实验报告：中英文本信息熵与齐夫定律分析

1. 实验目的

本次实验的核心目标是加深对信息论中“熵”这一核心概念以及齐夫定律（Zipf's Law）的理解，并探究其在自然语言处理中的实际应用。具体目的如下：

- **数据采集与处理**：利用网络爬虫工具，从互联网上分别采集足量的英文和中文文本样本，并进行深度清洗，去除标点、数字及其他噪声。
- **多维度熵计算**：设计并编程实现算法，计算所采集样本的熵。

  a) 对于**中文语料**，分别计算**汉字**的概率和信息熵，并借助 jieba 分词工具，统计并计算**汉语词汇**的概率和信息熵。
  b) 对于**英文语料**，分别计算**英文字母**的概率和信息熵，以及**英文单词**的概率和信息熵。

- **齐夫定律验证**：利用收集的英文文本数据，通过绘制词频-排行对数图来验证齐夫定律。
- **规模效应分析**：通过逐步扩大文本规模（从 100 万至 600 万字符），重新计算上述指标，进行对比分析，探究熵值随样本规模的变化规律。

实验项目中所涉及到的技术细节和代码已上传至 GitHub 仓库，链接地址为 https://github.com/originality666/nlp_homework/tree/main/frequency_entropy_ziplaw_task。

2. 实验原理

- **信息熵 (Shannon Entropy)**

信息熵是信息论的基石，用于度量一个随机变量的不确定性程度。对于一个由离散字符组成的文本，其信息熵定义为每个字符的信息量的统计平均值。其计算公式如下：

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \cdot \log_b P(x)$$

其中，$\mathcal{X}$是随机变量 X（字符）的取值空间，$P(x)$是字符在文本中出现的概率。熵的单位通常是“比特/字符”（bits/char），表示每个字符平均携带的信息量。熵值越高，代表文本的随机性越强，字符分布越复杂。

- **齐夫定律 (Zipf's Law)**

齐夫定律是一个经验定律，它指出在一个足够大的自然语言语料库中，任意一个词的出现频率（Frequency）与其在频率表里的排名（Rank）成反比关系。即：

$$\text{Frequency} \propto \frac{1}{\text{Rank}}$$

在双对数坐标系（log-log plot）中，词频-排行曲线将呈现出一条斜率近似为-1 的直线。

3. 实验环境

（1）编程语言： Python 3

（2）核心库：

- requests：用于发送 HTTP 请求，获取网页的原始 HTML 内容。
- BeautifulSoup4：用于解析 HTML 文档，高效地提取纯文本内容。
- matplotlib：用于数据可视化，绘制文本规模与信息熵关系的曲线图。
- argparse：用于解析命令行参数，提高脚本的灵活性和可配置性。
- collection：用于统计字符频数。
- jieba: 用于中文文本的自动分词。
- regex: 用于实现更强大的文本正则清洗。

4. 实验步骤

整个实验流程主要分为两个阶段：数据采集和数据计算分析。

4.1. 样本采集与预处理

（1）数据来源

为保证样本的多样性，本次实验从多个网站采集数据。

- **中文样本：** 从 https://www.gutenberg.org/网站中采集的中国古典文学作品节选，如《红楼梦》、《三国志》、《西游记》、《狄公案》等共 16 部作品，附录中给出了详细的作品网址。
- **英文样本：** 从 https://www.gutenberg.org/网站中采集的外国古典文学作品节选，如 Pride and Prejudice、MOBY-DICK、Alice's Adventures in Wonderland、Frankenstein 等共 8 部作品，附录中给出了详细的作品网址。

（2）爬虫实现

实验使用 Python 编写了 scraper.py 脚本来自动化样本采集过程。该脚本首先

从一个预设的 URL 列表（chinese_seed_urls.json 和 english_seed_urls.json）读取待抓取的网址，网址在附录中给出。随后，它模拟浏览器行为发送 HTTP 和 GET 请求，获取网页内容。

（3）数据清洗

从网页直接获取的 HTML 源码包含了大量与文本内容无关的标签（如 <script>, <footer>）和样式信息。为提取干净的文本，clean_text 函数执行了以下清洗操作：

- 利用 BeautifulSoup 库解析 HTML 结构。
- 移除所有<script>, <style>, <footer>等非内容承载标签。
- 提取并拼接剩余标签中的文本内容。
- 使用正则表达式将文本中连续的空白符（包括空格、换行符、制表符）替换为单个空格，确保文本格式的规范化。
- 清洗后的文本被保存为本地.txt 文件，用于后续分析。
- 在核心分析脚本 analyzer.py 中，通过 clean_text 函数执行了更精细的文本清洗：
  a) 中文语料：移除所有非中文字符，包括拉丁字母、数字、标点符号和空格，仅保留汉字。
  b) 英文语料：移除所有非英文字母的字符，并将所有字母转换为小写，用单个空格分隔单词。

