宋浩瑜 ZY2203207

摘要

本文基于 Entropy_of_English 论文，使用中文语料库，采用信息熵的公式，进行了中文基于文字以及基于分词的一元、二元、三元模型的计算，并通过停用词消除、jieba 库分字或词等操作估计信息熵的计算方法。本文介绍了中文信息熵的概念和计算方法，并分析了不同领域文本的信息熵值的差异。同时，我们还介绍了二元信息熵和三元信息熵的概念及其在自然语言处理中的应用。

1. 简介

信息熵是信息论中一个重要的概念，用于衡量信息的不确定性和复杂度。中文语言的复杂性和多样性使得计算中文信息熵变得复杂。因此，本文采用信息熵的公式，使用中文语料库，通过停用词消除、jieba 库分字或词等操作，进行了中文基于文字以及基于分字或词的一元、二元、三元模型的计算，并估计了信息熵的计算方法。在本文中，我们分析了不同领域文本的信息熵值的差异，并介绍了二元信息熵和三元信息熵的概念及其在自然语言处理中的应用。

2. 实验方法

2.1 信息熵

信息熵是信息论中用来度量信息的不确定性和复杂度的一种方法。对于一个离散随机变量 X，其信息熵可以通过以下公式计算：

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

其中，P(x) 是 X 取值为 x 的概率。该公式表示了 X 的所有可能取值的信息量的平均值。式中对数一般取 2 为底，单位为比特。但是也可以取其它对数底，采用其它相应的单位，它们间可用换底公式换算。

2.2 语言处理

对于中文文本，其信息熵可以通过以下步骤计算。

假定 S 表示某一个有意义的句子，由一连串特定顺序排列的字或字或词 x1, x2, ...xn 组成，n 为句子的长度。现在想知道 S 在文本中出现的可能性，即 P(S)。此时需要有个模型来估算，不妨把 P(S)展开表示为 P(S)=P(x1, x2, ...xn)。利用条件概率的公式，S 这个序列出现的概率等于每一个字或词出现的条件概率相乘，于是 P(x1, x2, ...xn)可展开为：

$$\mathrm{P(x1, x2, \ldots xn)} = \mathrm{P(x1)P(x2|x1)P(x3|x1, x2)\ldots P(xn|x1, x2, \ldots xn - 1)}$$

其中 P(x1)表示第一个字或词 x1 出现的概率；P(x2|x1)是在已知第一个字或词的前提下，第二个字或词出现的概率；以此类推。

显然，当句子长度过长时，P(xn|x1,x2,...,xn-1)的可能性太多，无法估算。若我们认为每个字或词都是相互独立的，即它的出现与其他的字或词没有关系，那

么 P(x1,x2,...,xn)可展开为：

$$\mathrm{P(S)} = \mathrm{P(x1)P(x2)P(x3)}\ldots\mathrm{P(xi)}\ldots\mathrm{P(xn)}$$

其对应的统计语言模型就是一元模型。

若我们假设任意一个字或词 xi 出现的概率只同它前面的字或词 xi-1 有关，S 的概率变为：

$$\mathrm{P(S)} = \mathrm{P(x1)P(x2|x1)P(x3|x2)}\ldots\mathrm{P(xi|xi-1)}\ldots\mathrm{P(xn|xn-1)}$$

其对应的统计语言模型就是二元模型。也可以假设一个字或词由前面 N-1 个字或词决定，即 N 元模型。

当 N=3 时，每个字或词出现的概率与其前两个字或词相关，为三元模型，对应 S 的概率变为：

$$\mathrm{P(S)} = \mathrm{P(x1)P(x2|x1)P(x3|x1,x2)}\ldots\mathrm{P(xi|xi-2,xi-1)}\ldots\mathrm{P(xn|xn-2,xn-1)}$$

