通过分词与变换器通向自然语言

David S. Berman[1, a] 和 Alexander G. Stapleton[1, b]

[1]伦敦玛丽女王大学理论物理中心，Mile End Road，伦敦 E14NS，英国
[a]a.g.stapleton@qmul.ac.uk[b]d.s.berman@qmul.ac.uk

arXiv:2601.03368v1 [cs.CL] 6 Jan 2026

2026年1月8日

摘要

自然语言的统计结构呈现出显著的规律性，其中尤为突出的是齐夫定律和希普斯定律的出现。尽管如此，这些特性与当代 Transformer 模型所用的现代分词方案之间有何关联，目前仍不甚明了。本文假设词频服从齐夫频率分布，分析多个语料库的信息含量（以香农熵衡量），并推导出槽位熵期望值的闭式表达式。随后，我们通过实证研究考察字节对编码（BPE）如何改变语料库统计特性，发现递归应用 BPE 会使词元频率趋向齐夫幂律，同时使经验熵呈现出特征性的增长模式。利用 Transformer 学习依赖上下文的词元概率分布的能力，我们在采用不同 BPE 深度进行分词的语料库上训练语言模型，结果显示，随着 BPE 深度增加，模型的预测熵与基于齐夫定律的预测越来越一致。基于注意力的诊断进一步表明，更深层的分词会减少局部词元依赖关系，使经验分布更接近弱依赖（近似独立同分布）状态。总之，这些结果阐明了 BPE 不仅是一种压缩机制，也是一种统计变换，能够重构自然语言的关键统计特性。

1 引言

Transformer 是当今生成式语言模型取得成功的基础。它们能够自回归地生成文本，源于它们学会了序列中给定前序词元时某个词元的条件概率分布。通过获得这一条件概率分布，模型便隐式地学会了自然语言中依赖上下文的信息含量。

以这种方式研究语言的信息含量并非新鲜事；探究语言的信息含量可以追溯到香农，他将热力学思想，尤其是熵的概念，应用于通信信道。香农根据各种词元出现的概率来定义熵。他的这一方法开创了信号处理中信息含量的解读与计算方式。不过在实践中，真实的联合词元概率是未知的，只有假设词元彼此独立分布时，才容易进行估算。多年来，研究语言信息背景一直采用这种方法。如今，随着大语言模型的出现，我们有机会训练模型来学习词元的完整条件分布，并以此描述和探究语料库中的信息。我们预计，随着学者逐渐认识到这些模型学会的不仅是预测下一个词元，还有词元的完整概率分布，使用大语言模型研究语言统计特征的做法将日益普遍。这可能会对信号处理、密码学等领域产生影响；在这些领域，语言的真实统计属性至关重要，也可能通过使用大语言模型得以揭示。

在这项工作中，我们希望研究语言的一项众所周知的统计属性——齐夫定律——与语言模型分词深度之间的关系。在深入探讨本文的细节之前，值得先指出的是，分词过程的研究中一直存在某种空白

以及它对大语言模型最终性能的影响。大多数大语言模型中的分词深度都是通过经验确定的超参数选择。从更抽象的角度来看，分词层级类似于考察语言时所采用的尺度，其中以原始字母为尺度的词元最为细粒度。随着分词深度提高，词汇表会不断扩大，子词、词语，甚至潜在的多词和句子级词元都会出现。这一选择类似于物理学中所谓的 重整化 尺度。统计场论是在特定长度尺度下描述的，而该尺度决定了它们的统计属性。所谓的 重整化群（*RG*）流 描述了当尺度发生变化时这些属性如何改变，这相当于诱导模型参数发生流动。（关于统计模型中的学习与精确重整化群流之间的联系，请参见 [1, 2]。）这类理论中的一个关键结果是所谓的 c定理，它说明了统计模型经历 RG 流时，其中的熵如何变化。受这一物理学现象的启发，我们将在语言中提出一个等价问题：“当我们改变分词深度时，语料库中的熵如何变化？”

Transformer 提供了一种自然的统计视角来回答这个问题，因为 softmax 给出了一个可信的概率分布，可供研究。我们使用该分布评估熵，然后在假设文本遵循齐夫定律的前提下，解析地计算文本的期望熵。比较这些结果后，我们可以证明，随着分词加深，经验熵（由 Transformer 给出）会越来越接近齐夫分布的熵的解析表达式。

本文的结构如下。我们首先介绍齐夫定律，接着概述 Transformer 模型和字节对编码的背景知识。随后，我们推导齐夫语料库的解析熵，并讨论将齐夫分布拟合到经验数据的稳健方法。原则上，单个词元频率足以确定齐夫拟合结果，但我们表明，将拟合范围限制在中等秩的词元上，能显著提高经验稳定性。

基于这一框架，我们测量熵随分词深度的变化，并将结果与齐夫定律的理论预测进行比较。最后，我们研究注意力机制揭示的文本中特有的尺度，这为理解齐夫统计规律的涌现提供了见解。

1.1 齐夫定律入门

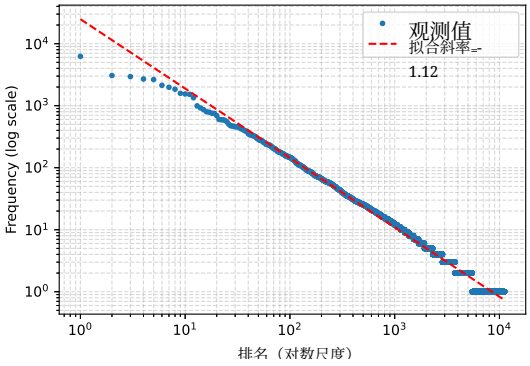
总的来说，几乎所有自然语言都有一个显著特征：在规模足够大的文本语料库中，词语秩的分布大致遵循齐夫定律 [3]。齐夫定律以语言学家 George Kingsley Zipf[1]，命名，指出一个词出现的频率 f_s 与其秩成反比 r_s，

$$f_s \sim \frac{1}{r_s}, \tag{1}$$

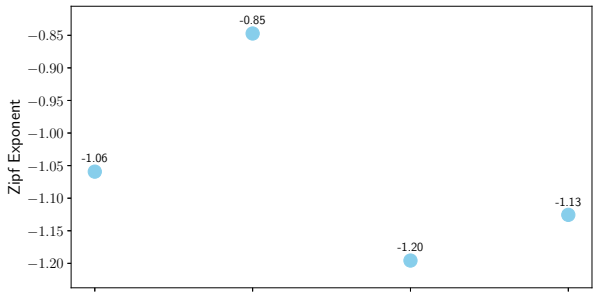
其中，词语的秩指的是它在按频率从高到低排列的频率表中的位置。尽管齐夫定律由来已久，它仍是一个活跃的研究课题；现代研究持续考察幂律行为在神经网络等现代应用中的表现 [5, 6]。齐夫定律几乎出现在所有语言中 [7]，，许多研究将幂律行为的实证观察 [6, 8]扩展到人类话语之外，甚至延伸至动物交流（例如，参见 [9]，该研究观察到了鲸鱼交流中的齐夫定律）。虽然齐夫定律尚未得到严格推导，但研究表明，如果每个 n-元组都具有相同的概率 [11]，便可从大型语料库的统计数据中推导出一种推广形式（曼德布罗-齐夫定律 [10]）——当然，这种简化假设并不适用于英语（例如，字母“e”的出现频率约为字母“z”的170倍 [12]）。

尽管齐夫定律存在许多可视为高阶修正的推广形式[2]，但其最基本的表述对于许多语料库而言仍然是非常出色的近似[3] （例如，见图1b）。图1a明确展示了频率与秩的完整对应关系，而图1b则展示了针对若干常见数据集拟合得到的齐夫指数。