4.2. 熵计算与齐夫定律验证

分析与计算由 analyzer.py 脚本完成，核心算法步骤如下：

- **语料加载：** load_corpus 函数读取所有清洗后的.txt 样本文件，合并成一个单一的基础文本语料。
- **熵计算：**
  a) **字符/字母熵：** 将文本视为字符序列，利用 collections.Counter 统计每个字符（中文为汉字，英文为字母）的频率，并代入香农熵公式计算。
  b) **词熵：** 首先使用 segment_text 函数对文本进行分词（中文使用 jieba.cut，英文使用 text.split），然后以词为单位统计频率并计算熵。
- **齐夫定律验证：** plot_zipf 函数统计英文单词的词频和排名，并在对数坐标下绘制“排名-频率”散点图，以验证齐夫定律。
- **规模扩展与分析：** analyze 函数以 100 万字符为步长，从 100 万到 600 万不断增加样本规模，对每个规模的文本重复执行上述熵计算和分析过程，并将结果保存。

5. 实验结果与分析

5.1. 字符熵和词熵的计算结果

我分别对中英文本在 100 万至 600 万字符的不同规模下进行了字符熵和词熵的计算，结果如下：

表 1 中文语料熵计算结果 (数据来源: results_chinese.json)

文本规模 (字符)	独立字符数	香农熵 (bits/char)	独立词汇数	词熵 (bits/word)
1,000,000	5,004	9.2894	84,585	11.6846
2,000,000	5,675	9.4780	152,212	12.1915
3,000,000	6,024	9.6051	215,914	12.6255
4,000,000	6,356	9.5862	266,109	12.6759
5,000,000	7,120	9.7314	341,492	13.0667
6,000,000	7,512	9.8540	435,303	13.4417

表 2 英文语料熵计算结果 (数据来源: results_english.json)

文本规模 (字符)	独立字符数	香农熵 (bits/char)	独立词汇数	词熵 (bits/word)
1,000,000	27	4.0931	15,891	9.9174
2,000,000	28	4.0919	19,542	9.8460
3,000,000	28	4.0890	22,709	9.8512
4,000,000	28	4.0880	24,991	9.8427
5,000,000	28	4.0864	26,975	9.8471
6,000,000	28	4.0856	28,307	9.8273

5.2. 熵随文本规模的变化分析

为了直观展示熵的变化趋势，我绘制了熵与文本规模的关系曲线图。

- **中文分析:** 从图 1 可以看出，无论是汉字熵还是词熵，都随着文本规模的扩

大呈现出明显的、持续的增长趋势 。这完美揭示了汉语系统的复杂性 。随着样本量的增加，新的、频率较低的汉字和词汇不断被纳入统计，使得独立字符数和词汇数持续增加，概率分布不断调整，从而导致整体熵值稳步上升 。两条曲线均未出现平缓的迹象，说明即使在 600 万字符的规模下，语料库仍远未达到饱和状态 。

- **英文分析：**与中文不同，图 2 显示英文的字母熵在文本规模超过 100 万后就基本保持稳定（约 4.09 bits）。这是因为英文的字符集非常有限（26 个字母），很小的文本量就能覆盖所有字符，其频率分布也迅速趋于稳定 。英文的词熵也表现出相对稳定，仅有微小波动（围绕 9.85 bits），说明常用词汇的分布在达到一定样本量后也已成型 。

