3

演进之路：从语言模型到提示工程

本章将介绍语言模型和提示工程的基本概念和技术原理。语言模型是人工智能领域的一个重要研究方向，其从雏形初现到如今的新时代，经历了数十年的发展和演进。其中，Transformer 是目前最成功的语言模型之一，其结构和原理被广泛应用于各种自然语言处理任务。同时，本章还将详细介绍语言模型的训练方式，包括自回归训练和基于人工反馈的强化学习。此外，读者还将了解到提示工程，这是一种提高语言模型性能的重要技术手段，通过多样化、问题重述、提供背景知识、梯度提示、提供示例、角色扮演、实验与评估等技巧，可以使模型更加智能、灵活和可控制，从而促进应用与创新。

在这个过程中，笔者将尽可能用通俗易懂的语言来描述相关概念，避免使用过多的专业术语。但是，一些最基本的公式和形式化定义仍然是必要的。只有理解了这些内容，才能更好地理解 ChatGPT 的原理，从而创造出符合特定的生活和工作需求的 ChatGPT 使用方式。

3.1 什么是语言模型

在前面的内容中，我们经常提到 GPT 是一个语言模型。那么什么是语言模型呢？简单来说，语言模型（Language Model，LM）是自然语言处理（Natural Language Processing，NLP）中的一个基础概念，它使用各种统计和概率技术来确定一个给定的符号序列在人类的自然语言中出现的概率。对于一个符号序列$x_1, x_2, \cdots, x_n$，语言模型可以计算其联合概率：

$$P(x_1, x_2, \cdots, x_n)$$

这个概率分布反映了这个序列作为一个连续片段在语料库中出现的频率。这个概率越高，说明这个符号序列越符合自然语言的习惯；这个概率越低，说明这个符号序列越不自然或者不通顺。

为什么需要语言模型呢？在自然语言处理中，经常需要对一个给定的句子进行某种操作，例如翻译成另一种语言、判断其是否合法、生成下一个词等。为了完成这些操作，需要有一个方法来评估一个句子的好坏或者可能性。这就是语言模型的作用。

举个例子，假设要翻译下面这个英文句子。

> I love natural language processing.

如果有两个候选的中文翻译。

- 我爱自然语言处理。
- 自然我爱处理语言。

显然，第一个翻译更合理且流畅，第二个翻译很奇怪且不通顺。那么，怎么量化这种直觉呢？一种方法是使用语言模型来计算每个翻译在中文中出现的概率，并选择概率最高的那个作为最终结果。如果我们有一个好的中文语言模型，它应该能够给第一个翻译分配更高的概率，因为第一个翻译更符合中文的习惯和规则。

GPT 是一类特殊的语言模型，称作自回归语言模型（Auto-Regressive Language Model，ARLM）。自回归语言模型的特点是，假设一个文本序列中的每个词都依赖于前面的词。也就是说，给定一个序列$X = (x_1, x_2, \cdots, x_T)$，其中$x_t$表示第$t$个词，或者准确地说应该叫

作 Token，自回归语言模型的目标是计算该序列的联合概率分布$P(X)$，并且根据链式法则（Chain Rule），将其分解为

$$P(X) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\cdots P(x_T|x_1,x_2,\cdots,x_{T-1}) = \prod_{t=1}^{T} P(x_t|x_{<t})$$

其中$x_{<t}$表示序列中小于t的所有 Token。因此，自回归语言模型实际上是建立了一个条件概率分布$P(x_t|x_{<t})$模型，即在给定前缀的情况下，预测下一个 Token 出现的概率。一个语言的 Token 数量是有限的，所有的 Token 的集合称为词表。语言模型可以预测词表中每个 Token 在此给定前缀的情况下出现的概率，所有的 Token 出现的概率组合到一起就形成了一个概率分布。接下来，只需要通过某种策略——称为解码算法——从语言模型输出的概率分布中选择一个概率比较大的 Token 作为序列的下一个 Token。将这个新的 Token 加入前缀中，再用语言模型预测下一个位置的概率分布，再选择一个新的 Token。迭代此过程就可以生成一段流畅的自然语言文本。

前面提到了几个相似的概念：Token、符号、语义单元、字和词，很多时候，它们被交替使用。接下来，笔者必须给出明确的解释，不然读者会很糊涂。在自然语言处理领域，Token 通常指的是文本处理的最小单位。一般情况下，语言模型会对文本进行分词或分字处理，从而将文本转换为由一系列 Token 组成的序列。这样做的目的是让模型能够理解和处理输入的文本。如果进行的是分词处理，那么词就是一个 Token。传统的语言模型大多采用分词的方法处理文本，特别是英文的单词之间天然存在空格来分隔，非常适合分词的处理方式。但是在中文文本中，词与词之间通常没有显式的分隔符，所以中文分词是一个很重要的、传统的自然语言处理研究课题。

