第1章 自然语言生成的研究背景

"What I cannot create, I do not understand."

——Richard Feynman(美国理论物理学家，1965 年诺贝尔物理学奖获得者)

1.1 自然语言生成的背景概述

语言与文字作为几千年来人类文明发展的自然产物，在人类交流中扮演了不可替代的角色。自然语言处理 (Natural Language Processing, NLP) 作为研究人类语言的分支领域，自人工智能诞生以来就受到广泛的关注。特别是近几年随着深度学习的兴起，自然语言处理在许多应用领域取得了长足的进步，如机器翻译、对话系统、自动摘要系统的性能都获得了很大提升。基于现代深度学习框架的自然语言处理模型，在性能上显著地超越了基于规则或传统机器学习的方法。

自然语言生成 (Natural Language Generation，NLG) 是自然语言处理中非常重要和基础的任务。从狭义上说，自然语言处理包括自然语言生成和自然语言理解。人类的自然语言交互可以分解为两个阶段：一是从大脑中的意义 (Meaning) 到语言的表达过程，即通常意义上的自然语言生成；二是从语言到意义的理解过程，即通常意义上的自然语言理解 (Natural Language Understanding，NLU)。因此，在人类的自然语言交互过程中，自然语言生成 (或表达) 与自然语言理解就是两个最重要的部分。在以自然语言为主要交互手段的现代人机交互系统 (如对话系统、语音助手等) 中，自然语言生成和自然语言理解也是其中最核心的功能组件。

自然语言生成一直是自然语言处理领域的重要研究分支，具有长远的历史。20 世纪 50 年代，自然语言生成作为机器翻译的子问题被首次提出。20 世纪 70 年代，自然语言生成开始为专家系统生成简单的解释，以及为数据库查询的返回结果编写自然语言的答案。20

世纪 80 年代早期，自然语言生成逐渐成为自然语言处理中一个独立的研究领域，研究者开始探索其独特的关注点和研究问题。20 世纪 80~90 年代，研究者提出统计语言模型，开始从概率统计视角刻画语言文字，开启了统计语言建模的新篇章。2003 年，Bengio 提出了前馈神经网络语言模型[1]，非常前瞻性地改变了传统语言模型的建模思路。2013 年，词向量[2] 的提出标志着基于神经网络的语言建模时代的开始。时至今日，基于神经网络的语言模型已经占据了自然语言生成方法的统治地位。

本书主要讲述现代自然语言生成的基本问题、算法、模型和框架，特别是基于神经网络和深度学习架构的模型与方法。本章将讨论自然语言生成的基本定义和研究范畴，介绍传统的模块化生成框架和现代端到端的生成框架，并讨论自然语言生成中的核心问题——可控性问题。

1.2 基本定义与研究范畴

自然语言生成的宽泛定义可以表述为：在特定的交互目标下，从给定输入信息 (输入信息可能为空) 生成人类可读的语言文本的自动化过程。自然语言生成随着任务设定的不同，输入多种多样，但输出一定是可读的自然语言文本。从输入的维度来说，自然语言生成系统的输入可以表述为四元组，即 $< \mathrm{CG}, \mathrm{UM}, \mathrm{KB}, \mathrm{CH} >$[3]。

- CG：语言生成任务的交互目标 (Communicative Goal)，即所生成的语言文本服务于何种通信、交互目的。常见的交互目标包括告知、说服、广告营销、推荐等。
- UM：用户模型 (User Model)，即所生成的语言文本的读者或受众。对同样的信息，不同性别、年龄、职业的用户喜欢阅读的语言表达方式和风格是不同的。这涵盖了个性化语言生成任务，如个性化的对话生成、个性化的广告语生成。
- KB：任务相关的领域知识库 (Knowledge Base)，如实体、关系、领域规则等信息。KB 提供了关于语言生成任务的背景知识。
- CH：上下文信息 (Context History)，即模型在生成当前文本时需要考虑的输入信息，可能为文本、数据、图像或视频等。