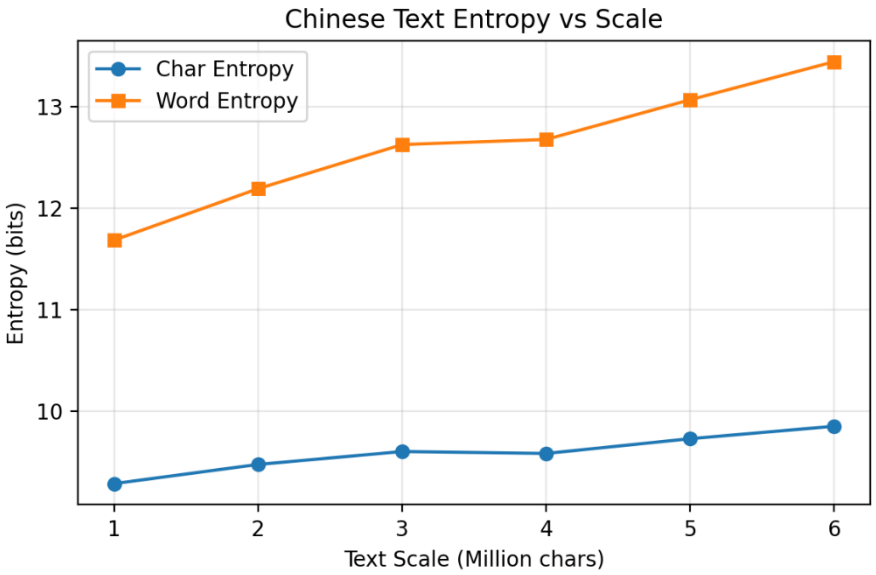

图 1 从 200 万至 600 万字符的不同规模下中文语料熵值统计结果

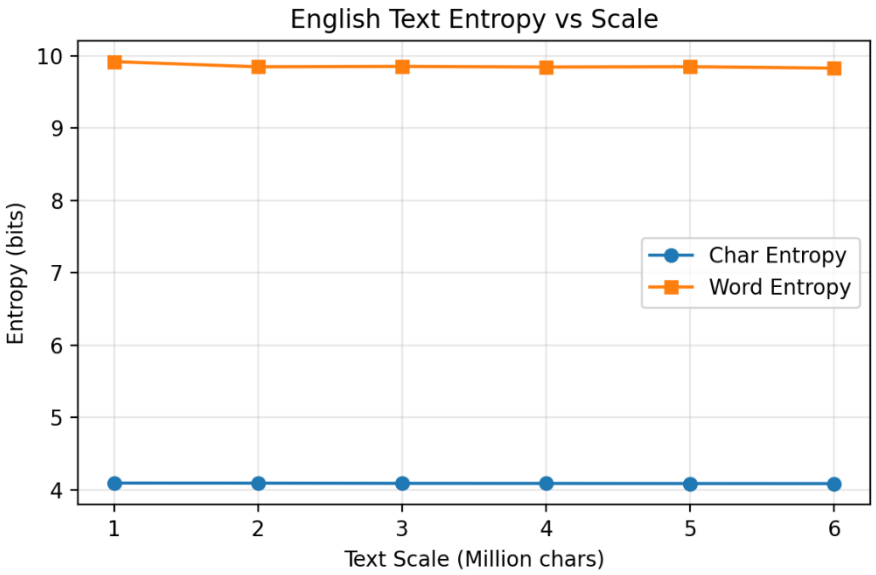

图 2 从 200 万至 600 万字符的不同规模下英文语料熵值统计结果

5.3. 齐夫定律验证 (英文)

我对约 600 万字符规模的英文语料进行了词频统计，并绘制了“排名-频率”双对数图。

如图 3 所示，数据点在双对数坐标系下清晰地呈现出一条线性递减的趋势，这与齐夫定律的理论预测高度吻合，成功验证了该定律在本次采集的英文语料上的有效性。

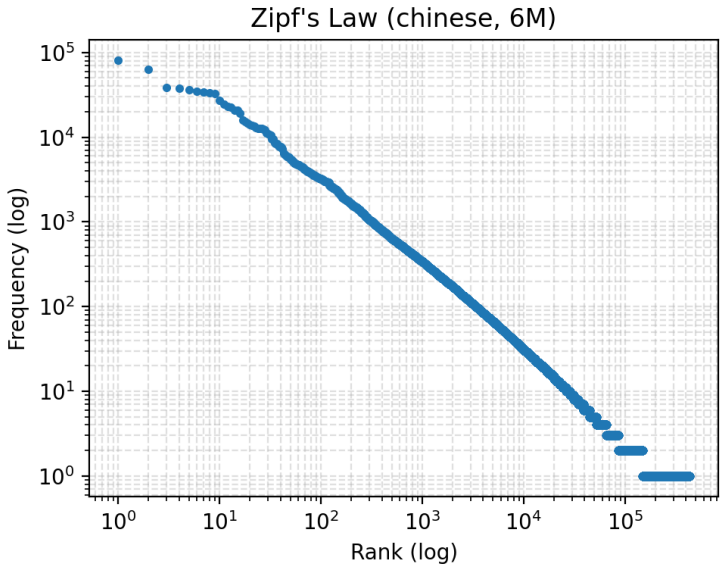

图 3 英文文本的齐夫定律验证 (6M 规模)

