sum_{n=1}^{N} \frac{\sum_{\substack{\text{co-occurrence of} \\ \text{all } w_1 \ldots w_n}} \mathrm{Info}(w_1 \ldots w_n)}{\sum_{\substack{\text{all } w_1 \ldots w_n \\ \text{in generated text}}} (1)} \cdot \exp\left\{\beta \log^2\left[\min\left(\frac{L_{gen}}{\bar{L}_{ref}}, 1\right)\right]\right\} \tag{6-3}$$

METEOR[67]：虽然 N-gram 匹配倾向于在系统生成的文本和目标文本之间执行

精确的字符串匹配，但 METEOR 使用 WordNet 匹配同义词，因为同义词的含义是相同的。此外，它还建议将单词组织成词语块，并以此来确定单词的有序程度。METEOR 的计算方法如公式 6-4 所示，F_{means}是调和平均值，它赋予召回率更大的权重，P 是单图精度，R 是单图召回率，其中$\propto$为可调控的参数，m 为候选翻译中能够被匹配的一元组的数量，c 为候选翻译的长度，r 为参考摘要的长度。惩罚指数是根据语句块的数量计算的，用来衡量单词的有序程度。

$$\mathrm{MENTOR} = (1 - \mathrm{Penalty}) \times F_{means} \quad (6\text{-}4)$$

$$\text{其中，} F_{means} = \frac{PR}{\propto P + (1-\propto)R}$$

$$P = \frac{m}{c}$$

$$R = \frac{m}{r}$$

ROUGE$_L$[68]：与 BLEU 不同，该指标主要关注模型的召回性能。此外，它使用最长公共子序列（LCS）来匹配系统生成的文本和参考文本。ROUGE$_L$ 的计算公式如公式 6-5 所示，其中，X 表示候选摘要，Y 表示参考摘要，LCS(X, Y)表示候选摘要与参考摘要的最长公共子序列的长度， m 表示参考摘要的长度， n 表示候选摘要的长度。

$$\mathrm{ROUGE_L} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \quad (6\text{-}5)$$

$$\text{其中，} R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n}$$

CIDEr[69]：与 NIST 分数类似，该指标也关注信息更丰富的 token。不同的是，它使用术语频率逆文档频率（TF-IDF）来达到目的，因为它将给予语料库中不经常出现但信息丰富的单词更多的权重。CIDEr 的计算方法如公式 6-6，把每个句子看成文档，然后计算其 TF-IDF 向量的余弦夹角，据此得到候选句子和参考句子的相似度。其中， c 表示候选标题， S 表示参考标题集合， n 表示评估的是 n-gram， M 表示参考字母的数量， $g^n(\cdot)$表示基于 n-gram 的 TF-IDF 向量。

$$\mathrm{CIDEr_n}(c,S) = \frac{1}{M}\sum_{i=1}^{M}\frac{g^n(c)\cdot g^n(S_i)}{\|g^n(c)\|\times\|g^n(S_i)\|} \quad (6\text{-}6)$$

7 未来研究方向

本文概述了当前用于数据到文本生成的代表性研究工作，并期望它可以促进未来的研究。系统总结了近几年来三元组到文本生成的相关研究。首先对数据到

文本生成的任务分类和任务定义进行介绍，然后围绕基于规则和模板方法、基于神经网络模型、以及基于预训练语言模型的生成方法进行系统介绍和总结。

为了推进这一领域的发展，目前仍然存在几个未解决的问题和未来的方向。首先在文本生成的可控性方面，使用预训练语言模型生成可控文本是一个有趣的方向，但仍处于非常早期的阶段。因为预训练语言模型通常在通用语料库中进行预训练，难以控制生成文本的多粒度属性（情绪、主题和连贯性）。然后在模型优化方面，微调是将预训练语言模型中学到的语言知识提炼并应用到下游生成任务的主要优化方式。目前，基于提示的学习已经成为一种高性能、轻量级的优化方法。未来的工作可以探索更多可以结合当前方法优点的优化方法。其次在语言相关性方面，如今几乎所有用于文本生成的模型都主要基于英语，这些模型在处理非英语生成任务时会遇到挑战。因此，与语言无关的语言模型值得研究，它需要捕获跨不同语言的通用语言和语义特征。例如如何重用现有的基于英语语言模型来生成非英语语言的文本。最后在道德问题方面，目前的语言模型是从网络上抓取的大规模语料库上进行预训练的，没有进行细粒度过滤，这可能会导致伦理问题，例如生成有关用户的私人内容。因此，研究人员应尽最大努力防止滥用语言模型。此外预训练语言模型生成的文本可能存在偏见，例如训练数据在性别、种族和宗教维度上的偏见[19]。因此，我们应该对语言模型实行干预措施以防止此类偏差。

参考文献

[1] Singhal A. Introducing the knowledge graph: things, not strings[J]. Official google blog, 2012, 5: 16.