[1]尽管齐夫在20世纪30年代观察到了幂律行为， [3], “齐夫定律”实际上可以追溯到埃斯图，[4], 他于1916年发表了这一发现。[2]其中或许最值得一提的是前文所述的曼德布罗结果，该结果指出 $f_s \sim \frac{1}{(r_s-a)^b}$ （其中 $a, b \in \mathbb{R}$ 由数据确定）。[3]在本文中，我们始终采用指数为 $\alpha = -1$ 的齐夫定律理想化形式。经验偏差则作为次主导修正处理。



(a) IMDB 数据集子集的齐夫分布行为



(b) 四个语料库子集的拟合齐夫指数

图 1：各数据集的齐夫统计。

1.2 Transformer 中的编码

机器学习和人机交互发展史上最具影响力的进步之一，可以说是Transformer架构。

Transformer 是一类特别适用于序列处理任务的神经网络架构[13–15], ，能够通过所谓的自注意力机制有效处理数据。Transformer 于 [14], 年问世，取代了循环架构，并支持并行处理输入词元，其样本质量和模型效能通常远超前代架构。Transformer 能够编码长程依赖和上下文相关特征，因此几乎已被普遍应用于各种场景，包括搜索引擎以及几乎所有现代大语言模型，例如[16–18]。

通用的 Transformer 语言模型以离散词元序列为输入，但模型本身处理的是连续向量。因此，需要一种编码方案，将词元映射为模型可以处理的向量表示。当然，可以自由选择（几乎）任何方案来编码 n-元组序列，但一个自然的问题是，应该如何优化该方案，使其具备更理想的特性，例如提升模型性能、加快损失收敛等。研究发现，当基于 Transformer 的模型所采用的分词方案遵循齐夫定律 [5] 时，通常表现更佳。最常见的编码类别之一称为子词编码 [19]。其中，或许最普遍的一种是字节对编码 [20], ，它能提供均衡的词汇表，同时避免词级词汇表急剧膨胀以及字符级模型效率低下的问题。这些编码使 Transformer 能够学习丰富的语义关系，同时保留语言的序列结构。

字节对编码（BPE）分词算法通过迭代合并出现频率最高的相邻符号对，构建更紧凑、因而可能更高效的词汇表。设语料库表示为由 M 按规范顺序排列的 [4]词元序列组成的集合，

$$\mathcal{D} := \{m_1, \ldots, m_M\},$$

其中每个 m_i 代表构成一条消息的一系列词元。消息中独一无二的 n元语法构成递归深度为 ι时的词汇表 $V^\iota = \{s_i^\iota\}_{i=0}^{V^\iota}$。

在每次迭代中，算法都会找出频率最高的一对相邻 n 元语法，

$$(\tilde{\mu}, \tilde{\nu}) = \arg\max_{(\mu,\nu)} \mathrm{freq}(s_\mu^\iota s_\nu^\iota),$$

其中，freq($s_\mu^\iota s_\nu^\iota$)表示语料库中拼接后的相邻对 $(s_\mu^\iota s_\nu^\iota)$ 的出现次数。为提高计算效率，实际操作中，自然不会贪心地构造所有对并逐一检查频率，而只会检查在本次迭代开始时数据集中已出现的对。

[4]这里的规范排序是指词元在自然语言中说出或写下时的排列顺序。

然后将最常见的一对合并为一个新符号 $\tilde{s}^{\iota}$，并根据规则替换每个序列中的所有出现项

$$m_i \leftarrow \mathrm{merge}_{s_{\tilde{\mu}} s_{\tilde{\nu}}}(m_i), \qquad i = 1, \ldots, M.$$

经过 ι 次迭代后，得到的词汇表由原有的 V^0 个词元以及新引入的 ι 个词元组成。最终的分词方案能够捕捉高频子词模式，同时对罕见或此前未见过的序列保持稳健。

由此可见，字节对编码方案可以稍作推广：每次迭代不必只执行一次合并，也可以执行多次。在这种情况下，该算法由两个参数决定：合并次数[5] ϕ 以及迭代次数（或递归次数） ι。在所有应用字节对编码的实验中，我们都将合并次数设为 1，即 $\phi = 1$，因此每次迭代都会形成一个新的词元。

2 齐夫分布语料库的熵

2.1 齐夫分布语料库的理论预测熵

本节考察齐夫定律的观测结果对统计模型的熵施加了哪些限制。考虑香农著名关系式 $S = -\sum_i p_i \log\, p_i$ 的一般形式，其中 p_i 是事件 Prob($X = x_i$) 发生的概率。设 $\mathcal{D} = \{m^{\mu}\}_{\mu=1}^{M}$ 为一个数据集，由从消息分布中抽取的 M 个样本构成，其中每条消息 m^{μ} 包含 N^{μ} n 个 n 元组。所有 n 个 n 元组构成词汇表 V。假设所有消息都是独立同分布（*IID*）的，那么在大 Γ 极限下，类型为 s 的一个 n 元组 n 的出现概率，可以通过经验近似为:

$$p_s = \frac{\sum_{\mu=1}^{M} n_s^{\mu}}{\Gamma}, \tag{2}$$

其中， $\Gamma := \sum_{\mu=1}^{M} N^{\mu}$ 是数据集中 n-元组的总数，而 n_s^{μ} 是第 μ条消息中类型为s的 n-元组数量。

此外，可以将每条消息 m^{μ} 视为由 N^{μ} 个槽位组成的集合，每个槽位都由一个具有给定概率的 n-元组填充。每条消息的长度也对应一个内在概率（例如 N^{μ}），不过在后续计算中，我们始终假定该长度已预先确定。

在最一般的情形下，在不作任何 IID 假设之前，支配消息 m^{μ} 中位置 i 的概率分布的熵定义为

$$S_i^{\mu} := -\sum_{s=1}^{V} p_{s,i}^{\mu} \ln(p_{s,i}^{\mu}), \tag{3}$$

其中 $p_{s\ ,\ i}^{\mu}$:=Prob($t_i^{\mu} = s$) 表示消息 m^{μ} 中位置 i 处的[6] t_i^{μ} -元组属于类型 s的概率。因此，可以计算所有消息及每条消息中所有槽位的平均槽位熵。由于满足 IID 条件，该熵为

$$S_* = \frac{1}{M} \sum_{\mu=1}^{M} \sum_{i=1}^{N^{\mu}} \frac{1}{N^{\mu}} S_i^{\mu} = S_0^0. \tag{4}$$

在这种情况下， S_* 与熵密度类似，因为它是一个强度量。

省略位置索引和消息索引（由于每个槽位均为独立同分布，因此它们不包含额外信息）后，每个槽位的词元分布的平均熵可写为:

$$\mathbb{E}[S] = -\sum_{s=1}^{V} \frac{n_s}{\Gamma} \ln(n_s/\Gamma) \tag{5}$$

$$= \frac{1}{\Gamma} \sum_{s=1}^{V} n_s (\ln \Gamma - \ln n_s), \tag{6}$$

[5]便于记忆地说，应将 ϕ与 $formations$的数量联系起来。[6]作为一种记忆方法，可以将 n-gram t_i^{μ} 视为第 μ条消息中的第 i个 $token$ 。

其中，在第一行中使用（2）来近似 p_s。

如果 n-元组的分布遵循 (1)，则可写为 $n_s = \frac{\gamma}{r_{s'}}$，其中索引 $s' = \rho(s)$ 满足 $\rho(s)$，它是一个排序函数，用于给出频率从高到低排列的列表中第 s 个 n-元组的位置[7]，而 γ 是数据集 $\mathcal{D}$ 固有的比例常数。定义有序频率 $f_s = n_{\rho(s)}$（满足 $f_1 > f_2 > \ldots f_N$）会有所帮助。根据齐夫定律，我们注意到一个虽显而易见却很了不起的事实：

$$\frac{f_s}{f_{s+1}} = \frac{s+1}{s} \implies f_{s+1} = \frac{s}{s+1} f_s, \tag{7}$$