以上定义几乎涵盖了所有的自然语言生成任务和场景，当然随着设定的不同，这些信息可以部分提供或全部提供。在输入中引入交互目标和用户模型使得这一定义具有普适的覆盖面，尤其包括了现代自然语言生成中的风格化生成、个性化生成等任务。

从研究范畴上看，自然语言生成在语言处理的基本问题上提供了独特的视角。**首先**，从人机交互的角度看，什么样的语言表达和语言行为，以及如何实现它们，才能更好地实现交互目标以促进人和机器之间的信息交流。**其次**，在特定的交互目标下，合适的语言表达的构成要素是什么，句法、语义、语用层面的约束应该如何形式化，语言学和领域知识应该如何表示又如何利用，自然语言生成中语言选择[①]的关键因素又是什么。**最后**，输入的信息应该如何被转换为高层次的符号化概念和文字 (例如，如何用语言文本描述复杂信息，如何抽象和概括原始数据以形成有意义、可理解的描述内容)，在这个过程中需要什么样的模型表达领域和世界知识，以及如何合理利用所关联的推理。显然，这些问题是整个自然语言处理领域基础的研究问题，它们普遍存在于几乎所有的自然语言生成任务和场景中。

1.3　自然语言生成与自然语言理解

自然语言生成和自然语言理解作为自然语言处理的两大分支，它们的共同特点在于两者都是研究关于语言和语言使用的计算模型。因此，它们的许多基础语言理论 (如关于语法、句法、语义、语用、篇章的理论) 是共通的。从一定意义上说，两者可以看成互逆的过程：自然语言理解将语言文本转换为计算机能处理的内在语义表示，而自然语言生成将计算机的内在语义表示翻译为人类可读的语言文本。

然而，两者也存在本质的不同。自然语言理解重在**分析**，目标是理解输入文本的语义、意图。语言分析过程通常是从底向上的过程：从词形 (Morphology)、语法 (Syntax)、语用 (Pragmatics)、篇章 (Discourse)，到最后的语义 (Semantics) 解析，需要在多个假设中选择最可能的一个或多个作为最终输出，其本质问题是假设管理 (Hypothesis Management)[4]。例如，在文本分类、词性标注、语义角色标注、自动问答、阅读理解等语言理解任务中，核

① 指对同一个语义可以选择不同的词汇、句式等进行表达。

心任务都是从假设空间中选择一个或多个类别标记、答案、选项作为最终模型的输出。自然语言理解的主要困难在于歧义 (同一个字面形式有多种可能的分析结果) 和输入信息不足 (需要字面以外的信息辅助才能做出分析和预测)。

而自然语言生成重在**规划和建构**，其遵循相反的信息流向：从语义到文本，从内容到形式。首先，自然语言生成是自上而下地在各种语言学层次上的规划过程，即从上层的语义出发，先要确定篇章和语用结构，再确定概念到词的映射，最后确定词形和具体的表型 (Surface Form)。其次，自然语言生成是考虑各种约束条件的从语义到文本的建构过程，这些约束条件包括文本长度、语言风格等。规划和建构的本质问题是确定选择，即选择合适的信息、词汇、句式来表达给定的输入信息。例如，在文本摘要中，需要从输入文档中选择合适的信息进行摘录；在生成最后的摘要时，需要选择合适的词和表达方式以生成通顺、流利的文本。

1.4 传统的模块化生成框架

传统的自然语言生成系统一般采用模块化的设计框架。一般认为，模块化的自然语言生成系统包括如下功能模块[3]。

(1) 内容选择 (Content Determination)

内容选择决定哪些信息应该出现在生成的文本中，哪些不应该出现。这个选择过程依赖多种因素，包括交互目标、生成内容的目标受众、输入信息源本身的重要性排序、输出的限制 (如长度、类别) 等。典型地，自动文摘系统中内容选择的两个关键因素就是信息源的重要性排序和摘要长度的限制。

(2) 文本结构化 (Text Structuring)

文本结构化决定需要表达的信息的先后顺序和结构。通常可以采用树状层次化结构或者篇章结构[①]确定表达信息的顺序和结构。

① Discourse relation 是指句子之间或子句之间的篇章关系，广泛采用修辞结构理论 (Rhetorical Structure Theory) 所定义的关系。

