Report of Deep Learning for Natural Langauge Processing

Tuo Zhao
975352925@qq.com

Abstract

本报告对包含 16 部小说的中文语料库进行了语料分析，在此基础上验证了齐夫定律，分别以字和词为单位，进行了三元语言模型的信息熵计算。

Introduction

齐夫定律是一种在自然语言处理和信息论中常见的经验规律。它是由美国语言学家乔治·齐夫提出的，描述了一种特定的统计现象——在任何一个自然语言的语料库中，一个词的频率与其频率排名成反比。具体来说，齐夫定律指出，一个语料库中最常见的词出现的频率大约是第二常见的词出现频率的两倍，是第三常见的词出现频率的三倍，以此类推。尽管齐夫定律本身是经验性的，而且在不同的数据集和条件下表现出一定的变化，但人们普遍认为它反映了语言使用中的一种优化策略：使用少量的词频繁地表达大多数想法，同时保留大量的词来表达特定或少见的概念，以此实现沟通的效率和精确性之间的平衡。本报告将 16 部小说的文本进行合并，进行齐夫定律的验证。

信息熵的概念最初由克劳德·香农在 1948 年提出，定义为一个随机变量不确定性的度量。对于信息熵直观来看：当所有可能的事件都具有相同的概率时，信息熵达到最大值。也就是说，当一个系统完全随机时，其不确定性最大。如果一个事件的发生是确定的（即，某个事件的概率为 1，其他为 0），则信息熵为 0。也就是说，当一个系统的结果是确定无疑的，那么它的不确定性是最小的，因此信息熵也是最小的。

三元语言模型（Trigram Model）与一元（Unigram Model）和二元语言模型（Bigram Model）相比，在处理自然语言处理（NLP）任务时通常更具优势。三元模型拥有更好的上下午捕捉能力，更准确的语言模式建模。本报告将 16 部小说的文本进行合并，按照一/二/三元语言模型计算了字/词的信息熵，并比较了不同模型信息熵的不同，对实验结果进行了分析。

Methodology

Task1：在中午语料库验证齐夫定律

具体来说，验证齐夫定律有以下步骤：

1. 准备语料库：选择一个中文文本作为实验的语料库。本报告中，将 16 部金庸小说进行合并得到完整文本。
2. 文本预处理：对语料库进行预处理。删除所有的隐藏符号、非中文字符和标点符号。由于中文是由连续的字符组成的，没有像英文一般的空隔，所以我们需要进行分词。实验中，我们选择的是 Jieba 库来进行分词。Jieba 是中文自然语言处理中广泛使用的分词工具。Jieba 支持三种分词模式：精确模式、全模式、搜索引擎模式。实验中所使用的是精确模式，将句子最精确地切开，适合文本分析。

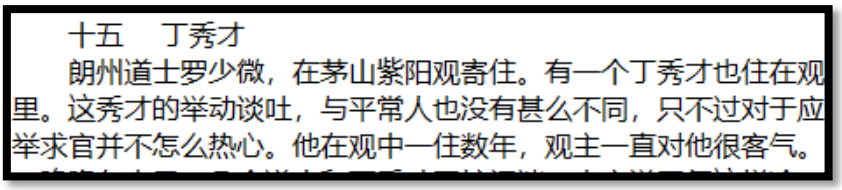
十五　丁秀才
朗州道士罗少微，在茅山紫阳观寄住。有一个丁秀才也住在观里。这秀才的举动谈吐，与平常人也没有甚么不同，只不过对于应举求官并不怎么热心。他在观中一住数年，观主一直对他很客气。

图 1　txt 文本中存在较多无关字符（换行符等）

3. 去除停用词:去除如“的”、“了”等高频但不携带具体信息的词。实验中使用了课后老师下发的中文停词库。
4. 计算词频：统计每个词出现的次数。
5. 绘制齐夫定律图：按照词频进行降序排序，然后每个词的排名（横轴）对其频率（纵轴）的对数图。并使用 numpy 库中的‘poly1d’函数生成拟合曲线，计算其斜率，最后验证齐夫定律。