其中，反复应用 (7) 可得出一个重要的推论：

$$f_s = \frac{f_1}{s}. \tag{8}$$

考虑 (6) 中第二项的内层求和，由于 ρ 是双射，且求和次序可以交换，因此可写为

$$\begin{aligned}
\sum_{s=1}^{V} f_s \ln f_s &= f_1 \ln f_1 + f_2 \ln f_2 + \ldots + f_V \ln f_V && (9)\\
&= f_1(\ln f_1 + \frac{1}{2}\ln f_2 + \ldots + \frac{1}{V}\ln f_V) && (10)\\
&= f_1(\ln f_1 + \frac{1}{2}\ln \frac{f_1}{2} + \ldots + \frac{1}{V}\ln \frac{f_1}{V}) && (11)\\
&= f_1(\ln f_1 + \frac{1}{2}(\ln f_1 - \ln 2) + \ldots + \frac{1}{V}(\ln f_1 - \ln V)) && (12)\\
&= f_1 \ln f_1 (1 + \frac{1}{2} + \ldots + \frac{1}{V}) - f_1(\frac{1}{2}\ln 2 + \ldots + \frac{1}{V}\ln V) && (13)\\
&= f_1 \ln f_1 H_V + f_1 \alpha(V) && (14)
\end{aligned}$$

其中，H_V 是第 V 个调和数，而 $\alpha(V) := -\sum_{s=2}^{V} \ln_s s$。回到 (6) 中的第二项，可类似地得到：

$$\sum_{s=1}^{V} f_s \ln \Gamma = \ln \Gamma f_1 (1 + \frac{1}{2} + \ldots \frac{1}{V}) = H_V f_1 \ln \Gamma, \tag{15}$$

其中，第二个等式是再次注意到 $f_s = f_1 f_s / f_1 = f_1 / s$ 后得到的。

回到熵的计算，我们得到了这个令人欣喜的表达式

$$\mathbb{E}[S] = \frac{f_1}{\Gamma}[H_V \ln \Gamma - \ln f_1 H_V - \alpha(V)], \tag{16}$$

它仅取决于整个数据集中共有的单个 n-元语法频率、词汇表大小 V，以及语料库中的 n-元语法总数。我们注意到，当词汇表大小 V 有限时，熵保持有限；在极限 $V \to \infty$ 下，对于齐夫指数 $\alpha \leq 1$，熵按对数发散。当然，在前面的计算中，将式 (16) 写成关于 f_1 的形式时，隐含地作出了一种选择。利用齐夫恒等式 (8)，式 (16) 可以更一般地写为

$$\mathbb{E}[S] = \frac{s f_s}{\Gamma}[H_V \ln \Gamma - \ln(s f_s) H_V - \alpha(V)]. \tag{17}$$

2.2 数值演示

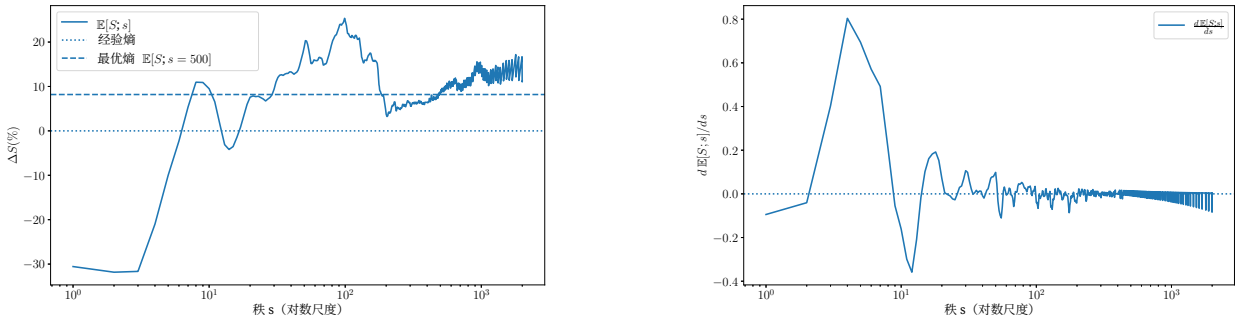
在本小节中，我们将进行一系列数值演示，以考察语料库的经验熵和预测熵，这些熵由式 (16) 计算得到。

[7]为了确保后续计算在数学上严谨，我们要求 ρ 必须是单射且满射（即双射函数）。

2.2.1 熵计算对 s 的依赖性

正如式（17）所示，需要为秩为 s 的词元确定（通过经验拟合）其频率 f_s。为简化上述推导，选择了 f_1。事实上，直观上可以将位置 s 选在秩表的中间附近，因为齐夫定律在极端秩上不太适用。在高频端，最常见的少数几个词（例如常用动词）的出现频率往往高于理想化幂律曲线的预测；英语的一般语言结构使这些词被过度使用[21]。此外，在低频端，罕见词并不遵循预期的平滑衰减，这是因为抽样效应、领域特定行为，以及只出现一次词的存在——这类词在单个数据集中只出现一次 [21]。

在图 2 中，我们展示了预测熵对齐夫分布最明显的语料库（见图1b）的依赖性，即 20Newsgroups 数据集。图 2a 展示了预期结果：除去频率表尾部齐夫定律最显著成立的区域外，式（17）是一个足够接近的近似（误差约在 10% 以内）。为便于阅读，我们删去了式（17）失效的最右侧尾部。我们注意到，两个图中的噪声行为源于 f_s 中的自然波动——其中一些噪声已通过移动高斯平均滤波器[8] 得到抑制。



（a）预测熵与经验熵的偏差随词元秩的变化。

（b）预测熵的梯度随词元秩的变化。

图 2：经验熵计算对秩的依赖性，即 rank(s)，固定频率 f 的位置或 20Newsgroups 数据集。

鉴于以上讨论，一个自然的问题是，选择哪个秩 s 来构造预测齐夫熵才是最优的。理想情况下，所选秩应使预测熵保持稳定，即预测熵相对于 s 的一阶和二阶导数均为零。因此，我们将选择梯度始终较小且稳定的区域中的 s。在图 2b 中，我们绘制了预测熵的梯度随 s 的秩变化的曲线（r_s）；从图中可以看出，梯度始终较小且稳定的区域位于 10^2 和 $2 \cdot 10^3$ 之间。在我们的数据中，梯度在 $s = 500$ 附近最为稳定，此时预测熵 $\mathbb{E}[S;\ s = 500] = 8.05$。

[8]高斯滤波器的步幅宽度设为 10。 ter.

3 Em字节对编码（BPE）下齐夫定律的涌现

由于自然语言似乎几乎普遍遵循齐夫定律，一个自然的问题是，Transformer 中使用的各种编码方案是否也表现出相同的规律？本文将探讨分词中广泛使用的字节对编码方案。

测量语料库中拟合得到的齐夫指数 $\alpha(\{f_s\})$ 会很有帮助，其中 $\alpha(\{f_s\})$ 定义为通过直接进行普通最小二乘（OLS）回归，拟合秩及其相对频率所得最佳拟合直线的指数。

由于 BPE 根据频率迭代合并 n-gram 序列来构建词汇表，因此涌现出的词汇表极有可能与语言学上的语素或已有词语并不对应。判断 BPE 词元频率中是否涌现出齐夫分布，有助于深入了解该编码方案在多大程度上保留、扭曲或重新发现了自然语言的语言结构。这类分析对于理解分词选择如何影响模型行为尤其重要：偏离齐夫分布模式可能会影响语义信息在各词元之间的分配。既然自然语言在经验上遵循幂律，那么我们自然也希望编码中呈现这一特性。换言之，齐夫定律在 BPE 下持续成立，可能表明该算法继承了语言更深层的统计属性，并进一步证明它适合作为自然语言建模的稳健分词步骤 [22]。

