

# Report of Deep Learning for Natural Language Processing

Yucheng Wang

wang\_eil@126.com

## Abstract

本文对金庸的 16 篇小说进行了语料分析，验证了齐夫定律，分别按照字和词为单位，从一元、二元、三元字\词计算了中文平均信息熵。

## Introduction

齐夫定律（Zipf's law）描述了自然语言中单词出现频率的规律：在自然语言的语料库里，一个单词出现的频率与它在频率表里的排名成反比。齐夫定律是一个实验定律，满足齐夫定律的语料在单词排名-频率对数图像（横坐标取单词排名的对数，纵坐标取频率的对数）中，将呈现出近似的直线。本文将对 16 篇小说的文字进行统计分析。

信息是个很抽象的概念，人们很难说清楚信息到底有多少。香农在《通信的数据理论》中指出任何信息都存在冗余，冗余大小与信息中每个符号（数字、字母或单词）的出现概率有关。香农借鉴了热力学中熵的概念，将信息排除了冗余后的平均信息量称为“信息熵”。在自然语言处理中，信息熵只反映内容的随机性（不确定性）和编码情况，与内容本身无关。信息熵越大，单个词提供的信息量也就越大，不确定性也就越大。通过计算信息熵，能够衡量词表意的精确程度，信息熵越小，表意越精确。

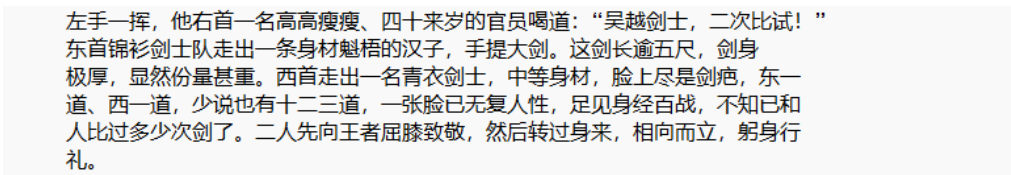
本文首先对金庸的 16 篇小说进行词频统计并验证齐夫定律，接着按照一元、二元、三元语言模型分别计算了字\词的信息熵，最后对实验结果进行了比较分析。

## Methodology

### Part1: 验证齐夫定律

验证齐夫定律需要对文本中的词和词频进行统计。具体方法如下：首先对文本进行预处理，删除所有的隐藏符号、非中文字符和标点符号；接着使用 jieba 库，对每个 txt 文件中的文本进行分词；然后统计每个词出现的次数并绘图观察。

在 txt 文件中，在正文之中夹杂着大量空格、换行符、标点符号等，如图 1 所示，这些会影响分词的结果。因此再进行分词之前，首先对文本进行预处理，只保留文本。



左手一挥，他右首一名高高瘦瘦、四十来岁的官员喝道：“吴越剑士，二次比试！”东首锦衫剑士队走出一条身材魁梧的汉子，手提大剑。这剑长逾五尺，剑身极厚，显然份量甚重。西首走出一名青衣剑士，中等身材，脸上尽是剑疤，东一道、西一道，少说也有十二三道，一张脸已无复人性，足见身经百战，不知已和人比过多少次剑了。二人先向王者屈膝致敬，然后转过身来，相向而立，躬身行礼。

图 1 txt 文本示例

Jieba 库是一款优秀的用于中文分词的库，它利用一个中文词库，确定汉字之间的关联概率，汉字间概率大的组成词组，形成分词结果。Jieba 库支持精确、全模式、搜索引擎三种分词模式。本文中对经过预处理的文本使用 Jieba 精确模式，它将一段文本切分为若干个中文单词，不会产生冗余，是最适合词频统计的模式。

在得到分词结果之后，对词频（词出现的次数）和词序（词出现次数的排序，出现多的排在前面）取对数画图。分别绘制了展示所有 txt 文件统计结果的总图和各个 txt 文件分图。