(3) 句子聚合 (Sentence Aggregation)

句子聚合决定哪些信息单元应该表达在一个句子中，或者某个信息单元应该被单独表达在某个句子中，以确保后续生成的句子流畅性和可读性。例如，在天气预报中，可能会在一句话中同时表达温度和湿度的信息：“今天相比昨天，温度更高，湿度也更大些。” 如果用两句话分别表达温度和湿度信息，则显得冗余和啰唆。

(4) 词汇化 (Lexicalisation)

表达同样的意思有许多不同的表达方式，词汇化的核心任务是确定合适的词汇以表达选定的信息单元。很多情况下，从概念到词汇的映射过程并不是简单直接的，而需要处理语义相似词、近义词、上下位词等语言学的变种，这个过程受许多约束变量的制约，如上下文、生成文本的风格、所表达的情感、立场等。

(5) 指称表达生成 (Referring Expression Generation)

指称表达生成确定对实体的指称表达，即在文本中使用合适的名称 (原名、别名、代词、反身代词等) 对实体进行引用。实体引用还可能涉及实体属性的使用，以便在上下文中无歧义地指称实体，例如，“积木块中最大的红色立方体” 从颜色、大小和形状三个方面对物体进行指称。

(6) 语言实现 (Linguistic Realization)

当以上所有的信息都确定后，语言实现负责形成句法、词形都正确的文本。这涉及句子成分的排序，人称、数、时态的一致，辅助词、功能词的插入等。

模块化的生成框架在自然语言处理的早期发展阶段中占据了统治地位，但在现代自然语言生成的不断发展中逐渐被新的框架所取代。不过，这些功能模块的划分对现在的研究仍然具有重要的指导和借鉴意义。实际上，在一个自然语言生成系统中，这 6 个模块的边界是不容易划分清楚的。例如，内容选择与文本结构化就存在紧密联系，它们的核心是确定内容和结构；句子聚合、词汇化、指称表达生成一起确定内容表达的微观结构。因此，三阶段模块化自然语言生成框架如图 1.1 所示，共包括以下三个步骤。

(1) **内容规划 (Content Planning)**

内容规划从宏观层面决定内容和结构，即解决 “说什么”(What to say) 的问题。这个步

骤实际上包括传统框架中的内容选择和文本结构化两个模块。内容规划的结果通常用树状的层次化结构表示，叶子节点代表要生成的内容，树状结构用于组织内容在文本中的顺序。

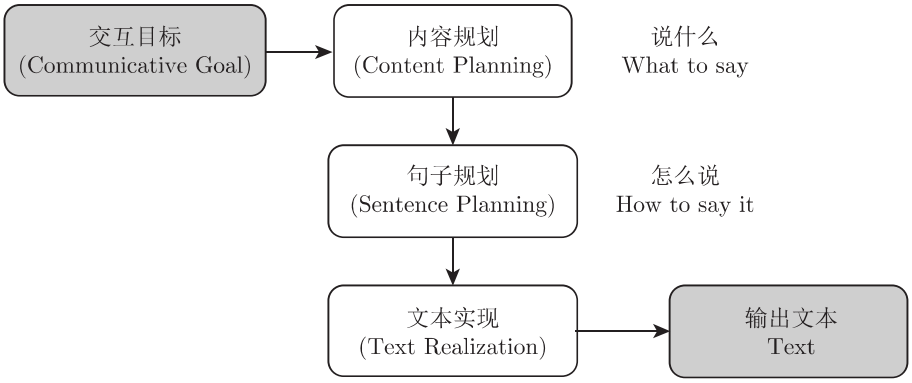

图 1.1　三阶段模块化自然语言生成框架

(2) **句子规划 (Sentence Planning)**

句子规划从微观层面决定词汇和句法结构，用以表达文档规划阶段确定的内容和结构，即解决“怎么说” (How to say it) 的问题。这个步骤包括句子聚合、词汇化和指称表达生成三个模块。同样地，句子规划的结果也可以用树状层次化结构表示，内部节点表示句子的结构，叶子节点表示单词或短语。

