语言学研究

N 元语法模型及其与大语言模型的关系

冯志伟[1,2] 张灯柯[1]

(1. 新疆大学 中国语言文学学院,乌鲁木齐 830046;
2. 黑龙江大学 俄罗斯语言文学与文化研究中心,黑龙江 哈尔滨 150080)

摘 要:N 元语法模型是一种语言概率模型,核心是通过前 N-1 个单词预测第 N 个单词的出现概率,其理论基础为马尔可夫假设——一个单词的概率仅依赖于其前面的单词,无需追溯过远的历史信息。自然语言处理中常用二元、三元或四元语法,且 N 元语法的阶数越高,计算难度越大。目前流行的大语言模型是 N 元语法模型的进一步发展,而 N 元语法模型则是大语言模型的语言学理论基础。同时,N 元语法模型仍适用于一些自然语言处理场景。

关键词:N 元语法模型;马尔可夫假设;数据稀疏;自然语言处理;大语言模型

中图分类号:H0-06 **文献标志码**:A **文章编号**:1674-6414(2026)02-0038-19

0 引言

被誉为“神经网络之父”的辛顿(Geoffrey Hinton)是英国皇家学会院士(FRS)、加拿大皇家学会院士(FRSC)和加拿大国家勋章(CC)获得者,也是诺贝尔奖获得者和图灵奖获得者。在 2025 年 8 月 26 日世界人工智能大会(World AI Conference, WAIC)上,他发表了题为“数字智能是否会取代生物智能”(Will Digital Intelligence Replace Biological Intelligence)的开幕式主旨演讲。

辛顿在演讲中回顾了过去 60 年人工智能发展中的两条主流路径:一条是以推理为核心的逻辑主义,另一条是以模拟人类认知为基础的连接主义。他认为,语言理解不是符号演绎,而是从大规模的语言数据中提取出概念之间的关联。辛顿在演讲中回忆他自己在 1985 年开发的一个小型的语言模型。他认为如今的大语言模型本质上是“它的后代”。尽

收稿日期:2025-12-11

基金项目:中国历史研究院“绝学”学科扶持计划项目“回鹘文献学”(2924JXZ006)、自治区高校基本科研业务费科研项目“维吾尔语离线自动语音识别系统研究”(XJEDU2024J021)的阶段性研究成果

作者简介:冯志伟,男,新疆大学新疆民汉语文翻译研究中心教授,博士生导师,黑龙江大学俄罗斯语言文学与文化研究中心教授,主要从事计算语言学研究。

张灯柯,男,新疆大学中国语言文学学院副教授,硕士生导师,主要从事计算语言学、少数民族语言信息化研究。

引用格式:冯志伟,张灯柯. N 元语法模型及其与大语言模型的关系[J]. 外国语文,2026(2):38-56.

管现在的模型拥有更深的网络结构和更庞大的参数规模,但核心机制并未改变。因此,他认为“大语言模型和人类理解语言的方式相同”。他说,“1985 年,我做了一个小型模型,尝试结合这两种理论,以此理解人们对词语的理解方式。我给每个词设置了多个不同特征,记录前一个词的特征后,就能预测下一个词是什么。在这个过程中,我没有存储任何句子,而是生成句子并预测下一个词。其中的相关性知识,取决于不同词的语义特征之间的互动方式 ”(Hinton,2025)。

辛顿在演讲中提到“我给每个词设置了多个不同特征,记录前一个词的特征后,就能预测下一个词是什么”。他在这里所说的“生成句子并预测下一个词”的方法就是 N 元语法模型(N-gram model),俗称“接龙”。这是一种非常重要的方法。在本文中,我们将讨论 N 元语法模型及其与大语言模型的关系。

1　语言的概率模型

自然语言的统计处理要依赖于语料库(corpus),即联机的文本或口语的集合体。在使用统计方法的时候,为了计算单词的概率,就需要统计在训练语料库中单词的数目(Feng,2023)。

例如美国布朗大学的布朗语料库(Brown corpus),这是一个规模为 100 万单词的语料库,样本来自 500 篇书面文本,包括新闻、小说、技能与爱好、通俗知识、官方文件、学术著作等不同的类别,这个语料库是由布朗大学在 1963—1964 年开发而成,是最早的英语文本语料库。

还有一个类型是口语语料库。与文本语料库不同,口语语料库通常没有标点符号。20 世纪 90 年代初的美国 Switchboard 语料库是一个关于陌生人之间电话会话的口语语料库,包含 2 430 个会话,每个会话平均 6 分钟,总共有 240 小时的会话,包含 300 万单词。

语料库可以用来估计语言中词汇的总数。例如,我们用“类符”(type)来表示语料库中不同单词的数目,也就是词典容量的大小;用“形符”(token)来表示使用中的单词数目。Brown 语料库中有 100 万个“形符”,包含 61 805 个“类符”。口语中的词汇不如书面语丰富,如 Switchboard 语料库有 240 万个“形符”和大约 2 万个“类符”。1992 年,Kucera 对莎士比亚(Shakespeare)的全部著作进行过统计,发现其中有 884 647 个“形符”,29 066 个“类符”。

基于这样包含不同类型符号统计的语料库,我们就可以提出一个语言的概率模型(probabilistic model of language)来计算整个句子的概率,或预测下一个单词的概率(Jelinek, 1997)。

最简单的单词序列的概率模型是单纯地假定语言中的任何一个单词后面可以跟随着

该语言中的任何一个单词,在数学普遍假设下,任何一个单词后面可能跟随的该语言中的任意的其他单词的概率都是相等的。例如,如果英语中有 100 000 个单词,那么,任何一个单词后面跟随其他任何单词的概率将是 1/100 000 或 0.000 01。显而易见,这是一种过分简化的假设。

在稍微复杂一些的单词序列模型中,任何一个单词后面可以跟随着其他的任何单词,不过后面一个单词要按照它在语料中正常的频度(frequency)来出现。例如,在英语语料库中,单词 the 的频度相对地比较高,在 1 000 000 个单词的布朗语料库中,它出现 69 971 次(也就是说,在这个特定的语料库中,有 6.997 1%的单词是 the)。相比之下,单词 rabbit 在布朗语料库中的出现频度较低,只出现 11 次(也就是说,在这个特定的语料库中,大约有 0.001 1%的单词是 rabbit)。