### Part2: 计算信息熵

对文本信息而言，若信息有  $n$  种单元  $x_1, x_2, \dots, x_n$ ，对应发生的概率为  $P(x_i)$ 。各种单元彼此独立，所以文本信息的平均不确定性应当为单个单元不确定性的统计平均值，称为信息熵。离散随机变量  $X$  的熵值  $H$  定义如下：

$$H(X) = E[I(X)] = E[-\ln(P(X))]$$

其中， $P$  为  $X$  的概率质量函数， $E$  为期望函数， $I(X)$  为  $X$  的信息量。当样本数量有限时，公式可以表示为：

$$H(X) = \sum_i P(x_i) I(x_i) = - \sum_i P(x_i) \log_b P(x_i)$$

其中， $b$  取 2 的时候，熵的单位是 bit； $b$  取自然常数  $e$  时，熵的单位是 nat； $b$  取 10 时，熵的单位是 Hart。本文中  $b$  取 2。

考虑语料中词与词之间的关系，可以分为一元语言模型、二元语言模型、三元语言模型、...、 $n$  元语言模型。本文只考虑至三元语言模型。

### Model1：一元语言模型

一元语言模型中，每个字\词出现的概率与其它字\词无关。此时，字\词信息熵的计算公式为：

$$H(X) = - \sum_{x \in X} P(x) \log_2 P(x)$$

其中  $P(x)$  可近似等于每个字或词在语料库中出现的频率

### Model2：二元语言模型

假设语料中的字\词出现具有马尔科夫性，其出现概率只与前一个字\词有关，此时为二元语言模型。当前字\词 $x$ 与前字\词 $y$ 组成了二元组 $(x, y)$ ，其信息熵计算公式为：

$$H(X|Y) = - \sum_{x \in X, y \in Y} P(x, y) \log_2 P(x|y)$$

其中联合概率 $P(x, y)$ 可近似等于每个二元词组在语料库中出现的频率，条件概率 $P(x|y)$ 可近似等于每个二元组在语料库中出现的频数与以该二元组的第一个词为词首的二元组出现的频数的比值。

### Model3：三元语言模型

在二元语言模型基础上，假设字\词出现的概率与前两个字\词有关，此时为三元语言模型，可以理解为单词 $z$ 正好出现在二元组 $(x, y)$ 之后，组成了三元组 $(x, y, z)$ ，其信息熵计算公式为：

$$H(X|Y, Z) = - \sum_{x \in X, y \in Y, z \in Z} P(x, y, z) \log_2 P(x|y, z)$$

其中联合概率 $P(x, y, z)$ 可近似等于每个三元词组在语料库中出现的频率，条件概率 $P(x|y, z)$ 可近似等于每个三元词组在语料库中出现的频数与以该三元词组的前两个词为词首的三元词组的频数的比值。