(3) **文本实现 (Text Realization)**

文本实现是指生成语法、句法、词形正确的文本内容，负责实现层面的语言表达。

传统的模块化自然语言生成框架一般采用两类方法实现：一是基于手写模板或语法规则的方法，二是基于统计的方法。当所处理的任务较简单、问题的规模较小时，基于手写模板或语法规则的方法不失为一个好的选择。模板一般表示为带有占位符的文本表述，如下所示：

“<Location> 附近有 <Cuisine> 类型的餐馆 <Restaurant>。”

其中，<Location>、<Cuisine>、<Restaurant> 表示相应的变量名。实例化时，只需将相应的值替换变量名，就可得到一个具体的文本，如“王府井 附近有烤鸭 类型的餐馆全聚德”。基于手写模板或语法规则的方法简单、实用，对输出文本完全可控，但缺点是死板、适用性有限，且很难扩展到语言表达形式丰富的生成场景中。

基于统计的方法一般可采取两种思路：其一，仍然基于手写的语法规则生成若干可能的候选文本，然后采用机器学习方法对这些候选文本进行排序，从中选出最优的结果；其二，直接使用统计信息影响生成过程的语言选择，不再使用先生成再过滤的策略。由于基于手写的语法规则不仅费时、费力，而且覆盖率有限，所以从大规模的树库资源中自动学习语法规则并利用这些语法规则生成文本成为了一个重要的研究方向。其中比较有代表性的是 OpenCCG 系统[5]，它基于组合范畴文法 (Combinatory Categorial Grammar，CCG)[6] 设计了一个覆盖度较广的英语表型实现器 (Surface Realizer)。该系统从宾州树库抽取 CCG 语法规则用于语言生成，并采用统计语言模型 (Statistical Language Models) 进行重排序。

基于手写模板或语法规则的方法考虑了许多细致的语言学规则和知识，而基于统计的方法更倾向于采用数据驱动的做法，自动学习数据隐含的规则和知识。增加统计信息的使用，往往也意味着语言学知识的减少。这种研究趋势也催生了完全数据驱动的现代语言生成框架——端到端的自然语言生成框架。

1.5　端到端的自然语言生成框架

传统的模块化自然语言生成框架属于“白盒”[①]的设计思路：每个模块的功能和职责相对明确，系统具有很好的可解释性，也方便故障诊断；但是，级联系统也会带来不可避免的错误传播问题，上一个模块的错误会传导至下一个模块，从而导致产生更严重的错误。现代深度学习的兴起，推动了各式各样基于神经网络的新自然语言生成模型，这些模型几乎都沿用了相同的端到端的自然语言生成框架，如图 1.2 所示。这是一种典型的“黑盒” (Black Box) 设计，传统的“内容规划—句子规划—文本实现”三个模块的功能被统一整合在一个解码器中。这种端到端的设计用一个模块就能实现所有模块的功能，不再纠缠每个模块的细节设计。采用数据驱动的方法训练模型，避免了手写模板或语法规则的麻烦。但是，这种设计存在不少问题：缺少对语言学知识的显式利用，缺少有效的手段来控制生成内容的质量，并且不能适用于数据资源不充足的情况。