我们可以根据这样的相对频度对下面将要出现的单词指派一个概率分布的估值。这样,如果我们看到了任何的符号串,我们就可以指派概率 0.069 971 给 the,指派概率 0.000 011 给 rabbit,从而来猜测下面可能出现的单词。

但是,仅仅根据相对频度还不够,我们还需要考虑上下文情况。例如,在如下符号串中:

Just then, the white

跟随着单词 white 之后,rabbit 似乎是一个比 the 更合理的单词。

这说明,我们不能简单地看单词的单独的相对频度,而是要看单词对于给定的前面一个单词的条件概率(conditional probability)。也就是说,我们要看 white 后面 rabbit 出现的概率。我们把这个条件概率表示为:

P (rabbit | white)

在语言的概率模型中,这样的条件概率是非常重要的,我们可以根据前面的单词来预测后面单词出现的可能性。

当前面给定的单词序列很长的时候,我们不知道用什么简单的办法来计算条件概率是多少。例如,在句子“Just the other day I saw a rabbit”中,如果我们要计算在上下文“Just the other day I saw a”之后出现 rabbit 的条件概率,也就是计算条件概率:

P(rabbit | Just the other day I saw a)

显然这是很困难的问题。我们不能在一个很长的符号串之后,来数每一个单词的出现次数;因为这时我们需要一个无比庞大的语料库。为此我们有必要对于上面所述的语言的概率模型加以改进和简化。

2 N 元语法模型

我们可以使用一个简化的办法来改进语言的概率模型—逼近方法:仅通过前一个单词

的概率来“逼近”计算当前单词的出现概率,也就是“二元语法模型”(bigram model)。

换言之,在二元语法模型中,我们不是计算条件概率:

P(rabbit | Just the other day I saw a)

而是使用如下的条件概率来逼近这个概率:

P(rabbit | a)

一个单词的概率只依赖于它前面单词的概率的这种假设叫做马尔可夫假设(Markov assumption)。根据马尔可夫假设,我们没有必要查看一个单词很远的“过去”,就可以预见到这一个单词将来的概率。这样的模型叫做马尔可夫模型(Markov Model)。

基本的二元语法模型可以看成是每个单词只有一个状态的马尔可夫链(Markov chain)。我们可以把二元语法模型(只看前面的一个单词)推广到三元语法模型(看前面的两个单词),再推广到 N 元语法模型(看前面的 N-1 个单词)。

对于二元语法来说,我们使用如下公式,就可以计算出整个符号串的概率:

$$P(w_1^n) \approx \prod_{k=1}^{n} p(w_k \mid w_{k-1})$$

在这个公式中 w 表示词(word 首字母),w_k 表示第 k 个词,w_{k-1} 表示 w_k 前面的第 k-1 个词。

下面我们使用只包含三个句子(共有 14 个单词)的微型语料库(mini corpus)来说明二元语法。

I am Sam

Sam I am

I do not like green eggs and ham

我们需要对这三个句子做进一步的加工,在句子的开头标以一个特殊的句首符号<s>,以便使句子中的第一单词具有二元的上下文。我们还需要在句子的末尾标以一个特殊的句末符号</s>。

这样我们可以得到如下所示的微型语料库(见图 1):

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

图 1　微型语料库

根据这个微型语料库,“<s>”出现 3 次,而“<s> I”组合出现 2 次,故得:

P(I | <s>) = 2/3 ≈ 0.67

仿此可以得到如下的二元语法概率计算的一些结果:

P(Sam | <s>) = 1/3 ≈ 0.33

P(am | I) = 2/3 ≈ 0.67

P(</s> | Sam) = 1/2 = 0.50

P(Sam | am) = 1/2 = 0.5

P(do | I) = 1/3 ≈ 0.33

这样一来,在上述的微型语料库中,句子 I am Sam 的二元语法概率为:

P(<s>I am Sam</s>)

=P(I | <s>) × P(am | I) × P(Sam | am) × P(</s> | Sam)

= 0.67 × 0.67 × 0.5 × 0.5

=0.112 225

为了获得更加清楚的印象,让我们来看语音理解系统中一个稍微复杂的例子。

美国斯坦福大学朱夫斯凯(Jurafsky)等在 1994 年研制成功一个基于语音的饭店咨询系统,叫做“Berkeley 饭店规划”(Berkeley Restaurant Project)。用户可以通过这个系统询问关于在加利福尼亚州(California State)的伯克利市(Berkeley)的饭店的问题,系统从地方饭店的数据库中检索合适的信息显示给用户(朱夫斯凯 等,2018)。

这里是用户提问的一些样本:

(1)I'm looking for Cantonese food.

我在找广东菜的饭店。

(2)I'd like to eat dinner someplace nearby.

我喜欢在附近的地方吃晚餐。

(3)Tell me about Chez Panisse.

请告诉我关于 Chen Panisse 饭店的情况。

(4)Can you give me a listing of the kinds of food that are available?

你可以给我已经准备好的各种食品的一个单子吗?

(5)I'm looking for a good place to eat breakfast.

我正在找一个适合吃早饭的地方。

(6)I definitely do not want to have cheap Chinese food.

我确实不想吃便宜的中国食品。

(7)When is Caffe Venezia open during the day?

近来 Venezia 咖啡店什么时候开门?

(8)I don't wanna walk more than ten minutes.

