中文信息处理

期末复习提纲

1 题型

1.1 单项选择题（Perl 编程）

1. 变量类型转换
2. 变量内插
3. 引用
4. 函数传参
5. 正则表达式：分组捕获、修饰符

1.2 简答题

1. 简述中文信息处理的主要问题以及通常的解决方法。
2. 结合课后作业简述中文的字、词频统计的技术要点。
3. 简述最大匹配法分词的基本思想和操作方法。
4. 简述采用隐马尔可夫模型（HMM）进行自动词性标注的基本原理。
5. 简述 TF-IDF 权重计算的基本原理及其在 NLP 领域的具体应用。

1.3 综合题

1. 编码格式判断、Perl 文件读写
2. 图解法析句

2 Perl 语言

2.1 变量

2.1.1 类型

名称	类型	标记	备注
Scalar	标量	$	**字母**或_开头 **字母**或**数字**随后
Array	数组、列表	@	

Hash	哈希、散列	%	
Handler	文件句柄		全大写

2.1.2 标量

2.1.2.1 数据类型

- 数值
- 字符串

2.1.2.2 类型转换

2.1.2.2.1 数值 > 字符串

最简十进制数字表达式

2.1.2.2.2 字符串 > 数值

第一个字符开始，截取合法数值表达式；

第一个非法字符开始，舍弃；

无合法字符，为 0。

2.1.2.3 算术运算符

2.1.2.3.1 数值

+ - * /

2.1.2.3.2 字符串

- 连接：.
- 重复：x

2.1.2.4 逻辑运算符

类型	数值	字符串
等于	==	eq
不等于	!=	ne
大于	>	gt
大于等于	>=	ge
小于	<	lt
小于等于	<=	le

2.1.2.5 转义和内插

单引号字符串**不**转义、**不**内插变量

2.2 文件读写

2.2.1 读一行

<文件句柄>;

2.2.2 写任意内容

print <文件句柄> $var[, …];

2.2.3 打开、关闭文件句柄

open(文件句柄, 文件名);

文件名：

- “<filename”或”filename”：读入句柄
- “>filename”：写入覆盖句柄
- “>>filename”：写入追加句柄

close(文件句柄);

2.2.3.1 指定编码

open(IN, '**<:encoding(gbk)**', a.txt') or die 'Failed to open file.';

open(OUTF, '**>:raw:encoding(utf-8)**', 'b.txt');

2.2.4 逐行读取

while (<文件句柄>) {

… $_ …

}

while (<*>) … 读目录项

2.2.5 –X EXPR

文件是否存在：-e “filename”

是否为目录：-d “name”

2.2.6 一次读完

- $all = join ‘’, <INF>;
- undef $/;
  $all = <INF>;

2.3 数组（列表）

2.3.1 获取一项

@arr = (1, 2, 3);

print $arr[2]; # Print 3

2.3.2 数组长度

自动增加。如：

@nums = (1, 2);

$nums[3] = 6; # 1, 2, undef, 6

$nums[2]++; # 1, 2, 1, 6

push @nums, 5; # 1, 2, 1, 6, 5

pop @nums; # 1, 2, 1, 6

unshift @nums, 1; # 1, 1, 2, 1, 6

shift @nums; # 1, 2, 1, 6

2.3.3 列表

2.3.3.1 基本形式

(1, “a”, $x, $x+2, @arr) 元素任意

2.3.3.2 qw

qw(1 2 3 str $a \t)

- 若干空白符号分割元素
- 均为字面值，**不转义、不内插**
- ()可以用其他配对符号替代，如<>、//、##等

2.3.4 数组遍历

for $x (@arr) {}

for $i (0 .. $#arr) {} # **$#arr** 为最大标号

2.3.5 列表赋值

@arr = (1, 2, 3);

($a, $b, $c) = @arr; # $a = 1, $b = 2, $c = 3

($a, $b) = ($b, $a); # 交换

2.3.6 数组切片

@a = (0..5);

@sub = @a[0, 3]; # @sub = ($a[0], $a[3])