① “白盒”即系统内部的模块和组件是可见的。

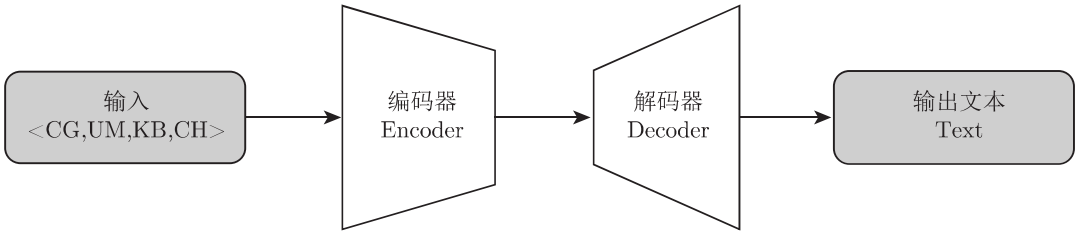

图 1.2　端到端的自然语言生成框架

绝大多数现代自然语言生成模型都采用了“编码器—解码器”框架，如图 1.2 所示。在这个框架中，输入 $X \triangleq < \mathrm{CG}, \mathrm{UM}, \mathrm{KB}, \mathrm{CH} >$ 经过编码器 (Encoder) 的处理被编码为向量表示，解码器 (Decoder) 则负责读取输入向量，生成所需要的文本。如前所述，这一框架将传统框架中的内容规划、句子规划、文本实现功能统一整合在解码器中。

可以从概率建模的角度对这一框架进行形式化。假设输入为 $X = < \mathrm{CG}, \mathrm{UM}, \mathrm{KB}, \mathrm{CH} >$，包含交互目标、用户模型、领域知识库、上下文信息；输出为 $Y = (y_1, y_2, \cdots, y_n)$，每个 $y_i\,(i = 1, 2, \cdots, n)$ 代表一个词。模型训练的目标是估计条件概率 $P_{\boldsymbol{\theta}}(Y|X)$，其中 $\boldsymbol{\theta}$ 表示模型的参数。在自回归 (Auto-Regressive) 生成模式[①]下, 条件概率 $P_{\boldsymbol{\theta}}(Y|X)$ 可以表达为

$$P_{\boldsymbol{\theta}}(Y|X) = \prod_{i=1}^{n} P_{\boldsymbol{\theta}}(y_i|Y_{<i}, X) \tag{1.1}$$

其中，$Y_{<i} = (y_1, y_2, \cdots, y_{i-1})$ 表示第 i 个位置已经生成的部分。模型的核心任务是估计概率分布 $P_{\boldsymbol{\theta}}(y|Y_{<i}, X)$，其中 $y \in \mathcal{V}$，$\mathcal{V}$ 表示词表，词表规模一般为 1000∼50000 量级。在自回归生成模式下，模型每次从概率分布 $P_{\boldsymbol{\theta}}(y|Y_{<i}, X)$ 采样从而生成一个单词 y_i，新得到的单词重新作为输入，再采样得到下一个单词 y_{i+1}。这一过程可用公式表达为

$$y_i \sim P_{\boldsymbol{\theta}}(y|y_1 y_2 \cdots y_{i-1}, X) \tag{1.2}$$

$$y_{i+1} \sim P_{\boldsymbol{\theta}}(y|y_1 y_2 \cdots y_{i-1} y_i, X) \tag{1.3}$$

以上采样形式有一个贪心解码 (Greedy Decoding) 的特殊情形：即每个时刻 i 选取概率最大的词作为生成结果，$y_i = \arg\max_{y \in \mathcal{V}} P_{\boldsymbol{\theta}}(y|y_1 y_2 \cdots y_{i-1}, X)$。但该方法的性能通常不如集束搜索 (Beam Search) 的策略，这部分将在第 3 章中详细阐述。

① 对应地，非自回归 (Non-Autoregressive) 生成模式一次性把所有的单词并行解码出来，将在第 7 章中详细介绍。

大多数的“编码器—解码器”框架广泛采用注意力机制 (Attention Mechanism) 以便获得更好的语言生成性能。其核心思想是，解码器在每个解码位置维护一个状态向量，并根据状态向量有选择性地利用编码器所得到的输入向量。一个典型的方法是，使用状态向量去匹配每个输入向量，得到所有输入向量上的权重分布。通过注意力机制，模型在不同解码位置对输入信息的利用是不同的。这使得解码器在不同的生成阶段有效地“注意”到不同的输入信息。已有的研究表明，注意力机制显著提升了文本生成的质量。这部分也将在第 3 章中详细阐述。