走十分钟以上的地方我不想去。

下文中的表1是“Berkeley饭店规划”中关于二元语法概率的一个样本,它说明了在单词eat之后可能出现的某些单词的概率,这些概率是从用户所说的句子中统计得出的。注意,这些概率编码说明了两个方面的事实:在本质上很严格的句法事实(例如,在eat之后常常会是一个名词短语的开头,如形容词、修饰词、名词等)和某些与文化有关的事实(例如,在美国询问如何找英国食品的概率是很低的)。

表1　“Berkeley饭店规划”中说明eat后最容易出现的单词的二元语法

eat on	.16	eat Thai	.03
eat some	.06	eat breakfast	.03
eat lunch	.06	eat in	.02
eat dinner	.05	eatChinese	.02
eat at	.04	eat Mexican	.02
eat a	.04	eat tomorrow	.01
eat Indian	.04	eat dessert	.007
eat today	.03	eat British	.001

除了表1中的概率之外,我们的语法还包括如表2所示的二元语法概率(<s>是一个特殊的单词,表示句子的开始)。

表2　“Berkeley饭店规划”中关于二元语法的更多片断

<s>I	.25	I want	.32	want to	.65	to eat	.26	British food	.60
<s>I'd	.06	I would	.29	want a	.05	to have	.14	British restaurant	.15
<s>Tell	.04	I don't	.08	want some	.04	to spend	.09	British cuisine	.01
<s>I'm	.02	I have	.04	wantthai	.01	to be	.02	British lunch	.01

有了这些数据,我们就可以计算“I want to eat British food”这个句子的概率了,计算时,我们只要把相邻两个单词的二元语法概率相乘在一起就行了。如下所示:

$$P(\text{I want to eat British food})$$

$$= P(\text{I}|\text{<s>}) \times P(\text{want}|\text{I}) \times P(\text{to}|\text{want}) \times P(\text{eat}|\text{to}) \times P(\text{British}|\text{eat}) \times P(\text{food}|\text{British})$$

$$= 0.25 \times 0.32 \times 0.65 \times 0.26 \times 0.001 \times 0.60$$

$$= 0.000\ 008\ 11$$

三元语法模型的计算原理与二元语法模型相同,不过这时我们用前面两个单词作为条件(例如,我们用P(food | eat British)来替代P(food | British)。为了计算在每个句子开头的三元语法概率,我们可以使用两个假想的单词(pseudo-word)作为三元语法的条件(也就是, P(I | <start1>, <start2>)),其中,start1和start2是位于句子开头的假想的单词。

N 元语法模型可以使用训练语料库和归一化(normalizing)[①]的方法而得到。

表 3 是从“Berkeley 饭店规划”中得到的一个二元语法的某些二元语法计数。注意,大多数的计数为零。实际上,我们选择这 7 个单词(I、want、to、eat、Chinese、food、lunch)样本时已经设法尽量使它们彼此接应得比较好;如果随机地选择 7 个单词,数据将会更加稀疏。

表 3 二元语法计数

	I	want	to	eat	Chinese	food	lunch
I	8	1 087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

这 7 个单词的一元语法计数如下:

I	3 437
want	1 215
to	3 256
eat	938
Chinese	213
food	1 506
lunch	459

用每个单词相应的一元语法计数来除它们各自的二元语法计数,进行归一化处理后,我们可以得到经过归一化之后的二元语法概率(见表 4)。

表 4 经过归一化之后的二元语法概率

	I	want	to	eat	Chinese	food	lunch
I	.002 3	.32	0	.003 8	0	0	0
want	.002 5	0	.65	0	.004 9	.006 6	.004 9
to	.000 92	0	.003 1	.26	.000 92	0	.003 7
eat	0	0	.002 1	0	.020	.002 1	.055
Chinese	.009 4	0	0	0	0	.56	.004 7
food	.013	0	.011	0	0	0	0
lunch	.008 7	0	0	0	0	.002 2	0

① 对于概率模型来说,所谓归一化,就是用某个总数来除,使得最后得到的概率的值处于 0 和 1 之间,以保持概率的合法性。

现在我们来研究不同的 N 元语法模型的一些例子,以便从直觉上了解这种模型的两个重要事实。

第一个事实是:当我们增加 N 的值的时候,N 元语法模型的精确度也相应地增加。

第二个事实是:N 元语法性能强烈地依赖于训练它们的语料库,特别是依赖于语料库的种类和单词的容量。

为了对于这样的事实有一个直观的了解,我们在《莎士比亚全集》的语料库上分别训练一元语法、二元语法、三元语法和四元语法模型,生成随机的例(9~12)四个相应例句。在下面的例子中,我们把标点符号也看成一个单词,而且,当用语料库来训练语法的时候,把所有的大写字母都改写成小写字母。在生成句子之后,为了便于阅读,我们再把有关的小写字母恢复成大写字母。

(9)a. To him swallowed confess hear both. Which. Of save on trail for are ay device and rote have

b. Every enter now severally so, let

(10)a. What means, sir. I confess she? then all sorts, he is trim, captain.

b. Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live. king. Follow.

(11) a. Sweet prince, Falstaff shall die. Harry of Monmouth's grave.

b. This shall forbid it should be branded. if renown made it empty.

(12)a. Will you not tell me who I am?

b. It cannot be but so.

我们训练模型的上下文越长,所生成句子的连贯性就越好。在一元语法生成的句子中,单词与单词之间没有连贯关系,而且没有一个句子是以句号或其他可以做句末标点的符号结尾的。在二元语法生成的句子中,单词与单词之间只存在着非常局部的连贯关系。三元语法和四元语法生成的句子,看起来已经比较像莎士比亚的句子了。仔细地查看一下四元语法生成的句子,它们更像莎士比亚著作中的句子。例如四元语法中的"It cannot be but so"这个句子,与莎士比亚著作《国王约翰》(King John)中的句子完全一样。

尽管莎士比亚的著作有很多不同版本,但是其总词数不会多于 100 万单词。前面我们说过,Kucera 曾经计算过莎士比亚全集的词数,形符数为 884 647 个,类符数为 29 066 个(包括专有名词)。这意味着,即使是二元语法模型,其数据也是非常稀疏的;从 29 066 个不同的单词("类符")可以形成 $29\,066^2$ 个(或者 8 亿 4 千 4 百万个)以上的二元语法关系,在这种情况下,用 100 万单词的训练集来估计那些不常见单词的频度,显然是非常不充分的。实际上在莎士比亚著作中不同的二元关系类型不会超过 30 万个。莎士比亚著作的规