根据以上不同的模型，可以根据其联合分布的随机变量，将信息熵改编成联合信息熵：

$$\mathrm{H(X|Y)} = -\sum_{y\in Y} P(y)\log\big(P(x|y)\big)$$

$$= -\sum_{y\in Y} P(y)\sum_{x\in X} P(x)\log\big(P(x|y)\big)$$

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

该联合信息熵可以用于后文的二元模型与三元模型的计算。

3. 实验过程

3.1 数据预处理

3.1.1 停词处理

停词包括文章中的标点以及常见语气助词等无实意的字或词，在查看语料库中，我们发现文章中仍有一些英文出现，并且有一些标点在被特定语言读取过程中可能会出现转义的情况，为了防止以上情况出现对于中文信息熵统计的影响，我们在原有停词表中做了以下处理：

1) 在停词表中增加小写、大写英文共 52 个字母
2) 在停词表中，对于一些特定标点增加转义字符比如将“\”调整为”\\”
3) 新增一些停词比如英文的逗号“,”等

3.1.2 分词处理

本文使用了 python 的 jieba 库来进行中文词汇的分词，该库的主要任务是将读取的字符串，按照数据库中的中文词汇，将中文字符串分成多个词组，便于后面进行词组信息熵的计算

3.1.3 数据的获取和处理

数据集为金庸先生的 16 本小说，其中包含了大量乱码与无用或重复的中英

文符号，因此需要对该实验数据集进行预处理。

具体来说就是，首先要删除无意义的字符比如空格、回车、制表符、段落符；其次根据已经获取的停词表，将相应的停词，如标点、英文字母、阿拉伯数字、无实意语气词等进行删除；最后可以选择直接使用汉字，或者使用 jieba 库将长字符串进行分词，得到多个词组。

3.2 计算信息熵

如果统计量足够，根据大数定理，词或二元词组或三元词组出现的概率大致等于其出现的频率。

一元模型的信息熵计算公式为

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

其中 P(x)可近似等于每个词在语料库中出现的频率。

二元模型的信息熵计算公式为

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

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词组在语料库中出现的频数与以该二元词组的第一个词为词首的二元词组的频数的比值。

三元模型的信息熵计算公式为

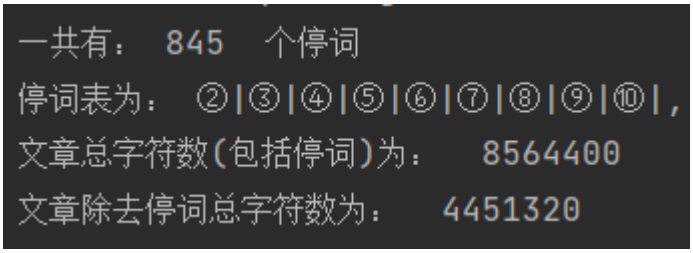
$$\mathrm{H(X|Y,Z)} = -\sum_{y \in Y, x \in X, z \in X} P(x,y,z)\log\big(P(x|y,z)\big)$$

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。

4. 实验结果

在本实验中，我们采用了基于 Python 的分字或词工具 jieba 进行实验。

总共整理的停词一共有 845 个，算上中英文标点，英文字母大小写以及一些中文的无实意词汇。语料库所有字符一共 87564400 个字符，出去空格等占位符以及停词之后剩余中文字符为 4451320 个汉字，汉字的种类为 5569。



首先我们尝试不使用分词，将语料库中的每个文字都当作相互独立的分布，搭建一元模型，语料库的总字数为 4451320 字，分词个数也就是汉字种类为 5569

字，平均词长为 799.30329，计算得到不分词基于词的一元模型的中文信息熵为 9.95495 比特/词。

```
一元模型长度： 4451320
出现的汉字种类为： 5569
平均词长: 799.30329
基于词的一元模型的中文信息熵为: 9.95495 比特/词
一元运行时间： 1.46821 s
```

之后采用了基于 Python 的分词工具 jieba 库进行分词。

在一元模型中使用的语料库字数为 2406939 字，分词个数为 261591 词，平均词长为 9.20115，计算得到基于词的一元模型的中文信息熵为 13.91353 比特/词。

```
一元模型长度： 2406939
出现的汉字种类为： 261591
平均词长: 9.20115
基于词的一元模型的中文信息熵为: 13.91353 比特/词
一元运行时间： 0.824 s
```