3.1 数值结果

3.1.1 BPE的递归应用导向齐夫定律

本节考察在字节对编码算法经过不同迭代 ι和合并 m 后，与齐夫所预测的幂律之间的偏差。图4展示了不同编码层级下拟合得到的幂律指数 $\alpha(\{f_s\})$，这一结果与 [5] 的研究高度吻合；该研究在 BookCorpus数据集 [23]上考察了等效过程。从简单的逐字符编码开始，再扩展到许多 n-元组，我们观察到，随着字节对编码迭代次数增加，词元频率越来越符合齐夫定律。这一结果部分由 [11]的开创性研究所预测。已知 [11] ，在每个字符出现概率相等的情况下，对于随机文本， α 为

$$\alpha = \frac{\ln(V+1)}{\ln V}, \tag{18}$$

这意味着，随着词汇表大小增加，$\ln(V+1)/\ln V$渐近趋于 1。然而，这一论证并不完整。其主要贡献依赖于词元独立同分布这一事实。增加合并次数会产生频率低得多的词元，更多 n-元组趋近于独立同分布（类似泊松分布）的极限[9]。

图3展示了在不同迭代次数 ι 下对 α 进行的一系列拟合，其中 α 是描述不断演变的语料库中词元的频率与秩之间关系的常用指数。字节对编码过程结束时，几乎所有词元都落在齐夫线上，其中 $\alpha = 1$。

[9]例如，参见第4.2节。

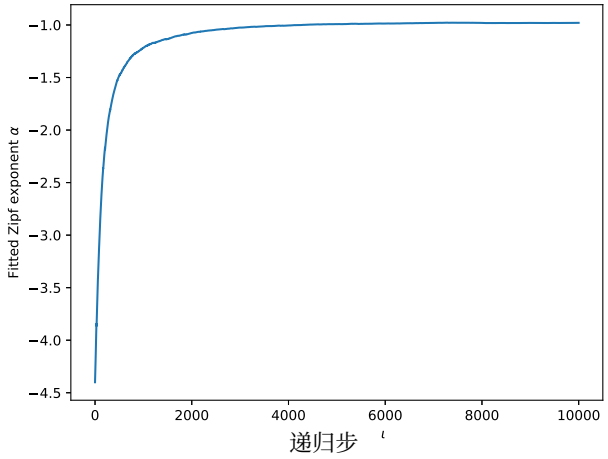
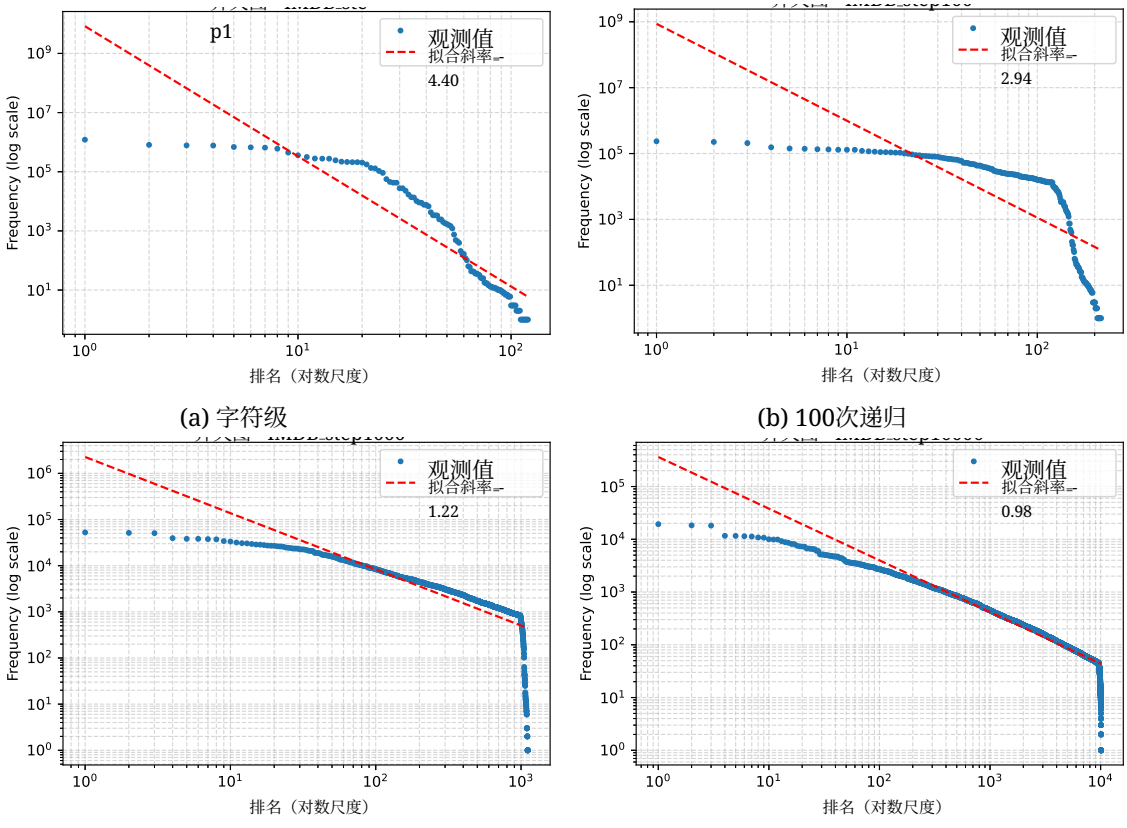


图3：递归应用BPE过程中 IMDB 数据集的拟合齐夫指数。



(a) 字符级　(b) 100次递归

(c) 1000次递归　(d) 10000次递归

图4：在 IMDB影评语料库上递归应用字节对编码，会生成齐夫分布特征越来越明显的语料库。

3.2 字节对编码下的经验熵

先前的讨论表明，熵可以作为衡量一组消息中随机性和信息含量的有效指标。由于第 2.1 节推导了数据集在齐夫极限下的熵，例如语料库反复应用字节对编码并经历大量迭代时的极限情况，因此探讨这一情形下数据集的理论熵和经验熵是很自然的。本节将考察在每次递归中合并数固定的条件下，对数据集递归应用字节对编码时的经验熵。

对于每次字节对编码迭代 ι，语料库可视为一个不断演变的数据集 $\mathcal{D}_\iota = \{m_\iota^\mu\}_{\mu=1}^{M}$，由 M 条消息组成，其中 m^μ 包含从一个定义在大小为 V 的词汇表上的底层分布中独立同分布抽取的 N_ι^μ 个已实现的 n-元组。消息的构成由第 1.2 节定义的算法生成。令 $n_{s,\iota} := \sum_{\mu=1}^{M} n_{s,\iota}^\mu$ 表示整个数据集中类型 s 的出现总次数，并令

$$\Gamma_\iota := \sum_{\mu=1}^{M} N_\iota^\mu$$

表示 $\mathcal{D}$ 中 n-元语法槽位的总数。在大-Γ_ι 极限下，观测到类型为 s 的词元的经验概率近似为

$$p_{s,\iota} \;=\; \frac{n_{s,\iota}}{\Gamma_\iota},$$

与式 (2) 相同。省略消息和位置索引（在 IID 假设下，它们不携带额外信息），则经验平均槽位熵为

$$(S_\iota)_* \;=\; -\sum_{s=1}^{V} p_{s,\iota} \ln p_{s,\iota} \;=\; -\sum_{s=1}^{V} \frac{n_{s,\iota}}{\Gamma_\iota} \ln\!\left(\frac{n_{s,\iota}}{\Gamma_\iota}\right), \tag{19}$$

这与式 (5) 一致。因此，对于每个 BPE 递归步骤，计算由当前分词产生的计数 $\{n_{s,\iota}\}$，根据式 (19) 的定义计算熵，并记录所得的经验熵 S_*。