在自然语言处理中，由于合成词和命名实体（如人名、地名、组织结构名）等复杂的词汇形式的存在，无论是中文还是英文，词汇量都可以看作无穷多。而语言模型通常使用有限大小的词表进行训练和预测，这就会面临一个未登录词（Out-of-Vocabulary，OOV）的问题。未登录词指的是在语言模型的训练数据中未出现过的词。当语言模型遇到未登录词时，它无法对其进行处理，从而影响了模型的准确性和鲁棒性。为了应对未登录词的问题，一个常见的解决方案是将子词单元（Sub-Word Unit）作为基本单位进行语言的建模，从而把未登录词分解成几个子词单元的序列进行处理。而中文中有着天然的子词单元，那就是字。于是许多中文语言模型采用字为基础单元进行建模。而现实总是比理论来得复杂，

一种更实际的情况是，语言模型的词表中既包含了字，也包含了常见的词，甚至还会包含一些不是词，但是会经常一起出现的字的组合，例如“是一”“我是”，等等。所以，ChatGPT 的基本处理单元既不是字，也不是词，还是叫 Token 更严谨。

在中文中 Token 通常被翻译为“标记”、“记号”或“令牌”。仅从个人喜好出发，笔者非常不喜欢这样的翻译。这是因为它们并不能很好地体现出 Token 与语言之间的关系。在自然语言领域，有些人将 Token 翻译为“符号”“词条”“词元”“单词片段”“文本片段”等。这些翻译虽然能够体现其在语言学中的含义，但它们又大多具备更丰富的词义，因此无法提供准确的表达。因此，在必须要进行严谨表达的地方，笔者更倾向于使用英文术语“Token”。在自然语言处理领域，由于技术术语的特殊性和专业性，英文术语通常更准确，因此使用英文术语会更恰当。为了保证文本通顺和可读性，在确定不会有歧义的情况下也会用“字”、“词”或“符号”表达相同的含义。无论使用哪种翻译或术语，关键是要准确地表达 Token 在自然语言处理中所代表的特殊概念。

3.2 语言模型的发展历程

语言模型作为自然语言处理领域中的核心问题之一，经历了漫长而曲折的发展历程。本节将带领读者回顾语言模型的发展历程，从最早的马尔可夫、香农、乔姆斯基等先驱们的工作开始，梳理语言模型发展的脉络，一路追溯到 21 世纪的神经语言模型。通过回顾语言模型的发展历程，我们可以更好地理解语言模型的基本原理，以及它们背后的思想和方法。

3.2.1 20 世纪 50 年代之前：雏形初现

马尔可夫被认为是第一位研究语言模型的科学家，尽管当时“语言模型”一词尚不存在。1906 年，马尔可夫提出了马尔可夫链，这个模型中只有有限多个状态，状态之间以一定的概率进行转换。马尔可夫证明了，如果状态转移概率确定不变，那么访问这些状态的概率将收敛到一个可计算得到的数学期望值。为了给这个模型一个形象的例子，1913 年，马尔可夫使用普希金的诗体小说《叶甫盖尼·奥涅金》构建了一个马尔科夫模型。他去掉文本中的空格和标点符号，将小说的前 20 000 个俄语字母分为元音和辅音，从而得到小说中的元音和辅音序列。然后，他用纸和笔计算出元音和辅音之间的转移概率。马尔

可夫的这个研究塑造了世界上第一个语言模型。

1948 年，香农发表了一篇开创性的论文《通信的数学理论》，开辟了信息论这一研究领域。在这篇论文中，香农引入了熵和交叉熵的概念。熵表示一个概率分布的不确定性，交叉熵则表示一个概率分布相对于另一个概率分布的不确定性。熵是交叉熵的下限。对于任何一个语言，熵是一个常数值，可以通过统计该语言符号所负荷的信息量的平均值得到。如果一种语言模型比另一种语言模型更能准确地预测单词序列，那么它应该具有更低的交叉熵。因此，香农的工作为语言建模提供了一个评估工具。后世的所有语言模型的训练都是在优化交叉熵，从而使建模的结果与真实的自然语言更接近。