Task2：计算信息熵

对于一个离散随机变量X，其概率分布P(X)，信息熵H(X)定义为：

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p\,(x_i)$$

其中p(xi)是随机变量X取特定值xi的概率，n是X的可能取值的总数，logb是对数函数，底数b通常取2（此时单位是比特），也可以取自然数e（此时单位是奈特）或者10（此时单位是哈特）。实验中取比特作为单位。

下面介绍一元语言模型、二元语言模型、三元语言模型。

Model1：一元语言模型

一元语言模型是最简单的语言模型。一元语言模型基于假设：每个词出现的概率只与它本身有关，而与它的前后文无关。具体来说：一元语言模型认为句子中每个词出现的概率是独立的。信息熵计算公式如下：

$$H = -\sum_{i=1}^{V} p(w_i) \log_2 p\,(w_i)$$

p(wi)是该词在语料库中出现的概率。V是词汇表的大小，wi是词汇表中的第i个词

这种模型的优点是易于理解和实现，但它忽略了词与词之间的关系，因此在实际应用中的表现可能不是很理想。

Model2：二元语言模型

二元语言模型考虑词与词之间的关系，即一个词出现的概率不仅与它本身有关，还与它前面的一个词有关。二元模型的信息熵计算需要考虑词与其前一个词的关系。它可以表示为条件信息熵的平均值：

$$H = -\sum_{i=1}^{V}\sum_{j=1}^{V} p(w_i, w_j) \log_2 p\left(w_j \mid w_i\right)$$

这里，p(wi,wj)是词wi后面紧跟词wj出现的联合概率，而p(wj | wi)是给定前一个词wi的情况下，词wj出现的条件概率。

Model3：三元语言模型

三元模型的信息熵计算进一步扩展了二元模型，考虑了一个词与其前两个词的关系。其信息熵可以表示为：

$$H = -\sum_{i=1}^{V}\sum_{j=1}^{V}\sum_{k=1}^{V} p(w_i, w_j, w_k)\log_2 p\left(w_k \mid w_i, w_j\right)$$