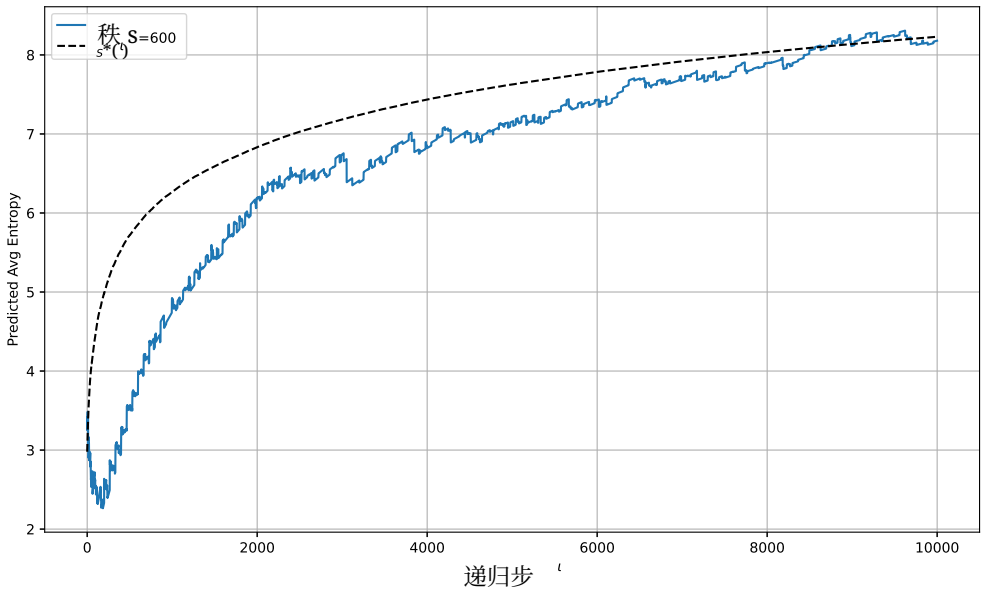


图 5：经验熵和预测熵随 BPE 递归步 ι 的变化。

图 5 展示了根据 (19) 得到的 $\mathbb{E}[S]$ and S_* 随 BPE 递归步骤 ι 变化的情况。对于较小的 ι，词汇表大小 V 足够小，且分布 $n_{s,\ \iota}$ 中低频类型相对较少，因此熵的度量值较低。随着合并过程推进，且

词元清单变得越来越精细，词汇表不断扩大，并出现许多新的低频类型。因此，在最初的数百个递归步骤中，经验熵迅速上升。当合并深度较大时，这种增长逐渐放缓，因为额外的合并往往会引入极罕见的类型，它们对 $-\sum_{s,\ \iota} p_{s,\ \iota} \ln p_{s,\ \iota}$ 的贡献很小。最终曲线呈现出先急剧上升、随后长时间缓慢增长的趋势。

经验结果表明，按照式（19）测量的语料库熵，会随着分词粒度的增加而单调上升。曲线的凹形说明存在边际效益递减：随着 ι 增大，词汇表随之扩充，经验分布也因此变得更广泛 $p_{s,\ \iota}$；然而，当 V 较大时，每个新生成的稀有类型所带来的增量贡献会变小。这凸显了 BPE 分词固有的权衡：细化词汇表能够提升观测类型的表达能力和多样性，但熵随之增长，反映出经验分布的尾部更重，计数稀疏性也日益显著$n_{s,\ \iota}$。此外，图 5 展示了使用式（19）得到的预测熵，其中 s 取值使得 rank(s) 位于频率表的中部，且期望熵相对于 s 变化的梯度足够小（此处的梯度大致与图 2b 相当，因此为简洁起见不再展示）。与第 2.2.1 节一样，我们再次观察到，当 f_s 足够小时或足够大时，熵预测在尾部会失效；但只要 rank(s) 位于数据集的中部，预测就足够准确。

4 使用 Transformer 进行探索

此前的讨论仅围绕大型数据集的经验测量展开，且未涉及模型。在本节中，我们将探索一个在 IMDB 数据集 [25] 上训练的 GPT-2 Transformer [24]。使用 Transformer 网络并对逻辑值应用 softmax 激活函数，即可为词汇表中的每个位置构建定义明确且经过正确归一化的概率分布。通过 softmax，可以定义真正的熵，而不受大型数据集极限的影响。因此，我们提出以下实验：考虑一组选用文本补全目标进行训练的网络，每个网络对应不同的分词级别。在第 3.1.1 节中，我们通过经验观测发现，在较大的字节对编码应用极限下，齐夫偏差 $|\alpha(\{f_s\}) - 1|$ 会趋近于 0。

4.1 数值设置

为了分析分词级别如何改变模型输出的信息特性，有必要区分熵的两个互补视角。语言模型生成文本时，会在每一步通过 softmax 函数生成一个涵盖整个词汇表的概率分布。最终选取哪个词元，取决于从该分布中进行的抽取。因此，我们可以通过考察预测分布本身的熵，或分析由此产生的采样词元序列的熵，来刻画模型的行为。前者反映模型在采样前的内部不确定性，后者则体现其实际输出的经验多样性。需要注意的是，这两种熵的概念虽然密切相关，但在小样本极限下并不等同。

模型 Softmax分布的熵。为了量化模型在生成时的不确定性，我们评估每个输出步骤的 softmax 分布熵。给定一个模型参数化形式，它会在大小为 V 的词汇表上产生逻辑值 $z \in \mathbb{R}^V$，相应的词元概率通过 Softmax变换获得

$$p_s = \frac{\exp(z_s)}{\sum_{s'=1}^{V} \exp(z_{s'})}. \tag{20}$$

瞬时预测熵定义如下：

$$H(p) = -\sum_{s=1}^{V} p_s \log p_s, \tag{21}$$

它衡量的是模型对词元空间的信念分布的离散程度。这种熵反映了模型在采样或解码之前的内部不确定性，因此与任何特定的生成

策略无关。通过在模型输出语料库上计算这一量，可以刻画分词粒度如何影响预测分布的尖锐程度或平坦程度，尤其是在词汇表不断扩大的情况下，以及 BPE 合并逐渐达到词元频率近似齐夫分布结构的状态时。

采样词元序列熵。 除了分析模型的预测熵，我们还可以考察模型经过 softmax 后生成的采样词元序列的经验熵。设 $\{x_t\}_{t=1}^{T}$ 表示生成的序列，其经验词元频率为 f_s。在 IID（用于式 (3)）与 BPE 迭代次数较大（ι）同时成立的极限下，预期分布将趋于齐夫分布，并且对于足够合理的 s，式 (17) 将占主导地位。

直观而言，样本熵

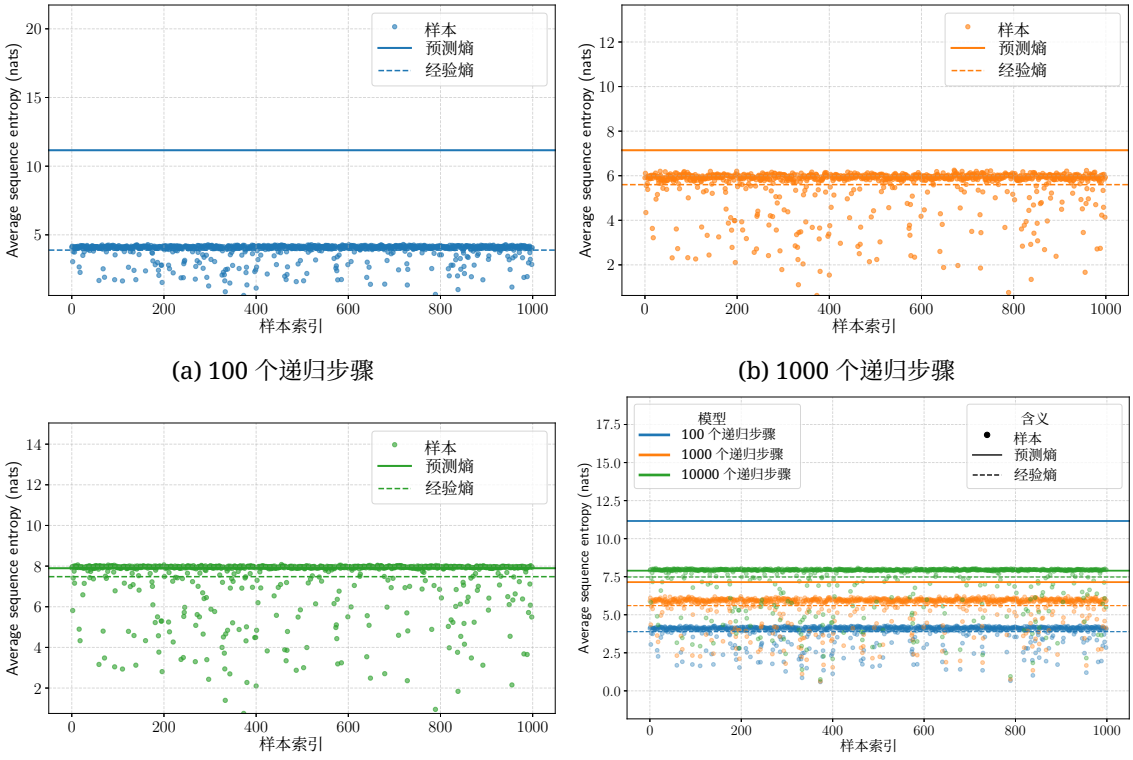
$$H_* = -\sum_{s\in\mathcal{V}} p_*(s)\log p_*(s) \tag{22}$$

