中文开放NLP讲义

李正华、、、等编著

对某一部分有贡献的同学，在每一章介绍其贡献；
贡献非常大的同学或老师（等完整版本出来），
并愿意长期维护和完善本书，最终放到封皮上

September 3, 2020

1 汉字编码问题

2005年夏天，我刚刚大三，开始接触NLP。有很长一段时间内我都困惑于汉字编码问题。汉字在计算机中如何表示？如何将一句话（可能中英文混合），切分为一个个字符。我知道，一句话在内存中对应一串连续的字节。我也知道GB（国标）编码和Unicode编码。但是具体怎么弄我感觉很乱。很可能是我的表达不清楚，问了一些师兄师姐也没有搞清楚。直到看了一个代码，才知道如何处理GB编码。但是对于Unicode的处理，是在很长时间后，看了另外一段代码才大概搞清楚。

真正把汉字编码弄清楚，要等到2015年我讲《中文信息处理》这门课时，专门了解了一下，才算真正清楚了。

1.1 GB编码

GB（国家标准）。GBK（国标扩展）支持更多的字符，包括繁体和简体。关于GB和GBK的关系，网上有很多。https://zhidao.baidu.com/question/568647091.html?qbl=relate_question_1&word=GBK%B1%E0%C2%EB和https://zhidao.baidu.com/question/243915749.html。

简单来讲，GB编码用2个字节来表示汉字。必须从左向右扫描，如果当前字节（byte）的首位（bit）是0，那么当前字节独立对应一个字符，ASCII码）；如果当前字节的首位是1，那么当前字节和下一个字节形成一个汉字。

可以计算一下，256*128=32768，足可以表示所有的汉字了。

1.2 UTF8编码

Unicode的目的是编码世界上所有的文字。UTF8是Unicode的一种具体实现，最常用。其他实现方式还有UTF7、UTF16、UTF32，在一些特定场景下使用。

UTF8用1-6个字节来表示一个字符。必须从左向右扫描，根据当前字节前几位，来决定当前字符由多少个字节构成。不但如此，后面字节的前2位为10，这一点可以

Table 1.1: UTF8编码规则，x表示0或1都可以。

字符对应的字节数	第0个字节	第1个字节	2	3	4	5
1	0xxx xxxx					
2	110x xxxx	10xx xxxx				
3	1110 xxxx	10xx xxxx	10xx xxxx			
4	1111 0xxx	10xx xxxx	10xx xxxx	...		
5	1111 10xx	10xx xxxx	10xx xxxx	...	...	
6	1111 110x	10xx xxxx	10xx xxxx	...	...	...

用来进一步验证UTF8编码的合法性。具体规则可以查看表1.1。详细情况可以看https://baike.baidu.com/item/UTF-8。

1.3 其他

Linux下有一个命令iconv，可以对文本文件进行编码转化。

```
iconv -f utf8 -t gbk a.txt
iconv -t utf8 -f gbk a.txt
```

Python（3.0）之后，支持Unicode编码，可以直接通过split，将一句话对应的字符串切分为字符（字）序列。也可以进行正则匹配。

Big Endian和Little Endian ：与汉字编码关系不大。只是针对大于1个字节的对象，比如Double、Int等，才需要考虑。

有些文本编辑软件，会根据文件编码，在文本文件前面插入几个字节，以作标识。这一点我现在还没搞清楚，同时感觉不重要，所以就不管了。