这里， p(wi,wj,wk)是序列wi,wj,wk出现的联合概率，而p(wk｜wi,wj)是在给定前两个词wi和wj的情况下，词wk出现的条件概率。

Experimental Studies

Task1：在中午语料库验证齐夫定律

对16部小说的文本合并得到的完整文本进行分词统计后，得到齐夫定律图如图2所示：

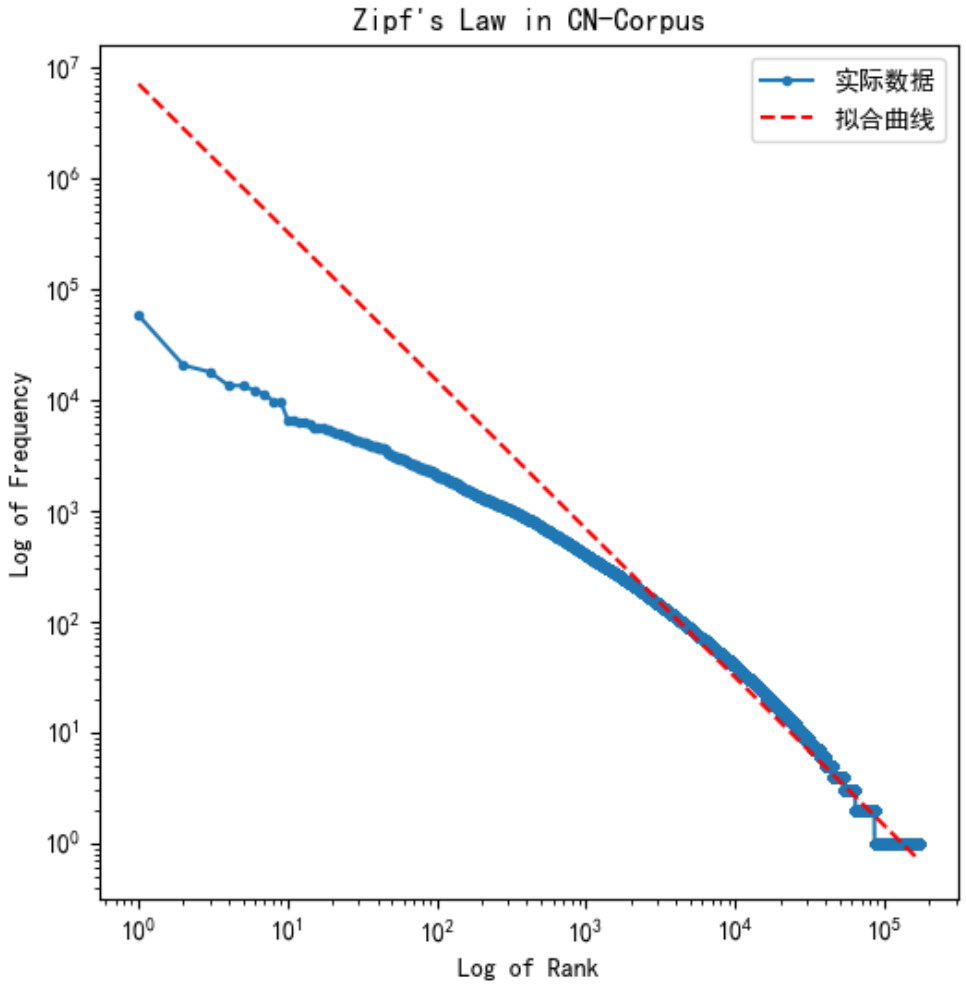

图 2 完整文本词频统计

图中拟合直线的斜率近似为**-1.3**，与理论上的**-1**有一定出入。实际应用中，不同的文本或语料库的词频分布可能会有所不同。文本的类型、语言风格、话题或作者的写作习惯等因素都可能影响词频分布，从而导致斜率与理论值存在偏差。