# Experimental Studies

## Part1: 验证齐夫定律

对 16 部作品分别进行分词统计之后，得到结果在图 1 中。

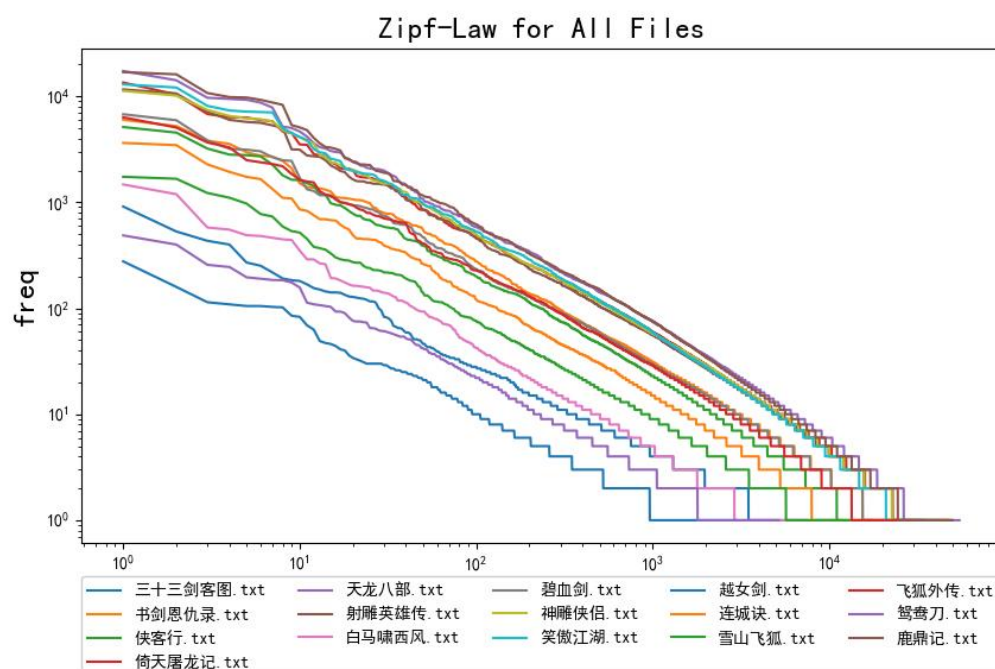
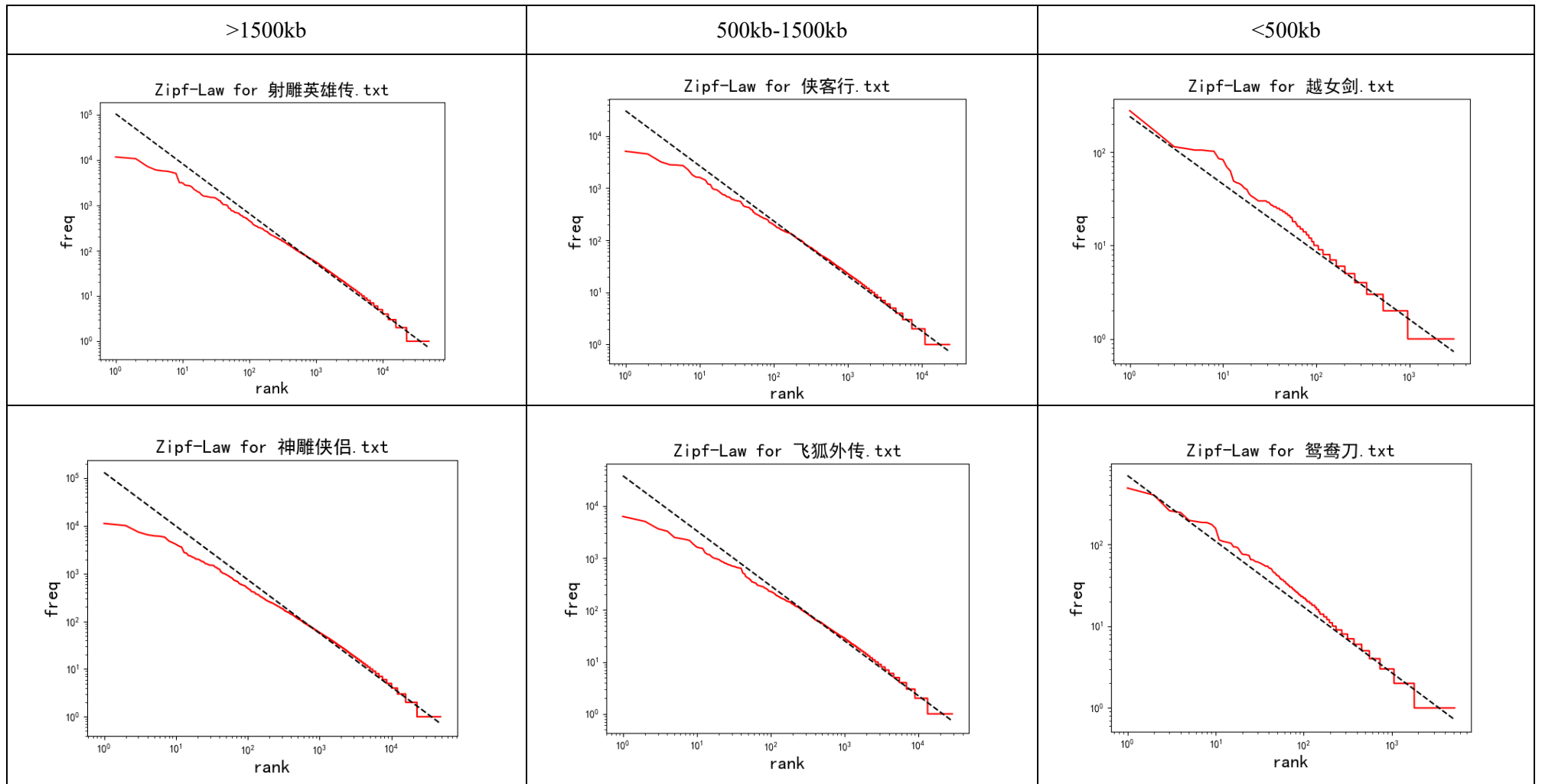