[2] Zhou L, Gao J, Li D, et al. The design and implementation of xiaoice, an empathetic social chatbot[J]. Computational Linguistics, 2020, 46(1): 53-93.

[3] Wang W, Jiao W, Hao Y, et al. Understanding and Improving Sequence-to-Sequence Pretraining for Neural Machine Translation[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2022: 2591-2600.

[4] El-Kassas W S, Salama C R, Rafea A A, et al. Automatic text summarization: A comprehensive survey[J]. Expert systems with applications, 2021, 165: 113679.

[5] Brown P F, Cocke J, Della Pietra S A, et al. A statistical approach to machine translation[J]. Computational linguistics, 1990, 16(2): 79-85.

[6] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in neural information processing systems, 2014, 27.

[7] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[C]//Proceedings of the 3rd International Conference on Learning Representations, 2015.

[8] 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 (Volume 1: Long Papers). 2017: 1073-1083.

[9] Iqbal T, Qureshi S. The survey: Text generation models in deep learning[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(6): 2515-2528.

[10] Qiu X, Sun T, Xu Y, et al. Pre-trained models for natural language processing: A survey[J]. Science China Technological Sciences, 2020, 63(10): 1872-1897.

[11] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[J]. Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.

[12] Kenton J D M W C, Toutanova L K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//Proceedings of NAACL-HLT. 2019: 4171-4186.

[13] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.

[14] Lewis M, Liu Y, Goyal N, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 7871-7880.

[15] Lewis M, Liu Y, Goyal N, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 7871-7880.

[16] Reiter E, Dale R. Building applied natural language generation systems[J]. Natural Language Engineering, 1997, 3(1): 57-87.

[17] Reiter E. An architecture for data-to-text systems[C]//proceedings of the eleventh European workshop on natural language generation (ENLG 07). 2007: 97-104.

[18] 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 (Volume 1: Long Papers). 2013: 1406-1415.
[19] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
[20] Chowdhery A, Narang S, Devlin J, et al. Palm: Scaling language modeling with pathways[J]. arXiv preprint arXiv:2204.02311, 2022.
[21] Covington M A. Building natural language generation systems[J]. Language, 2001, 77(3): 611-612.
[22] Reiter E. An architecture for data-to-text systems[C]//proceedings of the eleventh European workshop on natural language generation (ENLG 07). 2007: 97-104.
[23] Stent A, Prasad R, Walker M. Trainable sentence planning for complex information presentations in spoken dialog systems[C]//Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). 2004: 79-86.
[24] Barzilay R, Lapata M. Collective content selection for concept-to-text generation[C]//Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. 2005: 331-338.
[25] Lapata M. Probabilistic text structuring: Experiments with sentence ordering[C]//Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. 2003: 545-552.
[26] Paiva D, Evans R. Empirically-based control of natural language generation[C]//Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). 2005: 58-65.
[27] Dušek O, Jurcicek F. Training a natural language generator from unaligned data[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015: 451-461.
[28] Dethlefs N, Hastie H, Cuayáhuitl H, et al. Conditional random fields for responsive surface realisation using global features[C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013: 1254-1263.
[29] 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. 2010: 502-512.

[30] Dušek O, Jurcicek F. Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016: 45-51.

[31] Ferreira T C, van der Lee C, Van Miltenburg E, et al. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures[C] //Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 552-562.

[32] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[J]. Advances in neural information processing systems, 2013, 26.

[33] Sutskever I, Martens J, Hinton G E. Generating text with recurrent neural networks[C]//Proceedings of the 28th international conference on machine learning (ICML-11). 2011: 1017-1024.

[34] Liu T, Wang K, Sha L, et al. Table-to-text generation by structure-aware seq2seq learning[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

[35] Gu J, Lu Z, 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 (Volume 1: Long Papers). 2016: 1631-1640.

[36] Distiawan B, Qi J, Zhang R, et al. GTR-LSTM: A triple encoder for sentence generation from RDF data[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1627-1637.

[37] Marcheggiani D, Perez-Beltrachini L. Deep Graph Convolutional Encoders for Structured Data to Text Generation[C]//Proceedings of the 11th International Conference on Natural Language Generation. 2018: 1-9.

