深度学习与自然语言处理第一次作业

ZY2314222 魏智兴

Abstract:

这作业分为两个部分：第一部分验证了中文语境下的 Zipf's Law，第二部分则计算了中文语料库的平均信息熵，包括字和词两个单位。针对第一部分，使用 Python 和 jieba 库对提供的中文语料库进行了分词，并通过 Python 的 map 函数统计了词频，以验证了 Zipf's Law。对于第二部分，则利用 Python 和 jieba 库对语料库进行了相同的分词处理，然后依次利用 Unigram、Bigram 和 Trigram 统计语言模型来计算中文的平均信息熵，这有助于更深入地理解自然语言处理以及统计语言模型的原理。

Introduction

（1）Zipf's Law

Zipf's Law 是指为：对某种语料库的单词频率统计中，单词频率与单词频率排名(降序)存在反比关系。Zipf's Law 表明，最常见的词汇出现频率通常为第二常见词汇频率的 2 倍，也是第三常见词汇频率的 3 倍。如在 Brown 语料库的美国英语文本中，“the”是最高频的词汇，频率约为 7%，而排在第二位的词汇“of”的出现频率则恰好约为 3.5%。

（2）信息熵

信息熵的概念最早由香农（1916-2001）于 1948 年借鉴热力学中的“热熵”的概念提出，旨在表示信息的不确定性。熵值越大，则信息的不确定程度越大。其数学公式可以表示为：

$$H(x) = \sum_{x \in X} P(x) log(\frac{1}{P(x)}) = - \sum_{x \in X} P(x) log(P(x))$$

（3）N-gram 模型

N-gram 是一种输入为一个句子，输出为该句子的概率的语言模型。N-gram 模型基于“第 N 个词的出现只与前面 N-1 个词相关，而与其它词都无关”的假设。假设 S 表示某个有意义的句子，其由一连串特定顺序排列的词ω1,w2,.., wn 组成，n 为句子长度。则 S 在文本中出现的可能性为:

$$p(s) = p(\omega_1, \omega_2, \dots, \omega_n)$$

利用条件概率公式得到:

$$p(\omega_1, \omega_2, \dots, \omega_n) = p(\omega_1)p(\omega_1|\omega_2)\dots p(\omega_n|\omega_1, \omega_2, \dots, \omega_n)$$

当 N=1 时，就是一元模型，N=2 就是二元模型，以此类推。N 元模型的数学表达如下所示:

$$P(\omega_i|\omega_{i-n+1}^{i-1}) = P(\omega_i|\omega_{i-n+1}, \dots, \omega_{i-1})$$

一元、二元和三元模型分别表示的含义为:N=1 表示与前面单词都没有关系; N=2 表示与前面一个单词有关;N=3 表示与前面两个词有关。

Methodology

M1 ： 验证步骤

我们将使用中文语料库验证 Zipf's Law，具体步骤如下:

（1）将语料库输入系统，然后对语料库中的词语进行分词，并记录每个词语的出现次数。

（2）将词语按照出现次数从高到低进行排列，利用对数坐标轴绘制词语排名与出现频率之间的关系图，其中横坐标表示词语的排名（取对数），纵坐标表示词语的频率（取对数）。

（3）如果图中的数据点基本位于一条直线上，并且符合对数关系，那么就验证了 Zipf's Law。

M2 ： 实验步骤

（1）遍历文件夹中的语料，并逐个读取其中的 txt 文件进行合并。

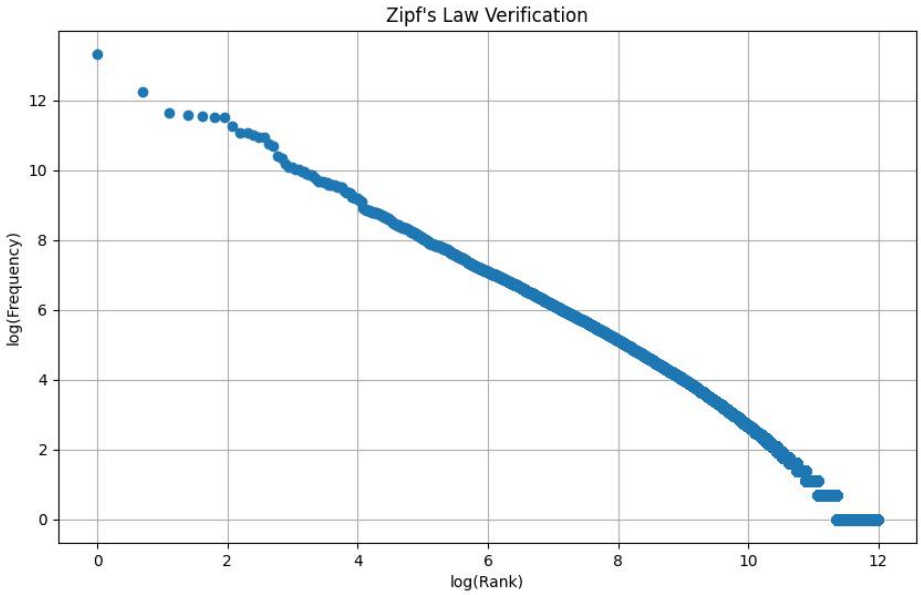
（2）对文件进行预处理，去除非中文字符以及与小说内容无关的部分，最终得到字符串形式的语料库。

（3）使用两种不同的模式（分词和分字符）生成词频字典。在分词模式下，使用 jieba 库的分词函数对原始语料进行处理；在分字符模式下，直接使用原始语料。

（4）利用一元、二元和三元模型的相关公式生成词频表。

（5）基于词频表的概率，利用相关公式计算文件的信息熵，分别以字和词两种形式进行计算。

Experimental Studies



如图所示，频率和词语的排名对应的数值点大致落在一条直线上，即可验证 Zipf's Law。

Experimental Studies

通过 python 的计算，获得了以字和词为单位，在一元、二元、三元模型下 N-gram 模型的平均信息熵

```
基于字的1元模型的平均信息熵为： 9.527281370040797
基于词的1元模型的平均信息熵为： 12.18045555087801
基于字的2元模型的平均信息熵为： 6.718080732511839
基于词的2元模型的平均信息熵为： 6.952339353610784
基于字的3元模型的平均信息熵为： 3.95098779271481
基于词的3元模型的平均信息熵为： 2.3054609809305355
```

通过对同一语料库下不同 N 值的 N-gram 模型 分析结果的对比，可以观

察到随着 N 值的增大，平均信息熵逐渐减小。这意味着随着序列长度的增加，信息的不确定性减小，特定位置的出现更具确定性。换句话说，在已知上下文的情况下，较长的序列使得对某处词组的出现更容易预测。

Conclusions

综上所述，此作业使用 Python 及其提供的 jieba 库对提供的中文语料库进行了分词和词频统计，以验证中文语境下的 Zipf's Law。同时，通过分词和词频统计，利用 1 元，2 元，3 元 统计语言模型计算了中文的平均信息熵，从汉字和词语两个角度进行了分析。这一过程加深了对自然语言处理基本概念以及统计语言模型的理解。