反映了模型实际生成的输出在词汇表上的分布情况。与反映分布层面模型不确定性的基于 softmax 的熵不同，样本熵体现了生成文本的实际多样性，既取决于预测概率，也取决于采样过程引入的随机性。在大样本、大词汇表的 BPE 极限下——此时语料库趋近齐夫分布——这两种熵概念之间的关系有助于理解模型如何将概率质量分配到越来越细粒度的 n-gram 结构上。

4.1.1 数值结果

图 6 展示了以 IMDB 语料库[10]中的消息为条件时，Transformer 输出的经验平均熵和预测平均熵。此前的实验（例如第 3.1.1 节）表明，词汇表较大的模型（即编码应用次数更多的模型）更符合齐夫定律。Transformer 网络的熵也间接印证了这一点。图中以虚线表示经验平均熵，以实线表示基于 IID 假设得到的预测平均熵。在字符级附近（100 个递归步骤），公式 (16) 对熵的估计约为实际值的三倍。这是由两个因素造成的：要使用上述公式，每个词元都必须独立同分布；但在这样的递归深度下，这一极限尚未充分达到（见第4.2节）。其次，更明显的是，在合并深度较小时，Transformer 输出并不太符合齐夫定律。对于这个数据集，随着递归深度增加，预测熵从上方逐渐逼近经验值；在递归深度较大的极限下，两者高度吻合。图 6 汇总了各个图表，更清楚地展示了这一行为。

[10]如前所述，必须在公式（17）中对 s 作出选择。这里我们选择 $s=1$，这样更稳妥，因为 Transformer 输出的频率尾部更符合齐夫定律。



(a) 100 个递归步骤

(b) 1000 个递归步骤

(c) 10000 个递归步骤

(d) 综合

图 6：网络 softmax 函数在合并深度逐渐增加时的熵。

4.2 IID 假设的有效性

在第 4.1.1 节中，讨论依赖于这样一个事实： n-元组是独立同分布的。鉴于 Transformer 模型，我们可以通过以下方式探究对 IID 假设的违反情况。考虑一个训练得足够充分的 基于 Transformer 的大语言模型[11]中的任意一个注意力头。

在注意力机制 [14],中，非对角线上的键查询交互可以被解释为输入语料库中 n-元组之间统计依赖的一种度量。设 $Q \in \mathbb{R}^{T\times d}$ 和 $K \in \mathbb{R}^{T\times d}$ 分别表示长度为 T的序列的查询矩阵和键矩阵，并令 A 为如 [14] 所示的未归一化注意力分数矩阵。对角线元素 A_{ii} 反映位置 i 关注自身的强度，而对于 $i \neq j$ ，非对角线项 A_{ij} 则表示不同 n-元组之间的交互。

如果语料库呈现出较强的跨词元依赖，那么对于许多 i ，都存在 $j \neq i$ 使得 A_{ij} 很大，这表明位置 i 上的查询会在其他位置找到其最近邻（位于键空间中）。相反，如果 n-元组基本独立，矩阵 A 将近似为对角矩阵，即

$$|A_{ij}| \ll |A_{ii}| \quad \text{for most } i \neq j, \tag{23}$$

这表明每个 n-gram 主要与自身或其局部上下文交互。

因此， A中非对角线元素绝对值的分布，为衡量 n-gram 的独立性提供了自然的定量指标：较大且结构明显的非对角线数值意味着存在依赖关系，而较小或分散的非对角线数值则表明语料库中的词元相对独立。

在实践中，现代语言模型拥有大量注意力头，这意味着必须多次重复这一过程。通常会引入邻居距离截断值，以降低计算复杂度。这是对自然语言的一种合理假设：如果一对词元在时间上相距甚远，那么它们之间存在有意义相关性的可能性就会显著降低。鉴于词元具有规范的（时间）顺序，可以为消息中的每个词元建立一组局部注意力值的均值，即

$$D_i := \{a_{ij} \mid |j - i| \leq \Lambda\}, \tag{24}$$

对于给定的截断值 Λ。随后， $\{D_i\}$ 的分布便可作为衡量每个n-gram 邻居相关性的指标。

由于本实验使用的 GPT-2 模型包含许多注意力头，因此必须选择要考察哪些注意力头。图 7 展示了模型中所有注意力头的合并 A_{ij} 直方图。每个注意力头单独来看，也都呈现相同的行为（有关模型中注意力头数量的详情，请参见附录 4.1）。可以看出，随着编码递归次数 ι 增加，按照注意力矩阵衡量，语料库中足够局部的 n-gram 之间的耦合会减弱。随着字节对编码应用次数增加，注意力值分布的均值越来越接近 0。此外，分布的方差也会变小，这意味着平均而言，与之“相关”的相邻词元数量也会减少。

[11]在本研究中，所有变 换器都经过了“充分训练”，即损失已收敛 等。

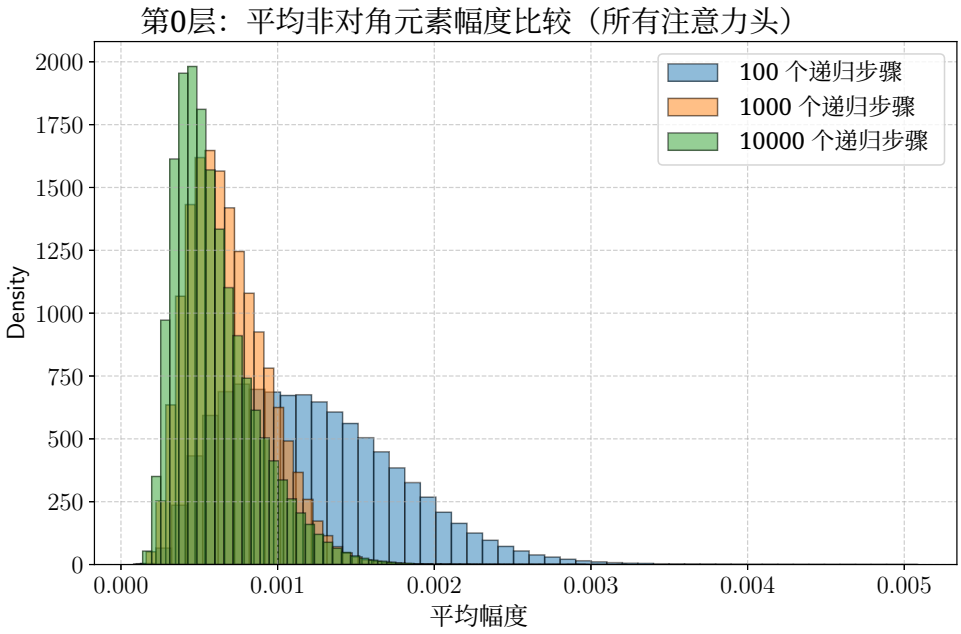


图 7：注意力矩阵的局部归一化非对角元素的幅度。

5 结论

在本研究中，我们考察了经过字节对编码处理的语料库的统计结构，以及其与齐夫分布行为出现之间的关系，并特别关注由此产生的词元分布所对应的熵。我们从对严格遵循齐夫频率定律时香农熵的理论分析入手，推导出平均槽位熵的显式表达式，该表达式仅取决于词汇表的大小和频率最高的词元。这为词元分布足够接近该定律的任何语料库的信息含量，确立了明确的定量预测。