@a[1, 3] = (“a”, “b”); # $a[1] = “a”, $a[3] = “b”

@a[0, -1] = @a[-1, 0];# 交换头尾元素

@b = (1, 3);

@sub = @a[@b]; # #sub = @a[1, 3]

2.3.7 数组内插

@arr = (0..3);

print “@arr”; # Print 0 1 2 3

“@arr” 等价于 join($”, @arr)，其中$”默认为空格，可以更改

2.4 哈希

2.4.1 表示

```
%sp = (“the”, 6, “half”, 3);

%sp = (
    “the”    =>  6,
    “half”    =>  3,
);
```

2.4.2 读取

```
$val = $hash{$key};

if (exists $hash{$key}) {}
```

2.4.3 写入

```
$hash{$key} = $val;

$hash{$key}++;
```

若修改不存在键值，Perl 自动建立其为 undef。

2.4.4 遍历

```
@arr = keys %hash;

for $key (@arr) {
    $val = $hash{$key};
}


while(($k, $v) = each %hash) {
}
```

2.4.5 哈希切片

```
%h = (‘a’=>1, ‘b’=>2, ‘c’=>3);

@vals = @h{‘a’, ‘c’};

@h{‘b’, ‘c’} = (20, 30);

@h{‘a’..’z’} = 1..26;
```

2.5 正则表达式

2.5.1 正则开关

/g：全局匹配（global）

/i：忽略大小写（ignorecase）

/s：令.匹配换行符（single line）

/m：令^和$匹配下一个嵌入的\n（multiline）

/x：忽略（大多数）空白且允许注释

2.5.2 匹配

=~匹配

!~不匹配

m/FOO/ # m 可省略

2.5.3 替换

$x = ~ **s/**FOO/BAR**/g # /g 则全局**

2.5.4 配对符号

//可换为其他配对符号，如##、()等，但此时匹配的 m 不可省略

=~省略，则对$_匹配或替换。

2.5.5 贪婪模式

$_ = “aabbccaaabbbcccd”;

/a.*c/; # 匹配 aabbccaaabbbcccd

/a.*?c/; # 匹配 aabbc

2.5.6 捕获分组

$_ = “2015-03-16”;

/(\d{4})-(\d\d)-(\d\d)/; # $1 = “2015”, $2 = “03”, $3 = “16”

($y, $m, $d) = /(\d{4})-(\d\d)-(\d\d)/;

s#(\d{4})-(\d\d)-(\d\d)#$2/$3/$1#; # 2015-03-16 -> 03/16/2015

2.5.7 非捕获分组

(**?:**…)

2.5.8 反向引用

@vv = /(..?)了一\1/;

匹配“看了一看”

2.5.9 全局匹配

/…/g

2.5.10 循环匹配

while (/(\d)(\D)/g) {

print “$1, $2\n”;

}

2.5.11 预搜索

2.5.11.1 正向预搜索

(?=PATTERN) # 向右看，有：PATTERN

(?!PATTERN) # 向右看，无：PATTERN

2.5.11.2 反向预搜索

(?<=PATTERN) # 向左看，有：PATTERN

(?<!PATTERN) # 向左看，无：PATTERN

2.5.12 汉字

Unicode 范围内：汉字\p{Han}，非汉字\P{Han}

2.6 子程序

2.6.1 定义

sub subname {

}

2.6.2 调用

&subname; # 无参

&subname(…); # 参数

subname … # 无歧义，省略参数

先定义后调用，可省略&

2.6.3 参数

Perl 将参数列表化为@_

sub println {print @_, “\n”}

@arr = (1, 2, 3);

&println(‘a’, ‘b’, @arr);# 即 print(‘a’, ‘b’, 1, 2, 3, “\n”);

2.6.4 返回值

return 或最后执行语句的值

sub max{ ($x, $y) = @_; if ($x > $y) { return $x; } else { return $y;} }

sub max { ($x, $y) = @_; if ($x > $y) { return $x; } $y; }

sub max{ ($x, $y) = @_; if ($x > $y) { $x } else { $y } }

sub max{ ($x, $y) = @_; $x > $y ? $x : $y; }

2.6.5 局部变量

my $x;