在二元模型中使用的语料库字数为 2406938 字，分词个数为 1738971 词，平均词长为 1.38412，计算得到基于词的二元模型的中文信息熵为 6.16066 比特/词。

```
二元模型长度： 2406938
出现的二元词组种类为： 1738971
平均词长: 1.38412
基于词的二元模型的中文信息熵为: 6.16066 比特/词
运行时间: 3.378 s
```

在三元模型中使用的语料库字数为 2406937 字，分词个数为 2308125 词，平均词长为 1.04281，计算得到基于词的二元模型的中文信息熵为 0.98183 比特/词。

```
三元模型长度: 2406937
出现的三元词组种类为： 2308125
平均词长: 1.04281
基于词的三元模型的中文信息熵为: 0.98183 比特/词
运行时间： 7.504 s
总共运行时间为： 50.63728 s
```

之后，我们又将金庸的 16 本小说拆开单独每部都进行分词的中文信息熵的计算。得到的结果如下表所示。

文章	分词数	一元平均词长	二元平均词长	三元平均词长	一元信息熵（比特/	二元信息熵（比特/	三元信息熵（比特/

					词）	词）	词）
白马啸西风	18894	2.74104	1.10621	1.01016	11.20355	2.71555	0.26341
碧血剑	138518	3.83611	1.14828	1.01589	12.93267	3.72926	0.37936
飞狐外传	124436	4.00451	1.15835	1.01841	12.71551	3.7769	0.38452
连城诀	62756	3.49091	1.13154	1.01417	12.2444	3.35008	0.30826
鹿鼎记	331493	5.49184	1.26548	1.03241	12.87692	4.706	0.66226
三十三剑客图	18541	1.87132	1.03813	1.00788	12.44395	1.64967	0.06839
射雕英雄传	258556	4.66097	1.18271	1.01739	13.1366	4.33668	0.46128
神雕侠侣	281054	5.04468	1.23495	1.04403	12.89119	4.36288	0.53801
书剑恩仇录	147809	4.07929	1.1634	1.01683	12.78319	3.91887	0.42717
天龙八部	333473	5.16078	1.22367	1.02693	13.18211	4.53363	0.56326
侠客行	99420	3.94117	1.18319	1.0232	12.34739	3.74133	0.45361
笑傲江湖	266684	5.35059	1.28173	1.03847	12.61731	4.57198	0.72479
雪山飞狐	37439	2.85903	1.11128	1.02231	12.12594	2.78409	0.23192
倚天屠龙记	272742	4.99619	1.22062	1.02511	13.00974	4.41928	0.56295
鸳鸯刀	10206	2.27255	1.08333	1.0107	10.99629	2.11402	0.18125
越女剑	4891	2.00287	1.08715	1.01075	10.2648	1.73819	0.22671

结论

本实验研究了如何测量中文的信息熵。应用了不分词、分词以及一元、二元、三元的发放测量语料库的中文信息熵。研究过程中能较为充分的考虑非实意中文

的情况，应用停词表进行筛选，并对比不同语料库的结果，进行总结和归纳，最后得到不分词基于词的一元模型的中文信息熵为 9.95495 比特/词；基于词的一元模型的中文信息熵为 13.91353 比特/词；基于词的二元模型的中文信息熵为 6.16066 比特/词；基于词的二元模型的中文信息熵为 0.98183 比特/词的结论。

可以看出随着 N 的增大，平均词长不断减小，并且文中所含的信息熵也在减小，为了进一步验证以上结论，我们分开对每本小说进行计算。发现平均词长和中文信息熵与分词数有较为明显的关系，或者说与样本集的大小有明显关系。以小说《笑傲江湖》与《越女剑》为例，前者有 266684 的分词以及 0.72479 的三元信息熵，后者仅有 4891 的分词以及 0.22671 的三元信息熵。可以看出样本越大其检测的结果越接近真实。

不过本文研究过程中仍发现了一些问题，检测的结果可能也与文本的类型有关，由于数据库内容都为小说，无法参照其他类型的语料进行对比。并且停词表中有“过”和“去”这种类型的停词，可能单一的汉字确实没有实意，不过有些组合的词汇比如“过去”、“过客”是有实际意义的，使用停词删除可能会造成信息的丢失，最终无法正确统计信息熵。