通过实证分析，我们证明，随着 BPE 应用次数的增加，BPE的递归应用会持续推动不同语料粒度下的词元频率趋近齐夫定律。随着合并次数增加，拟合得到的齐夫指数趋近于 1，经验熵则先急剧上升，随后进入边际效益递减阶段。这一现象反映了词汇表不断扩展，纳入越来越细粒度但也越来越罕见的子词单元。由此得到的熵曲线与齐夫极限下推导出的理论预期高度吻合；在合并深度较小时出现的偏差，则可归因于频率结构不符合齐夫分布，以及分词早期阶段的 IID 假设不成立。

通过将分析扩展到在相同语料库上训练的 Transformer 模型，我们进一步观测到，模型 softmax 输出的熵特性与底层分词的齐夫结构相呼应。使用更大词汇表训练的模型——对应于更大的 BPE 递归深度——其预测熵与齐夫预测值更加吻合，这表明模型内化并重现了语料库的统计规律。补充的基于注意力的诊断结果表明，更深层次的分词会降低局部词元间相关性，使经验分布越来越符合理论熵计算所依赖的独立性假设。

综合来看，这些结果表明，字节对编码不仅将语言结构压缩为实用的子词词汇表，还会逐步重构自然语言的核心统计规律，这些规律体现于齐夫定律之中。BPE 下可预测的熵行为的出现，为理解分词选择如何塑造 Transformer 模型所面对的信息图景奠定了严谨的基础。更广泛地说，我们的发现凸显了熵作为统一工具的作用，可用于刻画语料库及其训练出的模型，并在经验语言结构与现代神经语言模型的概率机制之间架起桥梁。

最后，我们再次强调重整化与语言学分词之间的类比。用重整化群流的语言来说，字节对编码的渐近行为相当于语言学中流向红外的过程。在这一极限下呈现的齐夫定律统计规律，便构成了语言学红外普适类。

数据可用性声明

本项目的代码可在 GitHub 上通过 https://github.com/xand-stapleton/natural-language- tokenisation 获取。代码仓库中包含了复现结果所需的所有脚本、算法及资源链接。

致谢

DSB 和 AGS 感谢 Pierre Andurand 在本研究期间提供的支持；DSB 还获得科学技术设施委员会（STFC）综合拨款 ST/T000686/1“振幅、弦与对偶性”的支持。AGS 感谢 J. Kerrison 就分词与最先进模型中蛋白质结构编码之间关系的相关研究提出意见。

参考文献

[1] David S Berman、Marc S Klinger 和 Alexander G Stapleton。贝叶斯重整化。*MachineLearning: Science and Technology*，4(4):045011，2023年10月。[2] David S. Berman 和 Marc S. Klinger。精确重整化群流的逆作为统计推断。熵，26(5):389，2024年4月。[3] George Kingsley Zipf。语言相对频率原理选论。哈佛大学出版社，1932年12月。[4] J. B. Estoup。速记级数。法国速记学院，1916年。[5] Yanjin He、Qingkai Zeng 和 Meng Jiang。词元分布遵循齐夫定律时，预训练模型表现最佳，2025年。[6] Vladimir Berman。两阶段符号过程中的齐夫分布：随机词汇过滤下的稳定性，2025年。[7] Shuiyuan Yu、Chunshan Xu 和 Haitao Liu。50种语言中的齐夫定律：其结构模式、语言学解释和认知动机，2018年。[8] Vladimir Berman。齐夫定律的语素起源：因子化组合框架，2025年。[9] Mason Youngblood。鲸类交流中的类语言效率。*Science Advances*，11(6):eads6014，2025年。[10] Benoit Mandelbrot。信息论与心理语言学。*BB Wolman and E*，1965年。[11] W. Li。随机文本呈现类似齐夫定律的词频分布。*IEEE Transactions on Infor-mation Theory*，38(6):1842–1845，1992年。[12] Robert Edward Lewand。美国数学协会教材：密码学数学。美国数学协会，美国华盛顿特区，2000年12月。[13] Joni-Kristian K¨am¨ar¨ainen。Transformer 序列建模导论，2025年。[14] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。注意力机制就是你所需要的一切，2023年。[15] Richard E. Turner。Transformer 导论，2024年。[16] Saidul Islam、Hanae Elmekki、Ahmed Elsebai、Jamal Bentahar、Najat Drawel、Gaith Rjoub 和 Witold Pedrycz。Transformer 在深度学习任务中的应用综述，2023年。[17] DeepSeek-AI、Aixin Liu、Aoxue Mei、Bangcai Lin、Bing Xue、Bingxuan Wang、Bingzheng Xu、Bochao Wu、Bowei Zhang、Chaofan Lin、Chen Dong、Chengda Lu、Chenggang Zhao、Chengqi Deng、Chenhao Xu、Chong Ruan、Damai Dai、Daya Guo、Dejian Yang、Deli Chen、Erhang Li、Fangqi Zhou、Fangyun Lin、Fucong Dai、Guangbo Hao、Guanting Chen、Guowei Li、H. Zhang、Hanwei Xu、Hao Li、Haofen Liang、Haoran Wei、Haowei Zhang、Haowen Luo、Haozhe Ji、Honghui Ding、Hongxuan Tang、Huanqi Cao、Huazuo Gao、Hui Qu、Hui Zeng、Jialiang Huang、Jiashi Li、Jiaxin Xu、Jiewen Hu、Jingchang Chen、Jingting Xiang、Jingyang Yuan、Jingyuan Cheng、Jinhua Zhu、Jun Ran、Junguang Jiang、Junjie Qiu、Junlong Li、Junxiao Song、Kai Dong、Kaige Gao、Kang Guan、Kexin Huang、Kexing Zhou、Kezhao Huang、Kuai Yu、Lean Wang、Lecong Zhang、Lei Wang、Liang Zhao、Liangsheng Yin、Lihua Guo、Lingxiao Luo、Linwang Ma、Litong Wang、Liyue Zhang、M. S. Di、M. Y Xu、Mingchuan Zhang、Minghua Zhang、Minghui Tang、Mingxu Zhou、Panpan Huang、Peixin Cong、Peiyi Wang、Qiancheng Wang、Qihao Zhu、Qingyang Li、Qinyu Chen、Qiushi Du、Ruiling Xu、Ruiqi Ge、Ruisong Zhang、Ruizhe Pan、Runji Wang、Runqiu Yin、Runxin Xu、Ruomeng Shen、Ruoyu Zhang、S. H. Liu、Shanghao Lu、Shangyan Zhou、Shanhuang Chen、Shaofei Cai、Shaoyuan Chen、Shengding Hu、Shengyu Liu、Shiqiang Hu、Shirong Ma、Shiyu Wang、Shuiping Yu、Shunfeng Zhou、Shuting Pan、Songyang Zhou、Tao Ni、Tao Yun、Tian Pei、