20 世纪 40 至 50 年代，有限自动机理论的出现催生了乔姆斯基语法结构，用于对语言进行符号化的表示。有限自动机起源于 20 世纪 50 年代，起初是 1936 年出现的图灵机的衍生物。图灵机被许多人认为是现代计算机科学的基础。1943 年还催生了 McCulloch-Pitts 神经元，这是一种对生物神经元的简化模型，是后世神经网络研究的基础。乔姆斯基在 1956 年首先将有限状态机作为表征语法的一种方式，并将有限状态语言定义为能够由有限状态语法生成的语言。之后，他又对此理论进行了扩展，提出了上下文无关文法。理论上，所有的自然语言都可以通过上下文无关文法进行建模，这一简洁而优美的模型奠定了形式语言理论在之后八十年中的主流地位，直到 21 世纪才被以深度神经网络为基础的神经语言模型所取代。

3.2.2　20 世纪的后五十年：由兴到衰

1956 年，John McCarthy、Marvin Minsky、Claude Shannon 等来自麻省理工学院、IBM、兰德公司和其他机构的计算机科学家在美国新罕布什尔州的达特茅斯学院进行了一次为期两个月的研讨会。在这次的研讨会上，“人工智能（Artificial Intelligence，AI）”这一术语被第一次正式地提出，从而开创了这一必将给世界带来巨大改变的学科。这些科学家们共同讨论了许多问题，包括人工智能的定义、如何用计算机模拟人类思维、如何让机器从数据中学习知识等。当时的自然语言处理研究主要采用符号主义（Symbolism）或逻辑主义（Logicism）的方法，即基于规则或逻辑来表示和推理知识。符号主义认为知识可以用符号系统表示，并通过符号操作实现推理；逻辑主义则认为知识可以用数理逻辑来表示，并通过定理证明来实现推理。

语言作为人类所独有的智慧产物，从一开始就是人工智能研究的重要方向之一。而机器翻译作为最具使用价值的人工智能任务，得到了最多的关注。韦弗（Warren Weaver）于 1961 年提出了机器翻译中的转换方法，即将源语言转换成中间表示，再转换成目标语言。这种方法后来被广泛应用于基于规则或基于知识的机器翻译系统中。

好景不长，1966 年，美国政府发布了《ALPAC 报告》，该报告对当时的机器翻译技术进行了长达两年的调查评估，得出了悲观结论：机器翻译进展缓慢，质量低劣，成本高昂，且看不到未来。该报告刺破了第一次人工智能泡沫，并导致美国政府大幅削减对人工智能项目的资助。当时有个非常著名的翻译例子，“The spirit is willing, but the flesh is weak”本意是心有余而力不足，但是当这句话被翻译成俄文，再翻译回英语时，则变成“The vodka is good, but the meat is rotten”，和原文的意思完全对不上。

之后，自然语言处理的研究仍在艰难地继续，特别是计算机运行速度的提升和内存存储量的提高使语言处理技术的若干细分领域得到了商业应用，例如，在语音识别、拼写和语法检查等领域，涌现出了一批成功的商业公司。但是在面对如机器翻译、自动问答和文本写作等复杂任务时，当时的模型始终无能为力，直到 2001 年神经语言模型的出现。

3.2.3 21 世纪：新时代

2003 年，Yoshua Bengio 和他的合著者提出了最早的神经语言模型，开创了语言建模的新时代[1]。Bengio 等人提出的神经语言模型对传统的基于 *n*-gram 的概率语言模型进行了改进，其核心是被称为词嵌入（Word Embedding）的用一个低维的向量表示单词或词组的方法，如图 3-1 所示。传统的词表示方法是独热向量，即通过词汇表大小的向量表示文本中的词，其中只有对应于该词的项是 1，其他所有项都是 0，所以是一个稀疏向量。词嵌入作为一种低维的稠密向量，可以比高维而又稀疏的独热向量更有效地表示一个词，具有非常良好的泛化能力、鲁棒性和可扩展性。神经语言模型是由神经网络通过自监督的方法迭代计算得到的，这大大减小了语言模型建模的计算量。