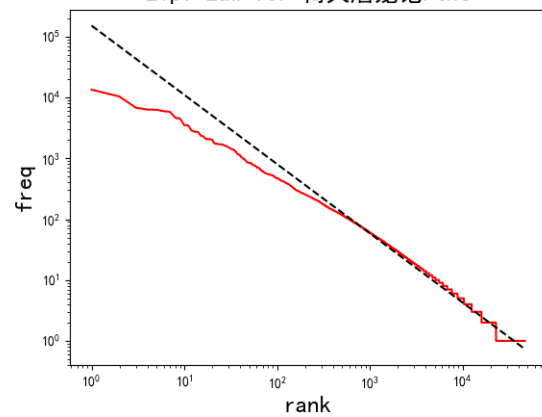
图 2 金庸作品词频统计

所有结果列在表 1 中。表格左列展示了大于 1500kb 的 txt 文件的分词结果，包括射雕英雄传、神雕侠侣、倚天屠龙记、笑傲江湖、天龙八部和鹿鼎记六部长篇小说；表格中间列展示了大于 500kb 小于 1500kb 的 txt 文件的分词结果，包括侠客行、飞狐外传、碧血剑、书剑恩仇录四部中篇小说；表格右列展示了小于 500kb 的 txt 文件的分词结果，包括越女剑、鸳鸯刀、三十三剑客图、白马啸西风、雪山飞狐、连城诀六部短篇小说。表格每一列中，从上到下文件大小逐个增大。