2.7 排序

sort {$a cmp $b} @arr; # 字符串比较

sort {$a <=> $b} @arr # 数值比较

2.8 引用

引用本身为标量（$开头）。

@arr1 = qw(1 2);

@arr2 = (1, 2, @arr1, 3); # 扁平化：1, 2, 1, 2, 3

@arr3 = (1, 2, \@arr1, 3);

2.8.1 匿名引用

2.8.1.1 数组

用[]

$aref = [1, 2];

2.8.1.2 哈希

用{}

$href = {'APR' => 4, 'AUG' => 8};

2.8.2 解引用

2.8.2.1 方法一

普通变量	引用	简写
$s	${$sref}	$$sref
@a	@{$aref}	@$aref
$a[3]	${$aref}[3]	$$aref[3]
%h	%{$href}	%$href
$h{'red'}	${$href}{'red'}	$$href{'red'}

2.8.2.2 方法二

$aref->[3]; # $$aref[3]

$href->{'red'}　　# $$href['red']

$aref = [[1,2,3],[4,5,6],[7,8,9]];

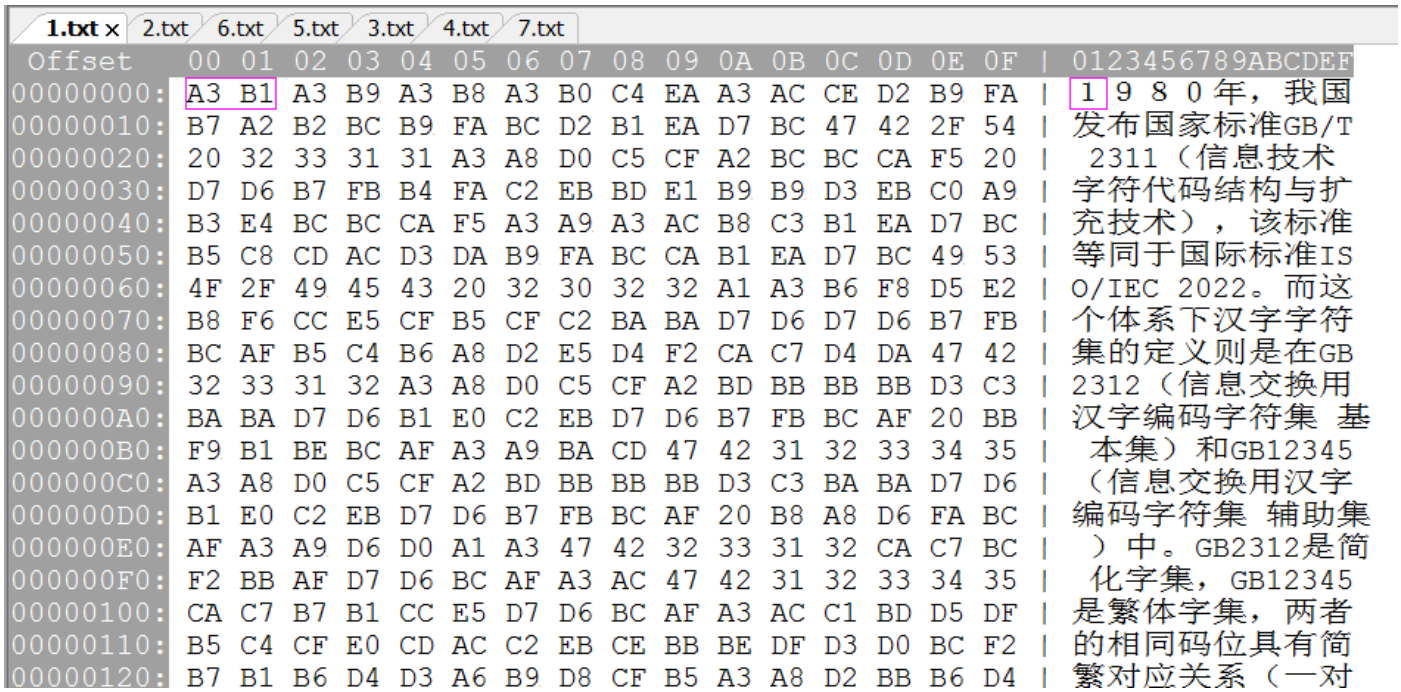
$aref->[2][2]　　# $aref->[2]->[2]，可省略后面的->