模如果用来训练五元语法,那就显得相当不够了;因此,我们的生成系统对于前面头 4 个词(It cannot be but)的五元语法,下面可能接续的单词只有 5 个(that、I、he、thou 和 so);很多包含四个单词的五元语法,它们后面的接续单词都只有 1 个。

为研究语法对训练集的依赖关系,我们继续采用与莎士比亚全集差异显著的《华尔街杂志》(WSJ)语料库训练 N 元语法,二者均为英语子集,直觉上一般会认为二者 N 元语法会相互重叠覆盖。为验证该直觉,我们根据基于《华尔街杂志》4 000 万单词的语料库分别训练一元、二元、三元语法并生成三个长句,为避免数据稀疏问题可能产生的零概率(zero probability),所有语法均采用平滑法(smoothing)处理以避免零概率(冯志伟,2017)。同时,为了便于阅读,我们用手工把英语的专有名词的首字母都改为大写字母。

(13) Months the my and issue of year foreign new exchange's September were recession exchange new endorsed a acquire to six executives(一元语法)

(14) Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planner one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her(二元语法)

(15) They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions(三元语法)

把这些句子同前面的那些根据《莎士比亚全集》语料库生成的句子相比较,二者之间显然没有重叠覆盖的现象,这种差异告诉我们,生成英语句子实际需要一个规模很大并包含不同文体、领域的语料库。尽管这样,像 N 元语法这样简单的统计模型也没有能力去模拟不同种类的不同风格。当我们阅读莎士比亚著作的时候,我们只想看见莎士比亚的句子,思想不会跳到《华尔街杂志》的那些文章上去。由此可见,用 N 元语法模型生成的句子对于语料有很大的依赖性。

N 元语法统计模型中的概率来自训练它的语料库。这个训练语料库(training corpus)需要精心地设计。如果训练语料库太偏向于某个任务或某个领域,那么,概率就可能太窄,对于新的句子缺乏一般性。相反,如果训练语料库太泛,概率就不能充分地反映有关的任务或领域的特点。

因此,当我们对于具有相关数据的某个给定的语料库应用语言统计模型的时候,一开始我们就要把数据分为“训练集”(training set)和“测试集”(test set)两部分。我们在训练集上训练统计参数,然后使用这些参数去计算测试集中的概率。

N 元语法为语言学研究提供了一种可量化、可计算、可验证的建模方式,从而弥补了传

统语言学在大规模语料分析和概率建模方面的不足。其具体价值如下：

①揭示语言使用的统计规律：N 元语法基于马尔可夫假设，通过统计词语的共现频率，就能够捕捉到语言中局部依赖关系的概率特征，帮助我们发现“某个词后面最可能接哪些词”，从而揭示语言使用的偏好性和习惯性搭配，为语言学研究从“规则驱动”（rule-driven）转向“数据驱动”（data-driven）提供了基础，尤其在语料库语言学中具有重要作用。

②为语言建模提供可计算框架：N 元语法是最早被广泛使用的语言模型之一，在语言学研究中，它可以作为语言结构的基准模型来评估更加复杂的语言结构模型的效果。例如，在句法分析或语义分析研究中，我们可以用 N 元语法模型作为对照，验证在引入句法特征或语义特征之后，是否能够显著地提升语言模型的分析能力。

③辅助语言结构与变异研究：通过分析不同语言、不同文体或历史时期的 N 元语法分布，我们可以比较语言变体的差异。其一，比较语言变体的共时差异，如比较文言文与白话文、方言与标准语的差异；其二，可以追踪语言演变的过程，诸如某种语言搭配现象的历史兴衰；其三，可以识别不同文体特征，从而揭示小说、历史文本、戏剧文本等不同文体的语言差异。

④为语言理论提供实证基础：传统语言学强调语言的结构性与规则性，而 N 元语法则强调语言的经验性与概率性。对比两者，可以促使研究者反思语言的一些基本理论问题。例如，语言知识是先天的规则系统，还是后天统计学习的结果？语法结构是否可以通过大规模语料归纳得出？进行这样的反思有助于推动基于经验的统计语言学与基于规则的形式语言学的对话，促使更复杂的语言模型的研制。

⑤支持低资源语言与古语言研究：对于缺乏完整语法描述的语言（如古汉语、濒危语言），N 元语法提供了一种无需深层语法知识即可进行初步分析的工具。在古汉语中，由于“字”与“词”界限模糊，N 元语法可作为字符级别的替代方案，辅助完成分词、搭配提取等任务。研究表明，基于 N 元语法的统计模型（如字符级的三元语法）在判断句子边界上具有良好效果，尤其在《论语》《孟子》等经典文本中，召回率可达 81%（陈天莹 等，2006）。

N 元语法虽简单，却为语言学研究提供了实证基础、建模工具和理论反思，是语言学家理解语言结构性与统计性双重本质的重要窗口。

3　N 元语法模型在自然语言处理中的应用

上面我们论述了 N 元语法模型在语言学研究中的应用。这里，我们进一步说明 N 元语法模型在自然语言处理（Natural Language processing，NLP）中的应用。

在统计机器翻译（statistical machine translation ）中，N 元语法模型是至关重要的。在汉英统计机器翻译中，如果要把源语言中文的句子 “他向记者介绍了声明的主要内容”翻译

成英文,使用统计机器翻译,我们得到了如下可能的英语粗译文:

(16) a. He briefed to reporters on the chief contents of the statement

b. He briefed reporters on the chief contents of the statement

c. He briefed to reporters on the main contents of the statement

d. He briefed reporters on the main contents of the statement

我们可以根据 N 元语法模型做出判断:Briefed reporters 比 briefed to reporters 的概率更高,main contents 比 chief contents 的概率更高,因此,(16d) 概率最高,最可能是通顺正确的译文。

在拼写更正(spelling correction)中,我们需要发现并且改正的拼写错误有的是由于真实的英语单词偶然造成的。例如,

(17) a. They are leaving in about fifteen minuets to go to her house.

b. The design an construction of the system will take more than a year.