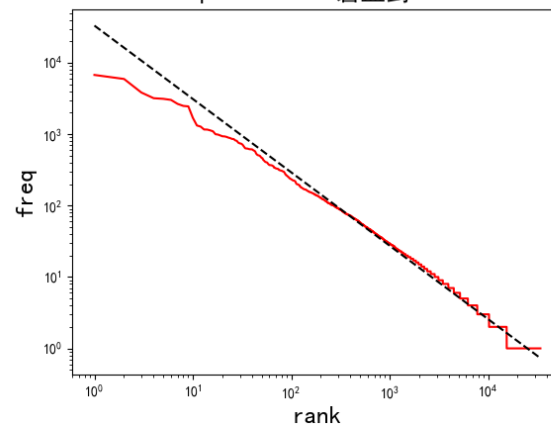
表 1: 金庸作品分词统计结果



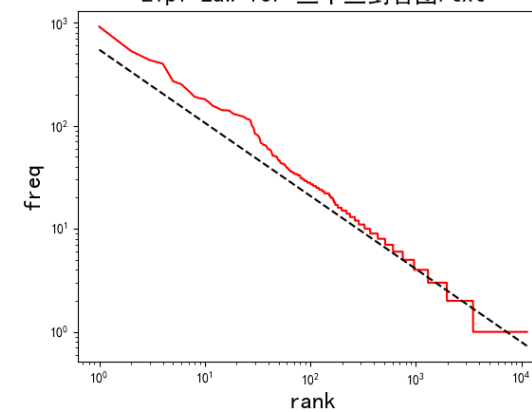
Zipf-Law for 倚天屠龙记.txt



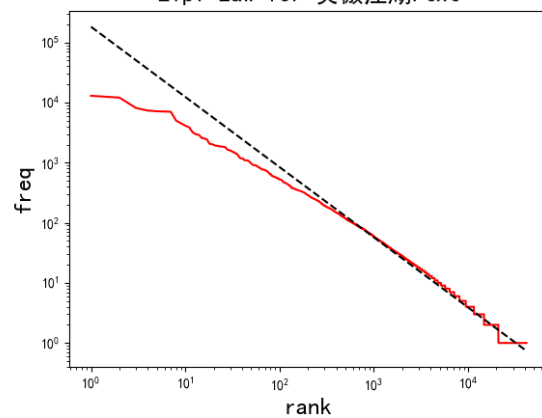
Zipf-Law for 碧血剑.txt



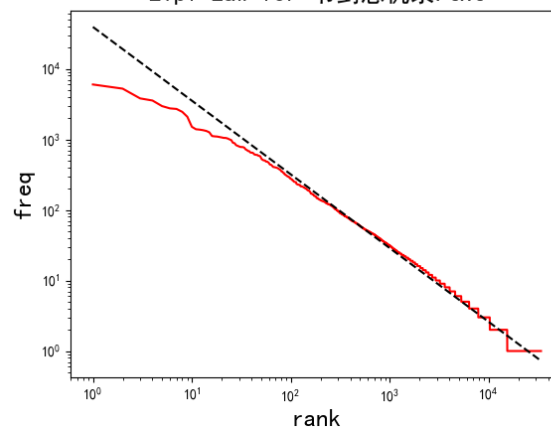
Zipf-Law for 三十三剑客图.txt



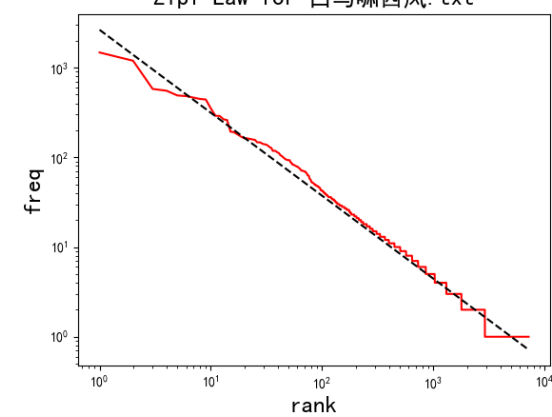
Zipf-Law for 笑傲江湖.txt



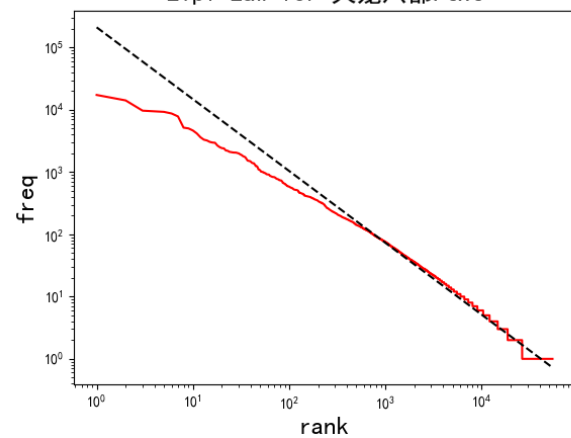
Zipf-Law for 书剑恩仇录.txt



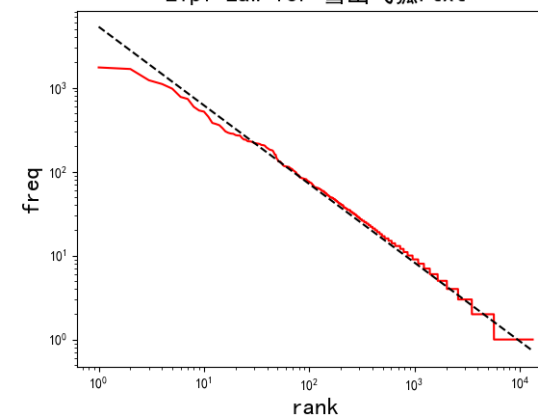
Zipf-Law for 白马啸西风.txt



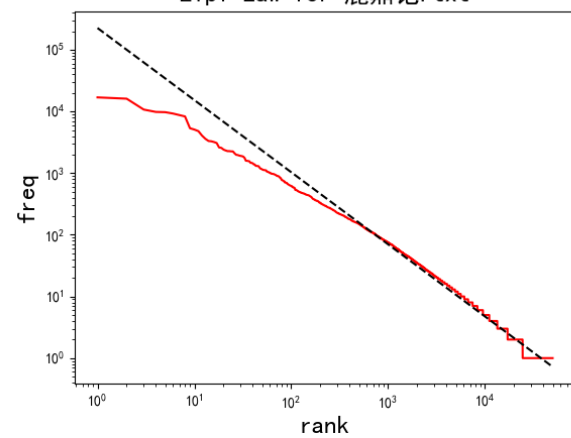
Zipf-Law for 天龙八部.txt



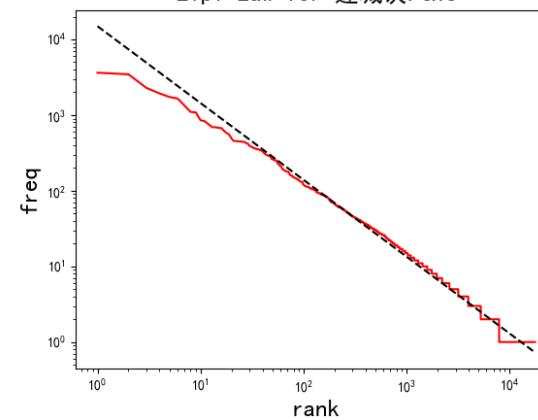
Zipf-Law for 雪山飞狐.txt



Zipf-Law for 鹿鼎记.txt



Zipf-Law for 连城诀.txt



## Part2: 计算信息熵

对 16 个 txt 文件分别计算一元、二元、三元字和词的信息熵，结果在表 2 中。

表 2 金庸作品字\词信息熵计算结果

文件名	一元词信息熵	一元字信息熵	二元词信息熵	二元字信息熵	三元词信息熵	三元字信息熵
三十三剑客图	11.68303	9.668624177	2.939823467	4.835899867	0.271427725	0.982776808
书剑恩仇录	11.71265	9.466103118	5.031292625	5.780877772	1.039173405	2.391070692
侠客行	11.18818	9.152722025	4.957527883	5.5914352	1.126911405	2.368947985