3 中文信息处理

3.1 编码判别

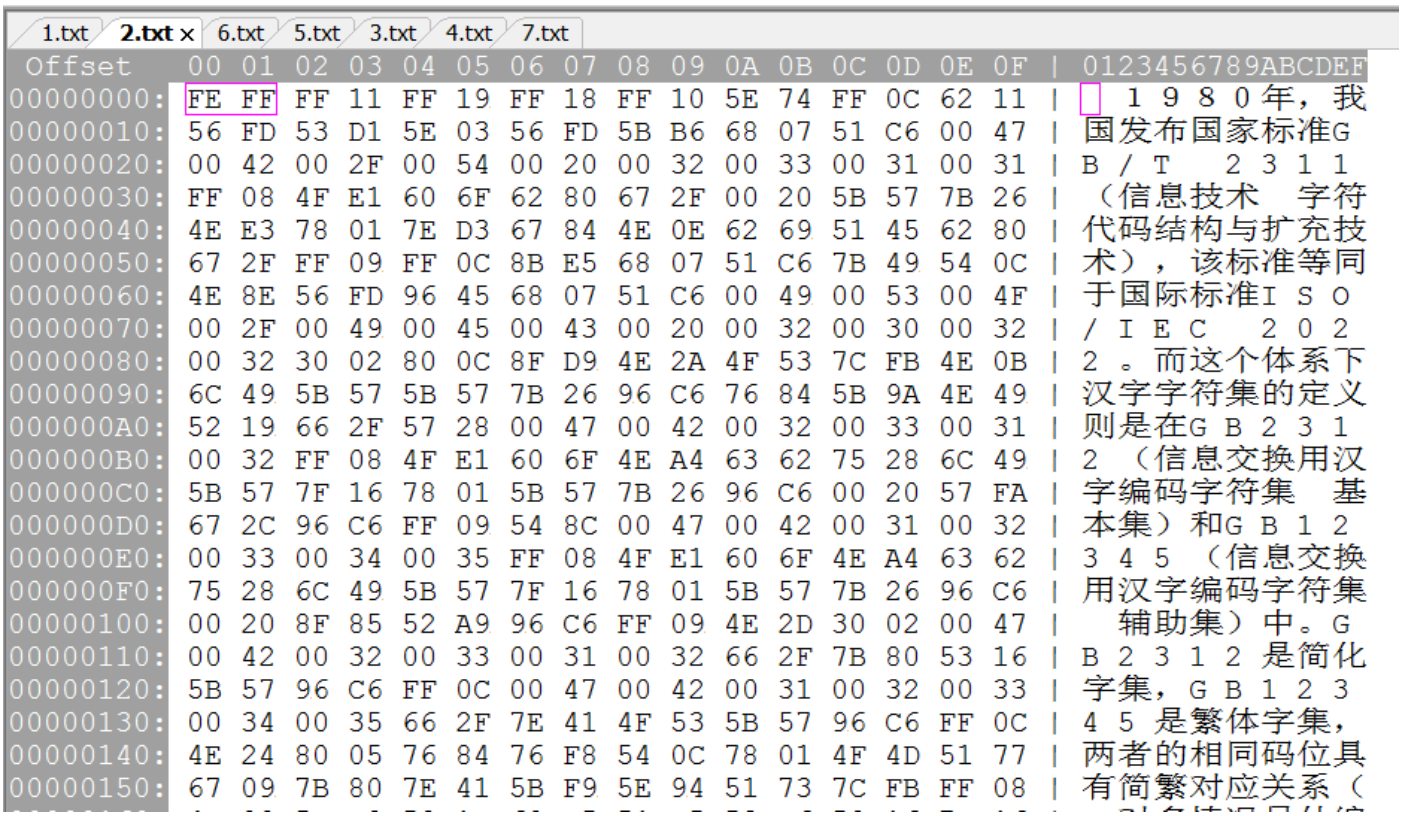
3.1.1 GB

一个汉字 2 字节，英文数字 1 字节。



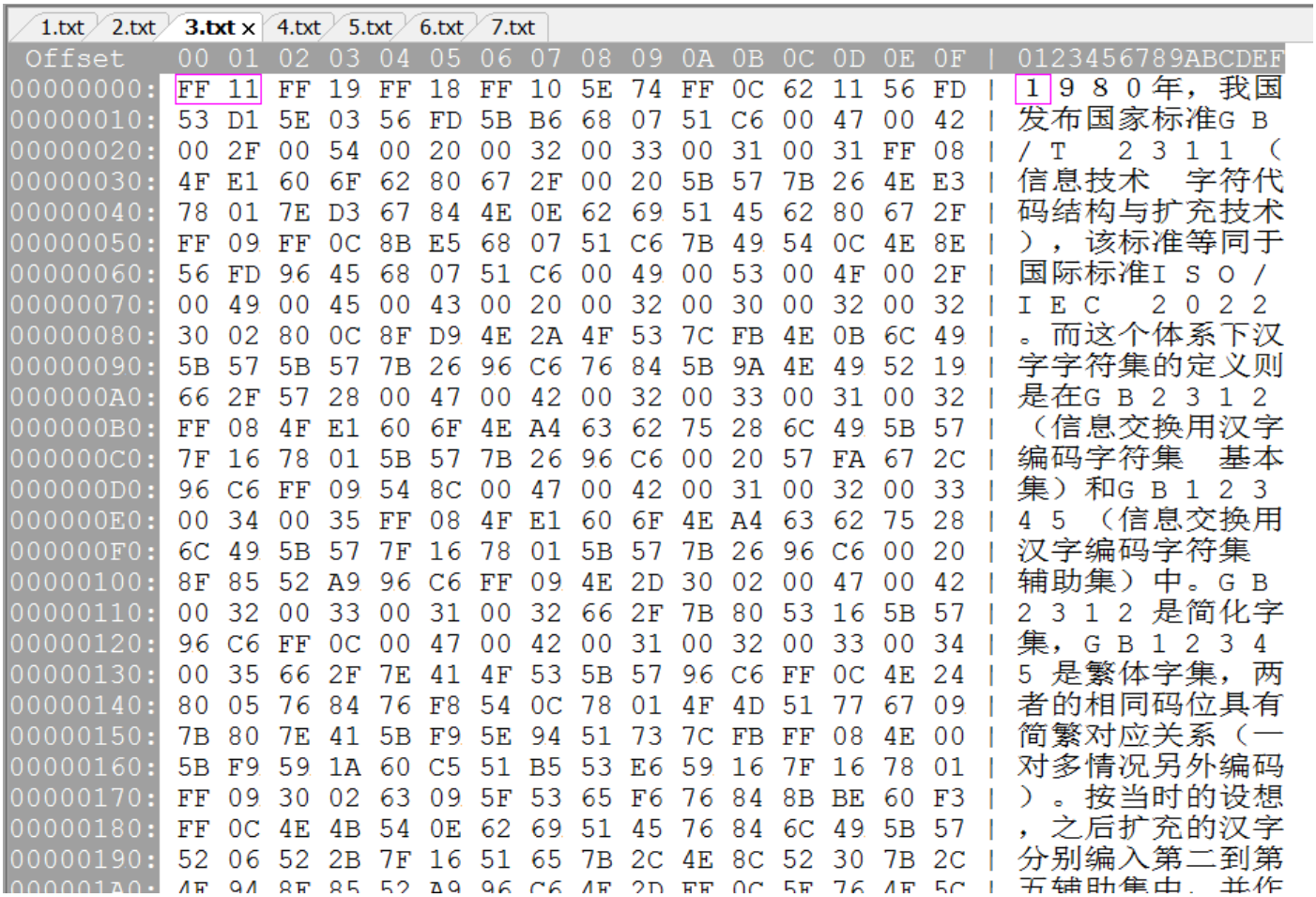
3.1.2 UTF16-BE with BOM

全部 2 字节。FE FF 的 BOM 头。



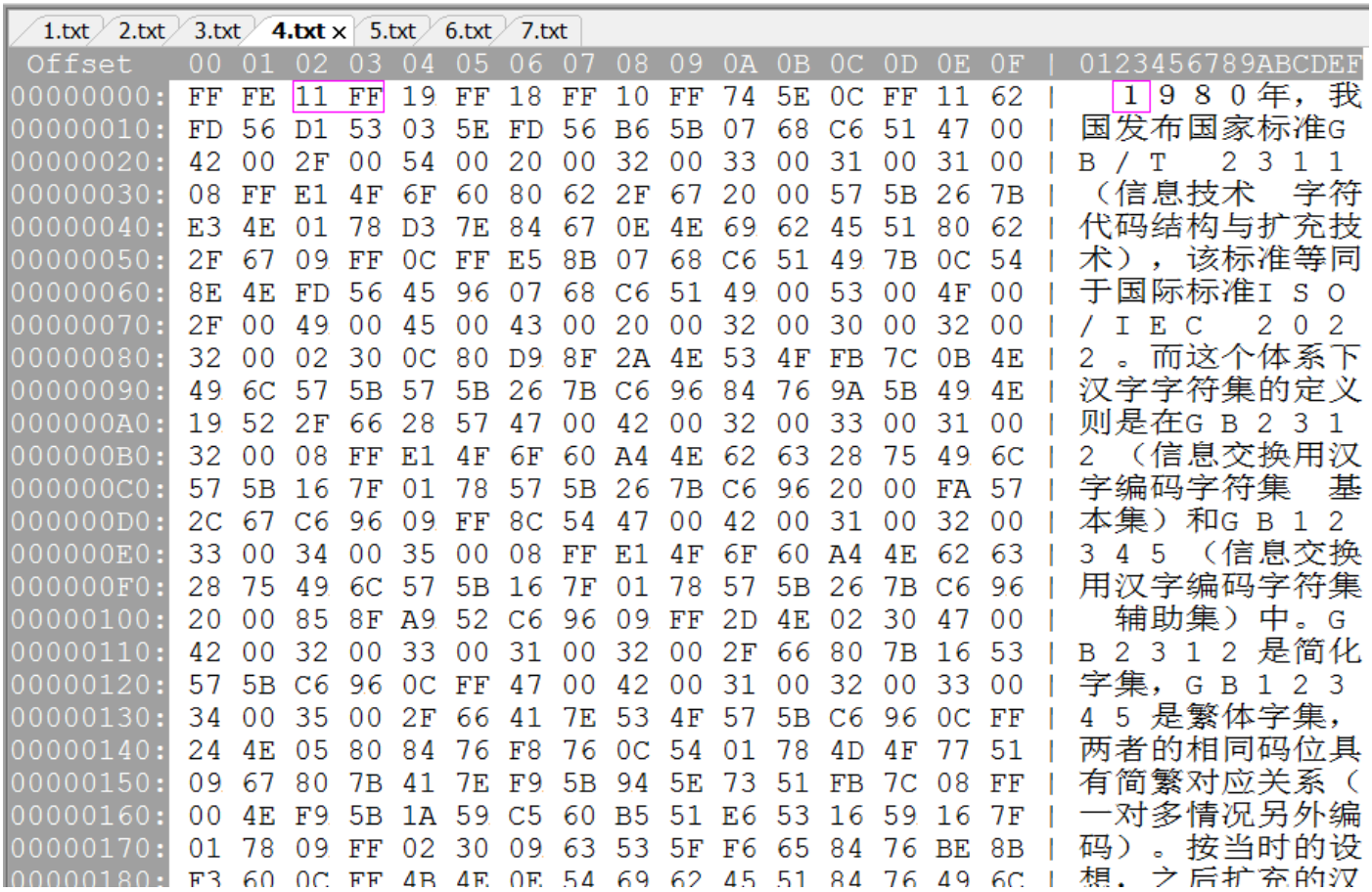
3.1.3 UTF16-BE

全部 2 字节。B 为 00 42，即知是 BE 了。



3.1.4 UTF16-LE with BOM

全部 2 字节。FF FE 的 BOM 头。



3.1.5 UTF16-LE

全部 2 字节。B 为 42 00，即知是 LE 了。

1.txt 2.txt 3.txt 4.txt **5.txt** × 6.txt 7.txt

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000:	11	FF	19	FF	18	FF	10	FF	74	5E	0C	FF	11	62	FD	56	1980年，我国