由于这些错误的单词 minuets 和 an 都是实际存在的真词(real words),如果仅仅根据是否在词典中出现来判断错误,那么,我们就不能发现这样的错误。然而,如果我们注意到 in about fifteen minuets 这个序列与 in about fifteen minutes 这个序列比较起来,前者出现的可能性小得多(minuets 的意思是"小步舞",与"in the fifteen"的结合概率很低),我们就可以发现这样的错误。拼写检查程序可以使用概率预测器来检查这样的错误,并且选择概率比较高的符号序列作为正确的答案。

除上面举例介绍的领域外,N 元语法模型在词类标注(part-of- speech tagging)、单词相似度计算(word similarity computation)等自然语言处理研究中以及在匿名作者辨认(authorship identification)、情感抽取(sentiment extraction)和手机的预测式文本输入(predictive text input)等应用系统中都起着至关重要的作用。

当前流行的一些语言模型背后原理都是 N 元语法。例如,如果我们要在英语字符串"The best thing about AI is its ability to"后继续生成英语,在这个字符串后可能出现的单词如下(见图 2):

图 2 直观地展示了"预测下一个词"的核心任务。我们可以将该图理解为一个坐标系,坐标系中的每一个点代表一个动词,其中横轴(下方,0~100)代表一个词在训练数据中出现的词频排名,排名越靠左(如 learn、predict),意味着这个词在语料库中越常见。纵轴(左侧,0.002~0.050)代表模型在预测下一个词时,分配给该词的概率,位置越高,表示模型在当下语境中认为这个词出现的可能性越大。

这就是"接龙"方法,也就是辛顿在世界人工智能大会(WAIC)演讲中介绍的"预测下

一个词是什么”的方法。N 元语法为这样的“接龙”方法提供了语言学理论的支持。

N 元语法模型的能力随着模型的元数的增高而增高。语言数据的规模越大,自然语言处理的效果越好。当然,N 元语法的阶数越高,计算的难度也越大,谷歌公司在 2007 年曾经研制过七元语法,也就是考虑当前词前面六个单词对于当前词的影响,计算难度已经很大了。

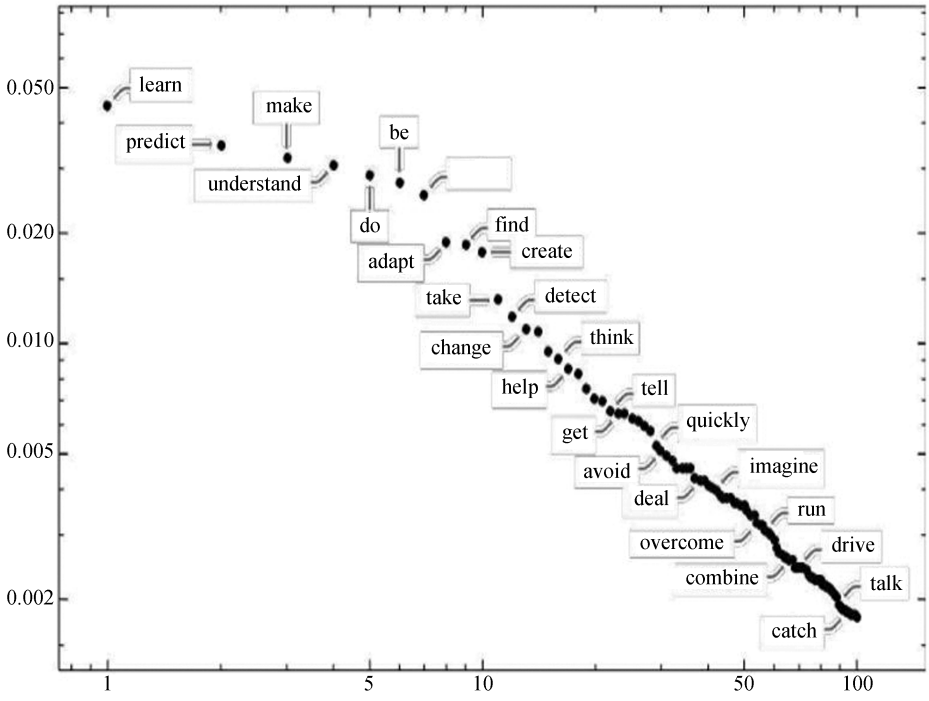

图 2　接龙

自然语言极为复杂,可能性无穷,即便训练语料规模庞大,也难以覆盖所有 N 元语法,易引发数据稀疏(data sparseness)问题。因此需采用平滑(smoothing)技术缓解该问题,为所有可能出现的字符串分配非零概率,避免零概率。平滑是调整最大似然估计以生成更合理概率的方法,其核心思路是提高低概率、降低高概率,让整体概率分布更均匀。

主要的平滑方法有:

① 加一平滑:为所有 N 元语法出现次数加一,从而避免零概率。

②加 K 平滑:使用小于 1 的常数 K 替代 1,更灵活地调整概率分布,从而缓解数据稀疏。

③Good-Turing 平滑:重新分配高频与低频的 N 元语法的概率,从而减少数据稀疏。

④Kneser-Ney 平滑:考虑词语在不同上下文中的多样性,从而减少数据稀疏。

⑤回退与插值平滑:在数据稀疏时回退到低阶 N 元语法或组合多阶的模型。

N 元语言模型从整体上来看与训练语料规模和模型的阶数有较大的关系,不同的平滑

算法在不同情况下的表现有较大的差距。Kneser-Ney 平滑是目前最有效的平滑方法之一。

平滑算法虽然较好地解决了零概率和数据稀疏的问题,但是,N 元语法模型仍然存在三个较为明显的缺点:

①N 元语法模型无法给长度超过 N 的上下文建模;

②N 元语法模型需要依赖人工设计数据平滑的规则;

③在 N 元语法模型中,当 N 增大时,数据的稀疏性随之增大,模型的参数量成指数级增加,并且模型受到数据稀疏问题的影响,其参数难以被准确地学习。

此外,N 元语法模型还忽略了单词之间的相似性。为了弥补这些缺憾,基于神经网络的大语言模型(Large Language Model, LLM)便逐渐成为研究热点(冯志伟 等,2024)。