倚天屠龙记	11.75888	9.393314695	5.519119879	6.021232587	1.32874868	2.805208589
天龙八部	11.7343	9.404179512	5.678051182	6.125052342	1.48103104	2.949471381
射雕英雄传	11.82819	9.438643974	5.470219072	6.062218777	1.26723314	2.760470343
白马啸西风	10.24059	8.912318037	3.99186782	4.582907199	0.741411833	1.619539077
碧血剑	11.73701	9.44582074	5.002228229	5.869495153	0.993648124	2.353364968
神雕侠侣	11.73143	9.37266856	5.532062981	6.063134996	1.350274831	2.84622701
笑傲江湖	11.39688	9.20635555	5.620990021	5.897762007	1.527724402	2.870867812
越女剑	10.0781	8.831324971	2.524786494	3.640491254	0.326721118	0.907851902
连城诀	11.0407	9.170997684	4.693122521	5.398802603	0.955601157	2.157033365
雪山飞狐	11.10847	9.201964064	4.112708441	5.163924125	0.693702907	1.758115838
飞狐外传	11.52629	9.307862668	4.996461919	5.75399103	1.053254549	2.407003856
鸳鸯刀	10.47979	9.035696056	3.083838935	4.216055962	0.431698431	1.116186187
鹿鼎记	11.44588	9.281244623	5.764380025	5.99322765	1.618532475	2.953126694