取其中的前18000个词，得到齐夫定律图如图3所示：

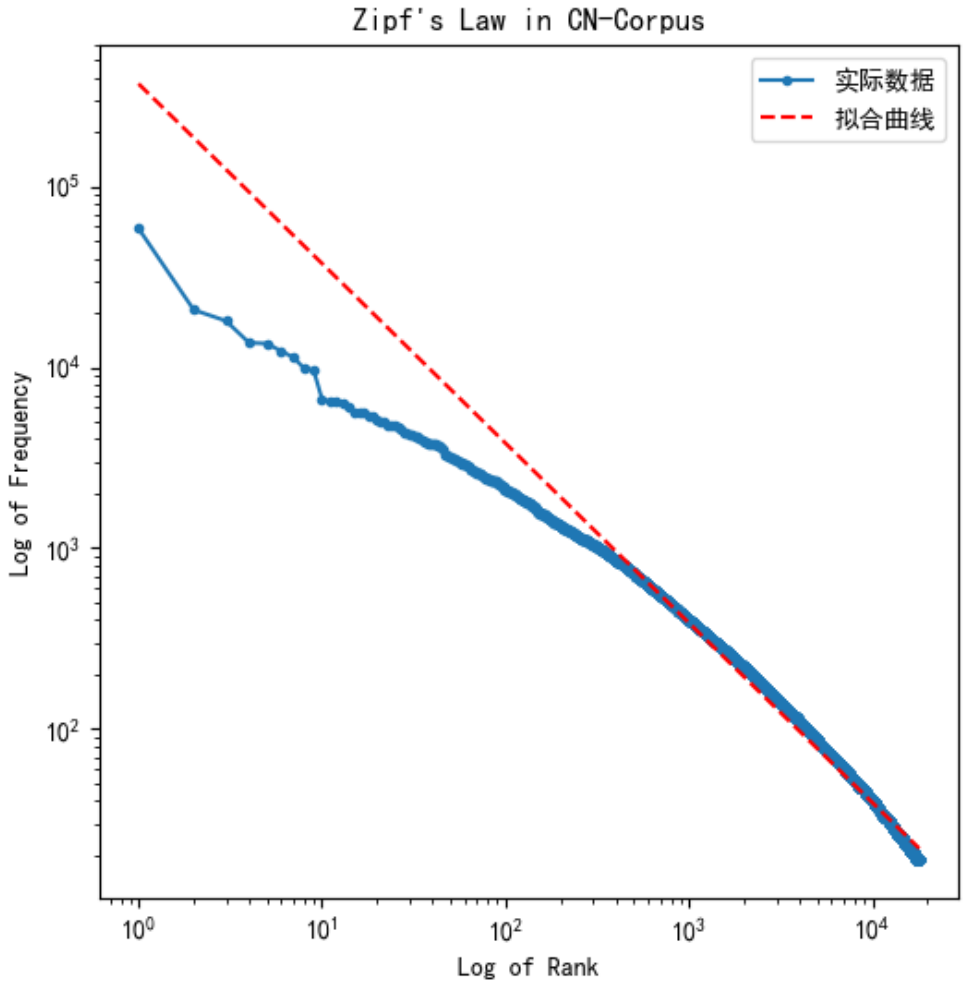

图 3 完整文本（前 18000 词）词频统计

图中拟合直线的斜率近似为**-0.99**，与理论上的**-1** 基本相同。在很多实际应用中，低频词的出现可能由于偶然性较大，不一定遵循幂律分布，这些词的随机性可能扭曲了整体的斜率测量。通过只分析前 18000 个词，可能有效地排除了那些低频但高变异性的词汇，从而使得拟合曲线更接近理想的幂律分布。同时，高频词通常在语言中有稳定的使用模式，这使得它们的分布可能更符合齐夫定律的理论预测。因此，分析时集中在这些词上，可能会看到斜率更接近-1。

Task2：计算信息熵

对 16 部小说的文本合并得到的完整文本进行一元/二元/三元字和词的信息熵计算后，得到结果如表 1 所示：

	一元语言模型	二元语言模型	三元语言模型
字信息熵	9.94005	**7.03605**	**3.50255**
词信息熵	**13.64119**	6.50428	1.15032

表 1 各语言模型字/词信息熵

根据实验结果我们可以发现：随着元数（即上下文的大小，如一元、二元、三元）的增加，字或者词信息熵通常会减少。当模型从一元模型过渡到二元和三元模型时，所考虑的上下文长度增加。例如，在一元模型中，每个词的出现是独立的，而在二元模型中，一个词的出现依赖于前一个词，在三元模型中，则依赖于前两个词。随着依赖的上下文增多，模型对每个单词出现的条件有了更准确的估计。随着元数的增加，模型能够更准确地预测下一个词的出现，从而降低了整体的信息熵。

同时我们可以观察到。一元语言模型中字信息熵小于词信息熵。而二元与三元语言模型反之。在中文中，尽管字的总数较多，实际使用频繁的字的数量相对有限，而词的组合性和多样性更大。更高的词汇多样性意味着更高的不确定性，因此一元语言模型词级别的信息熵高于字级别。而在二元与三元语言模型中，在字级别，尽管单个字的多样性不及词，但字的组合产生的二元和三元序列数量庞大，且分布相对均匀。这意味着在字级别模型中，预测下一个字的不确定性相对较高。词作为语言的基本单位，承载了更丰富的语义和语法信息，特别是在二元和三元模型中，词与词之间的搭配更能反映固定的语言习惯和规律，因此其条件概率分布相对集中。例如，“不可思议”这样的词组在实际语言使用中固定性较高，其预测不确定性较低。

Conclusion

本报告验证了齐夫定律，分析了不同的条件下，拟合直线斜率变化的原因。同时利用一元二元三元语言模型分别计算各模型的字词信息熵，分析了不同语言模型信息熵的差异。

Reference

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C., & Mercer, R. L. (1992). An estimate of an upper bound for the entropy of English. *Computational Linguistics*, *18*(1), 31-40.