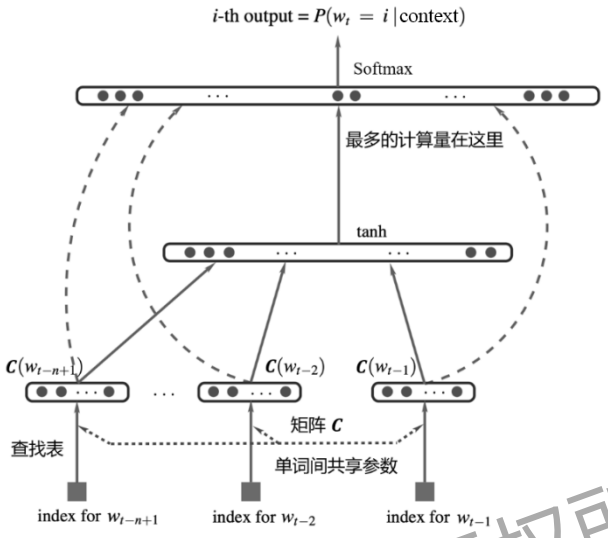

图 3-1

在 Bengio 等人的工作之后，词嵌入方法和神经语言建模方法经历了快速的发展。其中，图 3-2 所示的 Word2Vec 是最具代表性的词嵌入方法之一，它由谷歌的 Tomas Mikolov 等人于 2013 年开发[13]。Word2Vec 考虑了单词之间的上下文关系，利用神经网络进行训练。Word2Vec 模型可以分为两种：Skip-gram 模型和 CBOW 模型。Skip-gram 模型将目标单词作为输入，预测周围的上下文单词，而 CBOW 模型则相反，将上下文单词作为输入，预测目标单词。这两种模型都使用了浅层神经网络，包括一个嵌入层和一个 Softmax 层，用来计算每个单词在上下文中出现的概率。

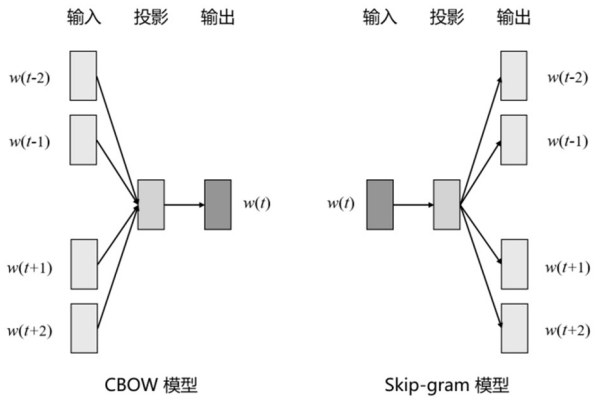

图 3-2

将词转换成低维稠密向量后，使得用一个神经网络计算一个连续的词序列（句子和文档）的语义成为可能。当时，最具代表性的神经网络模型是循环神经网络（Recurrent Neural Network，RNN），以及它的后续变种如长短时记忆网络（Long Short-Term Memory networks，LSTM）、门控循环单元（Gated Recurrent Unit，GRU）等[14]，如图 3-3 所示。RNN 在处理序列数据时，会保存一个内部状态，用来把前面的信息传递到后面。这种内部状态形成了一种记忆机制，可以帮助 RNN 处理序列数据中的上下文信息。与传统的词袋模型相比，RNN 可以更好地处理上下文信息，因此在语义建模和自然语言处理任务中表现得更好。LSTM 和 GRU 是 RNN 的变种，旨在解决 RNN 中的梯度消失和梯度爆炸问题。LSTM 通过引入门控机制来控制神经网络中信息的传递，从而实现记忆的保持，可以有效地解决长序列问题和长期依赖问题。GRU 通过减少门的数量来简化 LSTM，并在保持性能的同时减少了参数数量。

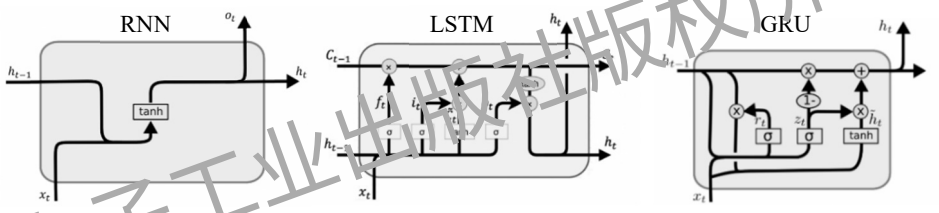

图 3-3

在这之后的一个重要改进是注意力机制（Attention Mechanism）[15]。在当时的神经机器翻译模型中，输入的整个序列会被 RNN 编码，形成一个固定长度的向量，再用另一个 RNN 解码，形成目标语言序列。这种模型叫作序列到序列（Sequence to Sequence，Seq2Seq）模型，也被称为编码器-解码器（Encoder-Decoder）架构。然而，这种做法存在一个严重的问题：由于中间向量的容量有限，在处理长序列或变长序列时，模型很难记住所有的信息，从而导致信息丢失和性能下降。注意力机制的核心思想是，给定一个查询向量和一组键值对，计算查询向量与每个键的相似度，然后将相似度转化为权重，并根据这些权重对每个值进行加权平均，得到一个加权和作为输出。在机器翻译任务中，输入序列被视为键，值和查询向量则是编码器的隐藏状态和解码器的上一个输出，这种方式可以实现将注意力集中在与输出相关的输入部分上。通过注意力机制和 Seq2Seq 模型的结合，神经机器翻译初步展现出惊人的能力，取得了显著的效果提升。但这还未挖掘出注意力机制的全部潜能，直到 2017 年 Transformer 模型的出现。