5.4. 语料特性对中文熵值的影响分析

- **汉字熵分析**：实验测得的汉字熵（最大约 9.85 bits）略高于理论参考值（约 9.71 bits ）。这可能是因为我们的样本全部来自古典文学，其用字范围和频率分布与更庞大、更多样化的现代通用语料库（理论值的计算基础）存在差异 。
- **中文词熵偏高分析**：本次实验计算出的中文词熵（在 600 万规模下达到 13.44 bits）与理论值 11.46 相比显著偏高。这与我们的语料来源密切相关。中国古典白话小说相较于现代汉语文本，具有以下特点：
  a) **词汇更丰富**：包含大量现代汉语中不常用的词汇、成语和表达方式。
  b) **用词更灵活**：作者倾向于使用多样化的词语来描绘场景和人物，重复率较低。
  c) **分词挑战**：jieba 分词库主要基于现代汉语语料训练，处理古典文本时可能会产生更多的分词错误或更细碎的切分结果，这两种情况都会导致独立“词汇”单元的数量（unique_words）虚增，从而推高了计算出的词熵。

6.关于“伸缩空间”的讨论

本次实验虽然达到了预期目标，但在方法和深度上仍有很大的“伸缩空间”，可从以下方面进行扩展：

- **样本的广度与均衡性**：未来可以扩展采集范围，覆盖新闻、科技、社交媒体等更多领域，并加入现代汉语语料进行对比，构建一个更具代表性的均衡语料库 。
- **分析单元的深化**：本次实验分析了字/字母和词两个层面 。可以向更复杂的层面扩展，例如计算 **N-gram**（二元或三元组）的条件熵，以更精确地度量语言的上下文相关性 。
- **针对性分词工具**：针对古典文献，可以尝试使用经过专门训练的或支持自定义词典的分词工具，以获得更准确的词切分结果，从而得到更可靠的词熵。

7. 结论

本实验成功地通过编程实践，完成了对中英文本在字符和词汇两个层面上的信息熵计算、齐夫定律验证及规模效应分析。实验结果清晰地表明：

- 无论是字符/字母熵还是词熵，中文的熵值均显著高于英文。这定量地证实了汉语系统在字符和词汇层面的复杂性和信息密度远超英文。
- 随着文本规模的增加，英文熵趋于稳定，而中文熵（包括字熵和词熵）持续增长，这反映了两种语言在字符集大小和词汇丰富度上的巨大差异。
- 语料的文体风格对熵值有显著影响。本实验采用的中国古典小说语料，因其词汇丰富且独特，导致了中文词熵的计算结果显著偏高。

通过本次实验，我不仅掌握了信息熵和齐夫定律的计算与验证方法，更深入地理解了样本选择、处理方法对实验结果的关键影响。

附录

本项目所提到的英文语料爬取网址如下：

- https://www.gutenberg.org/files/2701/2701-h/2701-h.htm
- https://www.gutenberg.org/files/1342/1342-h/1342-h.htm
- https://www.gutenberg.org/files/11/11-h/11-h.htm
- https://www.gutenberg.org/files/84/84-h/84-h.htm
- https://www.gutenberg.org/files/98/98-h/98-h.htm
- https://www.gutenberg.org/cache/epub/2759/pg2759.txt
- https://www.gutenberg.org/cache/epub/1184/pg1184.txt
- https://www.gutenberg.org/cache/epub/1257/pg1257.txt

本项目所提到的中文语料爬取网址如下：

- https://www.gutenberg.org/cache/epub/24264/pg24264.txt

- https://www.gutenberg.org/cache/epub/24185/pg24185.txt
- https://www.gutenberg.org/cache/epub/25146/pg25146.txt
- https://www.gutenberg.org/cache/epub/25606/pg25606.txt
- https://www.gutenberg.org/cache/epub/27582/pg27582.txt
- https://www.gutenberg.org/cache/epub/7367/pg7367.txt
- https://www.gutenberg.org/cache/epub/27686/pg27686.txt
- https://www.gutenberg.org/cache/epub/25142/pg25142.txt
- https://www.gutenberg.org/cache/epub/23910/pg23910.txt
- https://www.gutenberg.org/cache/epub/23962/pg23962.txt
- https://www.gutenberg.org/cache/epub/52200/pg52200.txt
- https://www.gutenberg.org/cache/epub/27582/pg27582.txt
- https://www.gutenberg.org/cache/epub/25393/pg25393.txt
- https://www.gutenberg.org/cache/epub/25327/pg25327.txt
- https://www.gutenberg.org/cache/epub/27166/pg27166.txt
- https://www.gutenberg.org/cache/epub/25134/pg25134.txt