4 N 元语法模型与大语言模型

现在的大语言模型的核心机制是“预测下一个词元”(next token prediction),其数学原理是“所罗门诺夫归纳推理”(Solomonoff's theory of inductive inference)。这里我们用尽可能通俗的方式介绍这个极为重要的归纳推理。

美国数学家所罗门诺夫(Ray Solomonoff)以“归纳推理的形式理论”(A Formal Theory of Inductive Inference)为题,于 1964 年分两个部分分别发表在计算理论的重要刊物《信息与控制》(*Information and Control*)的两期之上,提出了所罗门诺夫归纳推理(Solomonoff, 1964:1-22)。

所罗门诺夫归纳推理可以如下定义:

给定序列(x_1, x_2, …, x_n),预测 x_{n+1}。归纳推理就是力图找到一个最小的图灵机(Turing machine),可以为(x_1, x_2, …, x_n)建模,从而准确地预测后续序列。

例如,如果一个序列是 n 个 1:(1, 1, 1,…),那么我们可以写出如下程序输出该序列:

```
for i := 1 to n
print 1
```

这个序列的描述长度就是 O(log(n))。

又如,如果我们给出序列(3, 5, 7),会有无穷多种预测后续的结果,其中一种是 9,因为程序有可能打印奇数,程序如下:

```
for i := 1 to n
print 2i+1
```

但也许计算机猜得不对,还有一种可能性是 11,因为程序也有可能是打印素数的。很

明显,打印素数的程序就要比打印奇数的程序复杂很多,也就是说素数的描述长度要大于奇数的描述长度。

这个“在下一个字符上下赌注”(bet on next symbol)的接龙问题,其实就是大语言模型的核心机制:“预测下一个词元。”

这个意义上,我们认为,大语言模型的数学原理就是“所罗门诺夫归纳推理”,而 N 元语法模型则是其语言学理论基础。

在 N 元语法模型中,当 N 增大时,数据的稀疏性随之增大,模型的参数量呈指数级增加,其参数难以被准确地学习和计算。

对此,本吉奥等人(Bengio et al. ,2000)提出了使用前馈神经网络(Feed-forward Neural Network,FNN)的语言模型。在前馈神经网络中,单词被映射为一个低维稠密的实数向量,叫做词向量(word vector),文本中的单词都被嵌入到向量空间中去,称为词嵌入(word embedding)。这样一来,自然语言处理就从词语的符号运算变为实数的矩阵运算,巧妙地解决了 N 元语法模型的数据稀疏问题,大大提高了运算的效率。

把 David、John、play、Mary、loves、like 等单词都映射到向量空间中,将其转化为用实数表示的词向量,我们便得到如下所示的单词向量空间分布图(见图 3)。

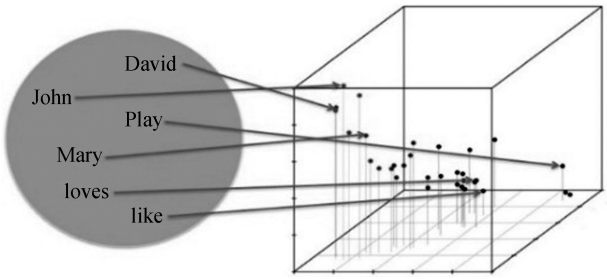

图 3　把单词映射到向量空间中

在图 4 中,句子“All data in deep learning must been represented as vectors”中的所有单词,在向量空间中都被映射成了词向量。

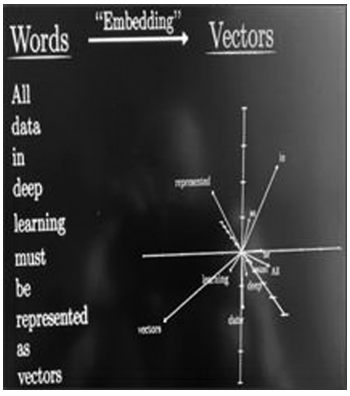

图 4　从单词(Words)到词向量(Vectors)

在向量空间中,转化成词向量的单词,构成了嵌入矩阵(embedding matrix),离散的语言符号就可以用矩阵来计算了,这样语言问题就变成了矩阵运算(matrix calculation)的数学问题。

因此,从数学原理来看,大语言模型是 N 元语法模型的新发展,相较于 N 元语法模型,大语言模型方法可以更加有效地避免零概率和数据稀疏问题,有些模型还可以避免对文本长度的限制,从而更好地给长距离依赖关系建模,大大提高了语言模型处理自然语言的能力。

2017 年,谷歌的瓦斯瓦尼等(Vaswani et al., 2017)8 人提出了 Transformer 架构,发表论文“注意力就是你需要的一切”(Attention Is All You Need),在机器翻译任务上取得了突破性进展,揭开了大语言模型研究的序幕。据说这篇论文在学术会议上宣读后,引起了数百位与会者的热烈讨论。

图 6 是 arXiv 论文库中自 2018 年 6 月以来包含关键词 Language Model(语言模型)以及自 2019 年 10 月以来包含关键词 Large Language Model(大语言模型)的论文数量趋势图。不难看出,2021—2023 年间以 Language Model 和 Large Language Model 为检索词的文章数量猛增。

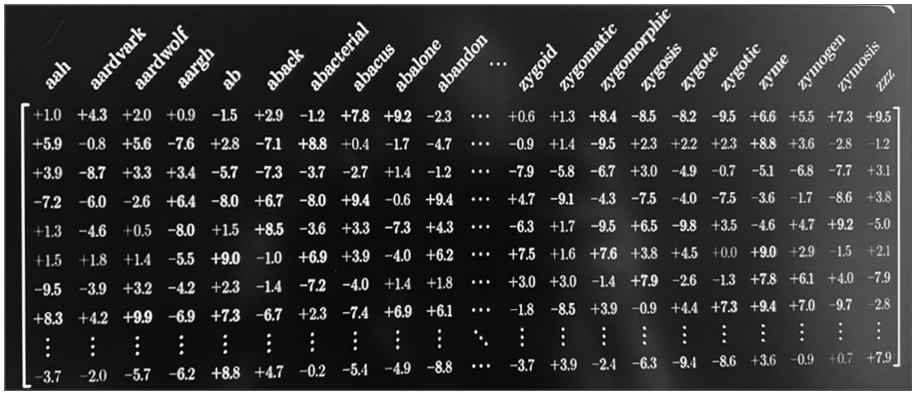

图 5 包含关键词 Language Model 和 Large Language Model 的论文数量趋势图

大语言模型的核心是 Transformer,主要有 5 个部分:

①词元化(Tokenization):把输入文本切分为词元(token)。

②嵌入(Embedding):把词元转换成词向量,嵌入到向量空间中。

③位置编码(Positional encoding):给词向量加入位置信息。

④Transformer[①] 模块:使用注意力机制(attention),在解码器端得到输出结果。

⑤Softmax:将输出结果转换为概率,并使得每一个结果的概率总和为 1。

① Transformer 是 Vaswani 等研制成功的基于注意力机制的语言模型,是当今大语言模型的核心技术。由于至今没有恰当的中文译名,故本文采用英文原名。

深度神经网络需要采用有监督方法(supervised approach),使用有标注数据的语料进行训练,因此,语言模型的训练过程不可避免地需要构造训练语料。但是在大语言模型中,模型的训练仅需要大规模无标注文本即可。这样一来,大语言模型的训练也成为典型的自监督学习(self-supervised learning)任务。由于互联网的发展,大规模文本数据的获取变得越来越容易,因此训练超大规模的语言模型也成为了可能。

2018 年,以 ELMo 为代表的动态词向量模型开启了语言模型预训练的大门(Peters et al. ,2018)。此后,以 GPT(Radford et al. ,2019)和 BERT(Devlin et al. ,2019)为代表的基于 Transformer 构架的大规模预训练语言模型的出现,使用注意力机制,使得自然语言处理全面进入了预训练—微调范式(pre-training and fine-tuning paradigm)的新时代。将预训练模型应用于下游任务时,不需要了解太多的任务细节,也不需要设计特定的神经网络结构,只需要“微调”预训练模型,使用具体任务的标注数据在预训练语言模型上进行监督训练,就可以显著地提升系统的性能。

2020 年,Open AI 发布了由包含 1750 亿参数的神经网络构成的、基于 Transformer 构架的生成式大规模预训练语言模型 GPT-3(Generative Pre-trained Transformer 3),明显地提高了大语言模型的效率(Radford A et al. 2020)。

2022 年 11 月 OpenAI 公司推出的 ChatGPT 大语言模型可以精确地反映单词之间的关系,能够生成通顺流利的句子,达到了很好的效果。由于阶数高,数据多,计算难度非常之大,需要很强大的算力。2023 年 11 月 OpenAI 公司推出的 GPT-4 Turbo,其上下文长度有 128K 之多,这意味着 GPT-4 Turbo 能够理解超过 300 页纸张的文本量。

除了预训练—微调范式外,研究人员通过语境学习(Incontext Learning,ICL),提出了面向大语言模型的提示语(prompt)学习方法、模型即服务范式(Model as a Service,MaaS)、指令微调(Instruction Tuning)等方法,在不同任务上都取得了很好的效果。与此同时,Google、Meta、百度、华为等公司和研究机构都纷纷发布了包括 PaLM(Chowdhery et al. ,2022)、LaMDA(Thoppilan et al. , 2022)、T0(Sanh et al. ,2021)等不同的大语言模型。这些语言模型突破了 N-元语法模型数据稀疏的局限,能够表示神经元之间极为复杂的关系(冯志伟,2025)。

那么,在当前大语言模型的研究热潮中,N 元语法模型还有价值吗? 我们认为,在特定场景或约束下,N 元语法模型仍有不可替代的价值,它具有如下的优点:

①训练代价低:N 元语法模型只需对语料做一遍计数,就可以完成模型的训练,不需要 GPU(Graphic Processing Unit,图形处理单元)提供强大的算力,也不需要大语言模型的反向传播、参数调整等复杂计算。

②推理速度快、功耗低:N 元语法模型只需要少量的计算就可以得到概率,适合嵌入式、移动端、实时语音识别解码器等对功耗敏感的应用场景。

③数据需求量小:在 1-10 万句的小语种语料的低资源场景下,大语言模型往往出现欠拟合,而 N 元语法模型已经可以给出可用的先验概率,比较轻便。

④可解释性与可控性强:大语言模型缺乏可解释性、可控性,而 N 元语法模型具有很强的可解释性和可控性,其概率可直接回溯到具体的 N 元语法计数,便于人工注入规则,满足文本审校、学校教育的应用需求。

⑤参数即概率,无需存储整个网络:大语言模型几乎存储了整个互联网的数据,而 N 元语法模型是轻量级的语言模型,它的规模只与词典的容量和阶数 N 有关,对于必须驻留内存的解码器非常友好。

在大语言模型时代,N 元语法模型并没有“退休”,哪里要速度、要小数据、要轻便、要可解释性、要可控性,N 元语法模型就出现在哪里。例如,某非洲低资源语言(只有 20 万句的语料)做拼写纠错,用 5 元语法模型加 Kneser-Ney 平滑就把词错误率从 8.4% 降到 3.1%,而用大语言模型 fine-tune mBERT 反而因过拟合,词错误率为 5.6%,其效果远不如 N 元语法模型。

总而言之,N 元语法模型具有“快、小、轻、省”的特点,而且有较强的可解释性和可控性;当硬件资源、数据量成为硬约束时,N 元语法模型仍是务实且高效的选择。在大语言模型的时代,N 元语法模型仍然具有它的价值。我们应当把大语言模型与 N 元语法模型结合起来,把以模拟人类认知为基础的连接主义和以推理为核心的逻辑主义结合起来,推进人工智能的发展。