在“编码器—解码器”框架中，编码器和解码器可以采用各种神经网络模型，一般两者可以有不同的网络结构和参数。最常见的选择是循环神经网络 (Recurrent Neural Network, RNN)，将在第 3 章中详细阐述。另一种常见的选择是 Transformer[7] 的结构，这是一种采用多头注意力 (Multi-head Attention) 机制的神经网络模型，将在第 4 章中详细介绍。由于编码器只需要编码信息，所以可以选择基于预训练语言模型如 BERT[8] 等，以充分利用语境化的语言表示 (Contextualized Language Representation)。自 2019 年以来，由于预训练模型的兴起，出现了以 GPT[9] (General Pretraining) 为代表的仅有解码器的生成模型。它们使用了统一的神经网络结构同时处理编码和解码部分。或者说，GPT 模型中的编码器和解码器共享同样的网络结构与参数。同时，预训练的“编码器—解码器”结构也开始被研究和使用。

1.6　典型的自然语言生成任务

自然语言生成任务的输入多种多样，在之前的定义中给出了一个非常普适的输入形式，即 $X \triangleq < \mathrm{CG}, \mathrm{UM}, \mathrm{KB}, \mathrm{CH} >$。引入交互目标和用户模型，能使该形式覆盖风格化语言生成 (正式语言和非正式语言、金庸风格和莎士比亚风格等)、个性化语言生成 (根据用户特征的不同生成不同文本) 等任务。但输入中最重要的部分还是上下文信息即 CH 部分，该部分和生成文本直接相关。其形式通常可以是文本、类别标记、关键词、数据、表格、图像、视频等。本节将分别从输入信息的形态和信息转换两个角度来概括典型的自然语言生成

任务。

从输入信息形态的角度来说，自然语言生成任务可以分成：文本到文本 (Text-to-Text)、数据到文本 (Data-to-Text)、抽象意义表示到文本 (Meaning-to-Text)、多模态到文本 (Multimodality-to-Text)、无约束文本生成 (Zero-to-Text) 等。下面详细阐述这些任务。

- **文本到文本 (Text-to-Text)**。输入是文本内容 (连续文字或关键词信息)。这是最常见的一类任务，主要包括文本摘要、机器翻译、句子化简、语义复述生成、对话生成、诗歌生成、故事生成等。
- **数据到文本 (Data-to-Text)**。输入是数值、数据类信息，如表格、键值对列表、三元组等。例如，根据球赛的统计数据表格生成相应的体育新闻报道，根据结构化的个人信息生成维基百科简介页面等。在这类任务中，往往不可能将所有的原始数据都体现在生成内容中，因此对数据的选择、比较、关联、概括非常重要。
- **抽象意义表示到文本 (Meaning-to-Text)**。输入是语义的抽象表示，生成任务需要将抽象意义表示翻译成自然语言文本。常见的输入形式包括抽象意义表示 (Abstract Meaning Representation) 和逻辑表达式 (Logic Form)。
- **多模态到文本 (Multimodality-to-Text)**。输入是图像、视频等类型的多模态信息，模型需要将图像、视频中表达的语义信息转换为自然语言文本。典型的任务包括图像描述生成 (Image Captioning) 和视觉故事生成 (Visual Story Telling，根据视频或多个图像生成故事)。
- **无约束文本 (Zero-to-Text)**。不给定任何输入，要求模型自由生成自然语言文本。一般来说，这些模型会从学习到的分布中采样，以生成多样但符合数据分布的文本。部分模型也会先采样一个随机向量，然后将该向量转换为对应的文本。该任务一般用于测试基础的生成模型，如 RNN 语言模型、生成式对抗网络 (Generative Adversarial Network) 和变分自编码器 (Variational Auto-Encoder) 等。

从“输入—输出”信息变换的维度来看，自然语言生成可以分为开放端语言生成 (Open-ended Language Generation) 和非开放端语言生成 (Non-open-ended Language Generation)。开放端语言生成是指输入信息不完备、不足以引导模型得到完整输出语义的任务。

故事生成是一个典型的开放端语言生成任务：给定故事开头一句话或者几个主题关键字，模型需要生成具备一定情节的完整故事。显然，这个场景下输入信息非常有限，模型还需要利用其他信息 (如知识、大规模其他语料) 或“创造”输入中没有的其他关键信息，才能完成故事情节的规划并生成有意义的故事。这类任务普遍具有“一到多”的特点，即同一个输入存在多种语义显著不同的输出文本。对话生成、长文本生成 (如故事生成、散文生成) 都存在这样的特点，属于开放端语言生成任务。注意，这里的“创造”是相对狭义的，意指生成在输入中未指定或未约束的部分内容。