[38] Koncel-Kedziorski R, Bekal D, Luan Y, et al. Text Generation from Knowledge Graphs with Graph Transformers[C]//2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics (ACL), 2019: 2284-2293.

[39] Sha L, Mou L, Liu T, et al. Order-planning neural text generation from structured data[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).

[40] Puduppully R, Dong L, Lapata M. Data-to-text generation with content selection and planning[C]//Proceedings of the AAAI conference on artificial intelligence. 2019,

33(01): 6908-6915.

[41] Moryossef A, Goldberg Y, Dagan I. Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 2267-2277.

[42] Zhao C, Walker M, Chaturvedi S. Bridging the structural gap between encoding and decoding for data-to-text generation[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 2481-2491.

[43] Chen K, Li F, Hu B, et al. Neural data-to-text generation with dynamic content planning[J]. Knowledge-Based Systems, 2021, 215: 106610.

[44] Puduppully R, Lapata M. Data-to-text generation with macro planning[J]. Transactions of the Association for Computational Linguistics, 2021, 9: 510-527.

[45] Song K, Tan X, Qin T, et al. MASS: Masked Sequence to Sequence Pre-training for Language Generation[C]//International Conference on Machine Learning. PMLR, 2019: 5926-5936.

[46] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485-5551.

[47] Lewis M, Liu Y, Goyal N, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 7871-7880.

[48] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. 2018.

[49] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

[50] Chen W, Su Y, Yan X, et al. KGPT: Knowledge-Grounded Pre-Training for Data-to-Text Generation[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020: 8635-8648.

[51] Kale M, Rastogi A. Text-to-Text Pre-Training for Data-to-Text Tasks[C]//Proceedings of the 13th International Conference on Natural Language Generation. 2020: 97-102.

[52] Li X L, Liang P. Prefix-Tuning: Optimizing Continuous Prompts for

Generation[C]//Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2021: 4582-4597.

[53] Dathathri S, Madotto A, Lan J, et al. Plug and Play Language Models: A Simple Approach to Controlled Text Generation[C]//International Conference on Learning Representations. 2020.

[54] Deng S, Yang J, Ye H, et al. LOGEN: few-shot logical knowledge-conditioned text generation with self-training[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2023.

[55] Chen Z, Eavani H, Chen W, et al. Few-Shot NLG with Pre-Trained Language Model[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 183-190.

[56] Chang E, Shen X, Zhu D, et al. Neural Data-to-Text Generation with LM-based Text Augmentation[C]//16th Conference of the European Chapter of the Association for Computational Linguistics. 2021.

[57] Kasner Z, Dušek O. Neural Pipeline for Zero-Shot Data-to-Text Generation[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2022: 3914-3932.

[58] Li J, Zhao W X, Wei Z, et al. Knowledge-based review generation by coherence enhanced text planning[C]//Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021: 183-192.

[59] 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. 2016: 1203-1213.

[60] Novikova J, Dušek O, Rieser V. The E2E Dataset: New Challenges For End-to-End Generation[C]//Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue. 2017: 201-206.

[61] Gardent C, Shimorina A, Narayan S, et al. The WebNLG challenge: Generating text from RDF data[C]//Proceedings of the 10th International Conference on Natural Language Generation. 2017: 124-133.

[62] Koncel-Kedziorski R, Bekal D, Luan Y, et al. Text Generation from Knowledge Graphs with Graph Transformers[C]//2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics (ACL), 2019: 2284-2293.

[63] Parikh A, Wang X, Gehrmann S, et al. ToTTo: A Controlled Table-To-Text Generation Dataset[C]//Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020: 1173-1186.

[64] Nan L, Radev D, Zhang R, et al. DART: Open-Domain Structured Data Record to Text Generation[C]//2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021. Association for Computational Linguistics (ACL), 2021: 432-447.

[65] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002: 311-318.

[66] Doddington G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics[C]//Proceedings of the second international conference on Human Language Technology Research. 2002: 138-145.

[67] Banerjee S, Lavie A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments[C]//Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. 2005: 65-72.

[68] Lin C Y. Rouge: A package for automatic evaluation of summaries[C]//Text summarization branches out. 2004: 74-81.

[69] Vedantam R, Lawrence Zitnick C, Parikh D. Cider: Consensus-based image description evaluation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 4566-4575.

[70] Chen Y, Hazarika D, Namazifar M, et al. Inducer-tuning: Connecting Prefix-tuning and Adapter-tuning[J]. arXiv preprint arXiv:2210.14469, 2022.