5 结论

N 元语法模型作为自然语言处理领域的经典方法,其背后是自然语言处理发展进程中最基础且核心的统计语言模型。N 元语法模型的核心意义在于将自然语言的序列性、上下文依赖性转化为可计算的统计规律。对于传统基于规则的自然语言处理模型来说,N 元语法模型无需依赖复杂的句法规则、语义知识库或语言本体研究成果,是典型的数据驱动型方法;对于目前主流基于海量数据的大语言模型来说,N 元语法模型对于数据的依赖程度并非不可或缺。可以说,N 元语法模型凭借其简洁性和高效性,在语音识别、OCR 等多个实际应用中发挥了重要作用。

尤其是在当前大语言模型的研究热潮中,各类模型均以高资源、大数据逻辑构建,要求语料数据规模大(一般为亿万级 token 数据)与硬件算力强(一般根据 token 训练技术要求

计算机高功率 CPU、大内存或高算力 GPU),对于语料数据量不足的低资源语言不友好,难以解决低资源语言“语料稀缺+缺乏训练”的双重约束,在目前应用开发的过程中,只能尝试通过数据增强、预训练模型迁移、半监督/无监督学习等策略,“打补丁”式降低大语言模型对数据规模的硬性需求。如果进一步考虑语言数据安全,要求脱机进行研究开发,那么将进一步降低大语言模型的效力。

同时,大语言模型背后的逻辑是基于词向量的数字计算,自然语言的各个词汇、短语等语言单位均转换成为矩阵中的数字,这些数字为何能代表人类语言的词汇、短语等我们不得而知,也就是说我们尚无法对大语言模型“技术上的成功”赋予人类知识性的解释。当面对应用出现的错误时,我们难以解释问题,也就难以“对症下药”。

N 元语法模型在特定场景下仍具有不可替代的价值。我们应当把大语言模型方法和 N 元语法模型做进一步的融合,从而形成更加轻量、更加具有可解释性的混合语言模型。

参考文献:

Bengio Y., R. Ducharme, P. Vincent et al. 2000. A Neural Probabilistic Language Model[G]//*Advances in Neural Information Processing Systems*,13:932-938.

Deng J., W. Dong, R. Socher et al. 2009. ImageNet: A Large-scale Hierarchical Image Database[G]//2009 IEEE Conference on *Computer Vision and Pattern Recognition*. IEEE:248-255.

Devlin J., M. W. Chang, K. Lee et al. 2019. BERT:Pre-training of Deep Bidirectional Transformers for Language Understanding [G]//2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). *Association for Computational Linguistics*:4171-4186.

Feng Z. 2023. *Formal Analysis for Natural Language Processing: A Handbook*[M]. Springer.

Hinton G. 2025. Will Digital Intelligence Replace Biological Intelligence? [EB/OL] WAIC Keynote Speech. Shanghai. 2025-08-26. URL: https://en.eeworld.com.cn/mp/aes/a403602.jspx

Jelinek F. 1997. *Statistical Methods for Speech Recognition*[M]. Cambridge: MIT Press.

Mikolov T., M. Karafiát, L. Burget et al. 2010. Recurrent Neural Network Based Language Model[G]//11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010). *International Speech Communication Association*, 2010:1045-1048.

Peters M. E. M. Neumann, M. Iyyer et al. 2018. Deep Contextualized Word Representations[G]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). *Association for Computational Linguistics*:2227-2237.

Radford, A., J. Wu, R. Child et al. 2019. Language Models are Unsupervised Multitask Learners[Preprint]. OpenAI blog. [2023-4-18]. https://pdfs.semanticscholar.org/41f9/45f59bd0d345d4e355fb72110524f6fdffdb.pdf

Radford A., K. Narasimhan, T. Salimans et al. 2018. Improving Language Understanding by Generative Pre-training[EB/OL]. OpenAI, San Francisco, CA, USA, 2018. arXiv:1801.06146. URL:https://openai.com/research/language-understanding.

Sukhbaatar S. ,A. Szlam, J. Weston et al. 2015. End-to-end Memory Networks[G]//*Advances in Neural Information. Processing Systems*(28):2440-2448.

Solomonoff R. 1964. A Formal Theory of Inductive Inference[J]. *Information and Control* (1):1-22.

Solomonoff R. 1964. A Formal Theory of Inductive Inference[J]. *Information and Control*(2):224-254.

Vaswani A., N. Shazeer, N. Parmar et al. 2017. Attention is All You Need[G]//*Advances in Neural Information Processing Systems*(30):5998-6008.

陈天莹,史晓东,刘群. 2006. 基于前后文 n-gram 模型的古汉语句子切分[J]. 中文信息学报(2):20-25.

冯志伟. 2017. 自然语言计算机形式分析的理论与方法[M]. 合肥:中国科学技术大学出版社.

冯志伟, 张灯柯. 2024. 人工智能中的大语言模型[J]. 外国语文(3):1-29.

冯志伟. 2025. 大数据时代的自然语言处理[M]. 北京:商务印书馆.

丹尼尔·朱夫斯凯,詹姆斯·马丁. 2018. 自然语言处理综论(第二版)[M]. 冯志伟,孙乐, 译. 北京:电子工业出版社.

N-gram Model and Its Relationship with LLM

FENG Zhiwei ZHANG Dengke

Abstract: A grammar model that predicts the N-th word using the preceding N-1 words is called an N-gram model. This is a probabilistic language model. The assumption that the probability of a word depends only on the probabilities of its preceding words is known as the Markov assumption. Based on the Markov assumption, we do not need to examine a word's distant past words to predict the probability of its future words. The basic bigram model can be viewed as a Markov chain where each word corresponds to a single state. We can extend the bigram model to a trigram model and further to an N-gram model. In natural language processing (NLP), bi-gram model or tri-gram model are typically used. Higher-order N-gram models increase computational complexity. Large language Models (LLMs) can be seen as further developments of N-gram models. The mathematical principle of LLMs is Solomonoff Inductive Inference, and the N-gram model is linguistics theoretical basis of LLMs. This paper also discusses the applications of N-gram models in NLP.

Key words: N-gram model; Markov assumption; data sparseness; NLP; LLM

责任编校:黄均