对应地，非开放端语言生成任务中，输入信息在语义上提供了完备甚至更多的信息，模型需要将这些信息用语言文字表述出来。机器翻译就是典型的非开放端语言生成任务：一般情况下，输入已经完整地定义了输出需要表达的语义，模型需要用另一种语言将其表达出来。语义复述生成可视为信息的等价变换，输入与输出的语义完全相同，只是表达形式不同。在文本摘要、句子化简这类任务中，输入给出了输出语义空间中更多的信息，模型需要通过信息过滤来选择合适的信息表达在输出文本中。在抽象意义表示到文本的任务中，输入完整地定义了输出所要表达的语义，因此模型只需要完成相应的语言实现 (Linguistic Realization) 即可。

表 1.1 所示为常见的自然语言生成任务与特点。

表 1.1　常见的自然语言生成任务与特点

任　务	任务类型	输入完备性	生成开放性	模型创造性
文本摘要	文本到文本	完备	非开放端	低
机器翻译	文本到文本	完备	非开放端	低
句子化简	文本到文本	完备	非开放端	低
语义复述生成	文本到文本	完备	非开放端	低
对话生成	文本到文本	非完备	开放端	高
故事生成	文本到文本	非完备	开放端	高
散文生成	文本到文本	非完备	开放端	高
表格—文本转换	数据到文本	完备	开放端	中
逻辑表达式—文本转换	抽象意义表示到文本	完备	非开放端	低
图像描述生成	多模态到文本	完备	非开放端	低
视觉故事生成	多模态到文本	非完备	开放端	中

1.7 自然语言生成的可控性

自然语言生成的可控性[①]是指模型在给定输入条件下生成不符合预期的文本，这些文本在语法、用词、语义等方面不符合人类语言的规范或者事先给定的约束。传统的模块化自然语言生成框架中，基于规则的方法往往能生成稳定可靠的文本。基于神经网络的端到端方法，由于引入了概率采样的机制，每次需要从模型估计的概率分布 $P_{\boldsymbol{\theta}}(y|Y_{<i}, X)$ 中采样 (参考公式(1.3))。考虑到词表规模较大，一般为 1000~50000 量级，因此概率分布中不可避免地存在大量出现概率很低的长尾词，再加上概率采样本身的随机性，基于采样的自然语言生成模型面临的可控性问题尤为严重。

如图 1.3 所示，自然语言生成的可控性问题可以从以下 4 个维度来概括。

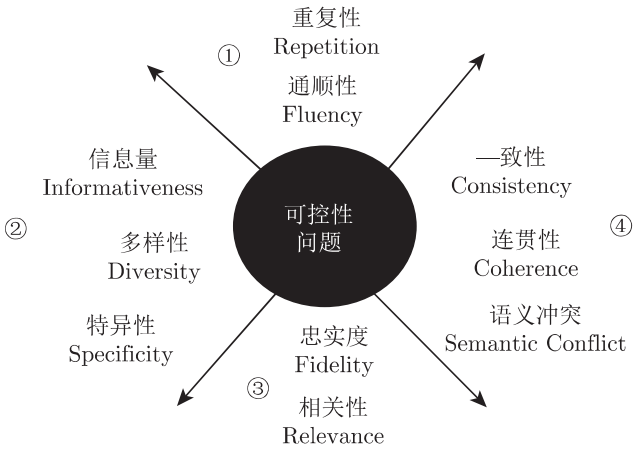

图 1.3 自然语言生成可控性的 4 个维度

1. 语法性问题

语法性问题是指生成文本是否通顺 (Fluency)，是否符合自然语言的语法，是否存在重复 (Repetition)。多数情况下，现代自然语言生成模型在大数据和大模型[②]的支撑下，通顺性几乎不存在显著问题。但在重复性问题上，即便现在最先进的预训练模型 GPT-2、GPT-3，