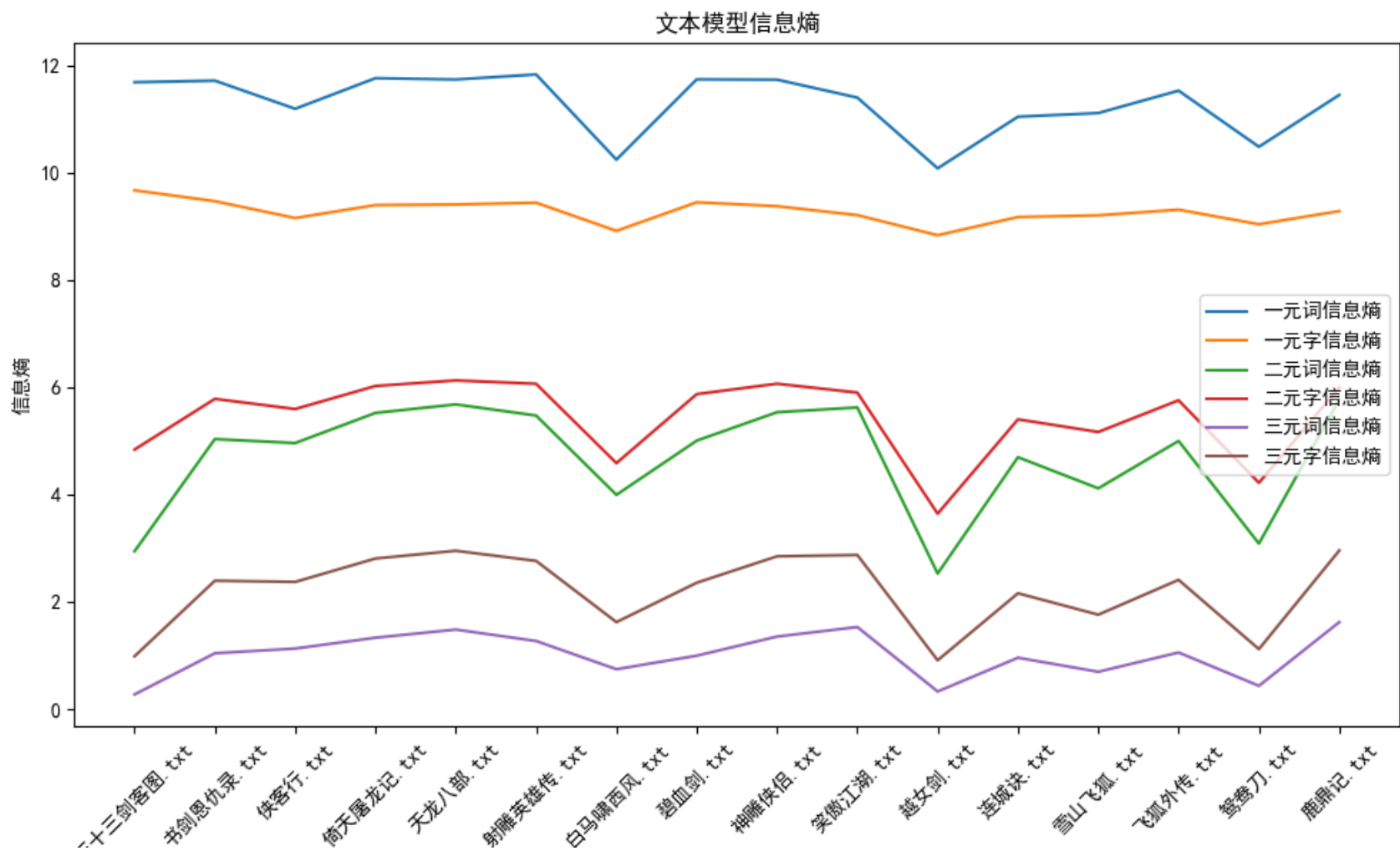


图3 金庸作品字\词信息熵计算结果对比

## Conclusion

对金庸的 16 部作品的 txt 文档进行了分词与词频统计分析，并分别计算了字和词的信息熵。

图 2 显示了所有词频曲线都与一条直线近似，验证了齐夫定律。从表 1 可以发现大于 1500kb 的作品的分词统计图形态几乎相同；小于 500kb 的作品的分词统计图围绕趋势线的波动比较大。这可能说明金庸在长篇小说中的写作风格是比较统一的，在短篇中则有更多的用词变化。

从图 3 中可见，无论是一元、二元、三元语言模型，字\词的信息熵在每个作品间的变化趋势是相同的，这可能说明金庸在统一部作品中的语言风格是相似的。同时可以发现，一元语言模型信息熵大于二元语言模型，二元语言模型信息熵大于三元语言模型，说明字数越多，表意越精确。值得注意的是，在一元语言模型下，词的信息熵大于字的信息熵，但在二元、三元语言模型中，词的信息熵小于字的信息熵。

在使用 Jieba 库进行分词前，对文本进行的预处理中删除了标点符号，这可能会影响分词的准确度，是可以在后续工作中改进的点。

## Reference

Brown P F, Della Pietra S A, Della Pietra V J, et al. An estimate of an upper bound for the entropy of English[J]. Computational Linguistics, 1992, 18(1): 31-40.