Tian Ye、Tianyuan Yue、Wangding Zeng、Wen Liu、Wenfeng Liang、Wenjie Pang、Wenjing Luo、Wenjun Gao、Wentao Zhang、Xi Gao、Xiangwen Wang、Xiao Bi、Xiaodong Liu、Xiaohan Wang、Xiaokang Chen、Xiaokang Zhang、Xiaotao Nie、Xin Cheng、Xin Liu、Xin Xie、Xingchao Liu、Xingkai Yu、Xingyou Li、Xinyu Yang、Xinyuan Li、Xu Chen、Xuecheng Su、Xuehai Pan、Xuheng Lin、Xuwei Fu、Y. Q. Wang、Yang Zhang、Yanhong Xu、Yanru Ma、Yao Li、Yao Li、Yao Zhao、Yaofeng Sun、Yaohui Wang、Yi Qian、Yi Yu、Yichao Zhang、Yifan Ding、Yifan Shi、Yiliang Xiong、Ying He、Ying Zhou、Yinmin Zhong、Yishi Piao、Yisong Wang、Yixiao Chen、Yixuan Tan、Yixuan Wei、Yiyang Ma、Yiyuan Liu、Yonglun Yang、Yongqiang Guo、Yongtong Wu、Yu Wu、Yuan Cheng、Yuan Ou、Yuanfan Xu、Yuduan Wang、Yue Gong、Yuhan Wu、Yuheng Zou、Yukun Li、Yunfan Xiong、Yuxiang Luo、Yuxiang You、Yuxuan Liu、Yuyang Zhou、Z. F. Wu、Z. Z. Ren、Zehua Zhao、Zehui Ren、Zhangli Sha、Zhe Fu、Zhean Xu、Zhenda Xie、Zhengyan Zhang、Zhewen Hao、Zhibin Gou、Zhicheng Ma、Zhigang Yan、Zhihong Shao、Zhixian Huang、Zhiyu Wu、Zhuoshu Li、Zhuping Zhang、Zian Xu、Zihao Wang、Zihui Gu、Zijia Zhu、Zilin Li、Zipeng Zhang、Ziwei Xie、Ziyi Gao、Zizheng Pan、Zongqing Yao、Bei Feng、Hui Li、J. L. Cai、Jiaqi Ni、Lei Xu、Meng Li、Ning Tian、R. J. Chen、R. L. Jin、S. S. Li、Shuang Zhou、Tianyu Sun、X. Q. Li、Xiangyue Jin、Xiaojin Shen、Xiaosha Chen、Xinnan Song、Xinyi Zhou、Y. X. Zhu、Yanping Huang、Yaohui Li、Yi Zheng、Yuchen Zhu、Yunxian Ma、Zhen Huang、Zhipeng Xu、Zhongyu Zhang、Dongjie Ji、Jian Liang、Jianzhong Guo、Jin Chen、Leyi Xia、Miaojun Wang、Mingming Li、Peng Zhang、Ruyi Chen、Shangmian Sun、Shaoqing Wu、Shengfeng Ye、T. Wang、W. L. Xiao、Wei An、Xianzu Wang、Xiaowen Sun、Xiaoxiang Wang、Ying Tang、Yukun Zha、Zekai Zhang、Zhe Ju、Zhen Zhang 和 Zihua Qu。《Deepseek-v3.2：推动开放式大语言模型的前沿发展》，2025。

[18] Partha Ray。《ChatGPT：背景、应用、关键挑战、偏见、伦理、局限性与未来前景综合综述》。物联网与信息物理系统，3，2023 年 4 月。

[19] Hugging Face。《Hugging Face 课程》，2022 年。https://huggingface.co/course，2022 年。 [在线；访问日期 ¡today¿]。

[20] Philip Gage。《一种新的数据压缩算法》。《*C* 用户杂志》，12(2):23–38，1994 年。

[21] R Harald Baayen。《词频分布》，第 18 卷。Springer Science & Business Media，2001 年。

[22] Jonas F. Lotz、Ant´onio V. Lopes、Stephan Peitz、Hendra Setiawan 和 Leonardo Emili。《超越文本压缩：跨尺度评估分词器》，2025 年。

[23] Jack Bandy 和 Nicholas Vincent。《解决机器学习研究中的“文档债务”：针对 BookCorpus 的回顾性数据表》，2021 年。

[24] HF 官方模型维护者。GPT2（修订版 909a290），2022 年。

[25] Aditya Pal、Abhilash Barigidad 和 Abhijit Mustafi。IMDB电影评论数据集，2020 年。

[26]Tom Mitchell。Twenty Newsgroups。UCI 机器学习资料库，1997 年。DOI：https://doi.org/10.24432/C5C323。

[27] Antonio Gulli。AG 新闻文章语料库，2023 年。访问日期：2025-12-17。

[28] Tiago Almeida 和 Jos Hidalgo。《短信垃圾信息集》。UCI 机器学习资料库，2011。DOI：https://doi.org/10.24432/C5CC84。

transformertask 的描述

本研究使用了 HuggingFace 的 GPT2LMHeadModel [24], , 该模型在 IMDB电影评论数据集上经过微调，用于情感感知文本生成。该模型基于 OpenAI 的 GPT-2 架构，采用多个 Transformer解码器层，每层都包含多头自注意力和前馈网络。表 1 总结了此模型变体在 10k 递归步分词中的关键架构参数。

本研究采用了该模型的预训练和微调版本。每个 Transformer块都包含一个多头自注意力机制，使模型能够捕捉词元之间的长程依赖，并包含一个用于处理这些表示的前馈网络。我们将上下文窗口限制为 128 个词元，以兼顾计算效率，并为句子级和段落级评论补全提供充足的局部上下文。表 1 列出了用于大词汇表的模型技术规格；对于较小的词汇表，参数均相同，仅嵌入层按比例缩放。

组件	规格
模型	GPT2LMHeadModel
Transformer 主干	GPT2Model
词元嵌入（wte）	嵌入（10115×256）
位置嵌入（wpe）	嵌入（128×256）
嵌入层随机失活	Dropout（$p = 0.1$）
Transformer 模块数量	$6\times$ GPT2Block
层归一化（ln 1）	层归一化（256，$\epsilon = 10^{-5}$）
自注意力模块	GPT2Attention
注意力投影（c attn）	一维卷积（$768 \to 256$）
输出投影（c proj）	一维卷积（$256 \to 256$）
注意力随机失活	Dropout（$p = 0.1$）
残差丢弃	Dropout（$p = 0.1$）
层归一化（ln 2）	层归一化（256，$\epsilon = 10^{-5}$）
前馈网络	GPT2MLP
全连接层（c fc）	一维卷积（$1024 \to 256$）
投影层（c proj）	一维卷积（$256 \to 1024$）
激活函数	NewGELU
多层感知机 dropout	Dropout（$p = 0.1$）
最终层归一化（ln f）	层归一化（256，$\epsilon = 10^{-5}$）
语言模型头	线性（$256 \to 10115$）

Ta表 1：本研究所用 GPT2LMHeadModel 的架构和超参数 .

B 数据集与上下文依赖

本说明中所包含的所有实验，均使用四种广泛采用的文本分类基准数据集之一。

• *20* 新闻组 [26] 数据集包含约 20,000 篇文档，均匀分布在 20 个主题新闻组中，涵盖政治、体育、宗教和科学等主题。20Newsgroups 常用于文档分类和上下文分析任务。

• *IMDB* 电影评论 数据集 [25]包含约 50,000 条带有二元情感标签（正面或负面）的英语电影评论，训练集和测试集各占一半。在本研究中，尽管对 GPT2 模型进行了训练和评估，我们仍将测试集和训练集数据集合并为一个统一语料库。该数据集是情感分析的标准基准，侧重于篇幅较长且观点鲜明的文本。

• *AG* 新闻 数据集 [27] AGNews 是一个大规模新闻分类语料库，包含超过 120,000 篇新闻文章，分为四个类别：世界、体育、商业和科学/技术。它经常用于评估短篇至中等长度文本上的主题分类模型。

• 短信垃圾信息集数据集 [28] 包含约 5500 条短信，标注为垃圾信息或正常信息（非垃圾信息）。由于短信长度较短且类别不平衡，该数据集常用于研究二元文本分类以及模型对含噪非正式语言的鲁棒性。

表 2 还列出了一系列关键数据集特征。

表 2：实验所用文本分类数据集的摘要统计。

数据集	**#样本**	**#类别**	**平均长度**	**Task**
20 新闻组	∼20,000	20	∼250 个词	主题分类
IMDB 影评	50,000	2	∼230 个单词	情感分析
AG 新闻	120,000	4	∼40 个单词	新闻分类
短信垃圾信息	5,574	2	∼15 个词	垃圾信息检测