① 这里的可控性是广泛意义上的，狭义的可控性是指当调整输入变量时模型改变生成文本的属性的能力。

② 指模型的参数规模很大，如几千万到几亿甚至几十亿。

在生成长文本时仍然存在显著的重复性问题。有研究者发现，神经网络模型在语言生成时存在自我加强 (Self-reinforcing) 的问题，很容易生成重复内容。

2. 信息量问题

信息量问题是指模型生成高频的、无意义的、通用的内容，生成文本的信息量 (Informativeness)、多样性 (Diversity)、特异性 (Specificity) 显著不足。这是现代语言生成模型中最根本的可控性问题。基于概率采样的模型，更容易在每个位置上生成常见词，使整个生成的文本也变得常见 (如 “好的，我知道了”“我不知道” 等)，损失了其本身应该具有的信息量、多样性和特异性。

3. 关联度问题

关联度问题是指生成内容与输入的相关性 (Relevance) 低，或者与输入信息不符合，忠实度 (Fidelity) 低。例如，在对话生成中，生成与输入相关的、有意义的回复仍然是一个比较大的挑战。尤其是在数据到文本的生成任务中，忠实度是一个非常重要的问题，所生成的文本中不能编造新数据或者修改给定的输入数据。例如，在体育新闻报道中，若给定输入是 “A 队打败了 B 队”，但模型输出是 “B 队打败了 A 队”，这种情况则是不可接受的。

4. 语义问题

语义问题是指生成文本与给定上下文一致性 (Consistency) 不足，前后连贯性 (Coherence) 差，或与常识存在语义冲突 (Semantic Conflict)。如何检测语义问题本身就存在困难，因此语义问题也是可控性问题中最难的一类问题。当前自然语言生成模型很容易生成通顺但存在语义问题的内容，如前后矛盾 (如 “我喜欢你，但我不喜欢你”) 或者与常识不符 (如 “四个角的独角兽”)。

自然语言生成可控性的另一个维度是社会学偏置 (Social Bias)，即现有的生成模型容易生成侵略性的、恶毒的、人身攻击的、性别歧视的、种族歧视的不合适内容。尤其是当训练数据的质量不高、包含较多不合适的数据时，模型表现的社会学偏置问题可能会进一步加剧。现有模型的社会学偏置实际上反映了人类自身根深蒂固的偏见。对于这类可控性问题，一种简单的做法是在后处理基础上加过滤模块。从模型控制的角度来说，可以引入

反似然的训练 (Unlikelihood Training) 目标，即降低不合适词的采样概率。目前，这个研究方向的工作还比较少。

1.8 本书结构

本书共 12 章。第 1 章介绍了自然语言生成的背景、范畴、基本框架、任务设定和研究挑战。第 2 章介绍了从统计语言模型到神经网络语言建模的发展过程，重点介绍了语言建模的思想与技术演化过程。第 3~6 章分别介绍了基于循环神经网络、基于 Transformer、基于变分自编码器、基于生成式对抗网络的语言生成模型，它们几乎覆盖了现代语言生成模型的基本架构。这 4 章的内容也代表了 4 种对于语言建模完全不同的思路，在现代语言生成模型中极具代表性。第 7 章介绍了前沿的非自回归的语言生成，相比自回归的语言生成，提供了一种崭新的视角。

考虑到规划在语言生成尤其是长文本生成中的重要性，第 8 章以数据到文本生成、故事生成为例，介绍了基于规划的自然语言生成。第 9 章介绍了融合知识的自然语言生成，阐述了引入知识的动机、常用方法、面临的挑战，并以对话生成、故事生成为例，阐述了知识应用的方式和机制。第 10 章介绍了常见的自然语言生成任务和数据资源。第 11 章详细介绍了自然语言生成的评价方法，包括评价角度、人工评价、自动评价、自动评价与人工评价的结合、统计相关性等。第 12 章对本书的写作思路进行了总结和回顾，并对未来自然语言生成领域的发展趋势进行了展望。