00000010:	D1	53	03	5E	FD	56	B6	5B	07	68	C6	51	47	00	42	00	发布国家标准GB
00000020:	2F	00	54	00	20	00	32	00	33	00	31	00	31	00	08	FF	/T 2311（
00000030:	E1	4F	6F	60	80	62	2F	67	20	00	57	5B	26	7B	E3	4E	信息技术 字符代
00000040:	01	78	D3	7E	84	67	0E	4E	69	62	45	51	80	62	2F	67	码结构与扩充技术
00000050:	09	FF	0C	FF	E5	8B	07	68	C6	51	49	7B	0C	54	8E	4E	），该标准等同于
00000060:	FD	56	45	96	07	68	C6	51	49	00	53	00	4F	00	2F	00	国际标准ISO/
00000070:	49	00	45	00	43	00	20	00	32	00	30	00	32	00	32	00	IEC 2022
00000080:	02	30	0C	80	D9	8F	2A	4E	53	4F	FB	7C	0B	4E	49	6C	。而这个体系下汉
00000090:	57	5B	57	5B	26	7B	C6	96	84	76	9A	5B	49	4E	19	52	字字符集的定义则
000000A0:	2F	66	28	57	47	00	42	00	32	00	33	00	31	00	32	00	是在GB2312
000000B0:	08	FF	E1	4F	6F	60	A4	4E	62	63	28	75	49	6C	57	5B	（信息交换用汉字
000000C0:	16	7F	01	78	57	5B	26	7B	C6	96	20	00	FA	57	2C	67	编码字符集 基本
000000D0:	C6	96	09	FF	8C	54	47	00	42	00	31	00	32	00	33	00	集）和GB123
000000E0:	34	00	35	00	08	FF	E1	4F	6F	60	A4	4E	62	63	28	75	45 （信息交换用
000000F0:	49	6C	57	5B	16	7F	01	78	57	5B	26	7B	C6	96	20	00	汉字编码字符集
00000100:	85	8F	A9	52	C6	96	09	FF	2D	4E	02	30	47	00	42	00	辅助集）中。GB
00000110:	32	00	33	00	31	00	32	00	2F	66	80	7B	16	53	57	5B	2312是简化字
00000120:	C6	96	0C	FF	47	00	42	00	31	00	32	00	33	00	34	00	集，GB1234
00000130:	35	00	2F	66	41	7E	53	4F	57	5B	C6	96	0C	FF	24	4E	5 是繁体字集，两
00000140:	05	80	84	76	F8	76	0C	54	01	78	4D	4F	77	51	09	67	者的相同码位具有
00000150:	80	7B	41	7E	F9	5B	94	5E	73	51	FB	7C	08	FF	00	4E	简繁对应关系（一
00000160:	F9	5B	1A	59	C5	60	B5	51	E6	53	16	59	16	7F	01	78	对多情况另外编码
00000170:	09	FF	02	30	09	63	53	5F	F6	65	84	76	BE	8B	F3	60	）。按当时的设想

3.1.6 UTF8 with BOM

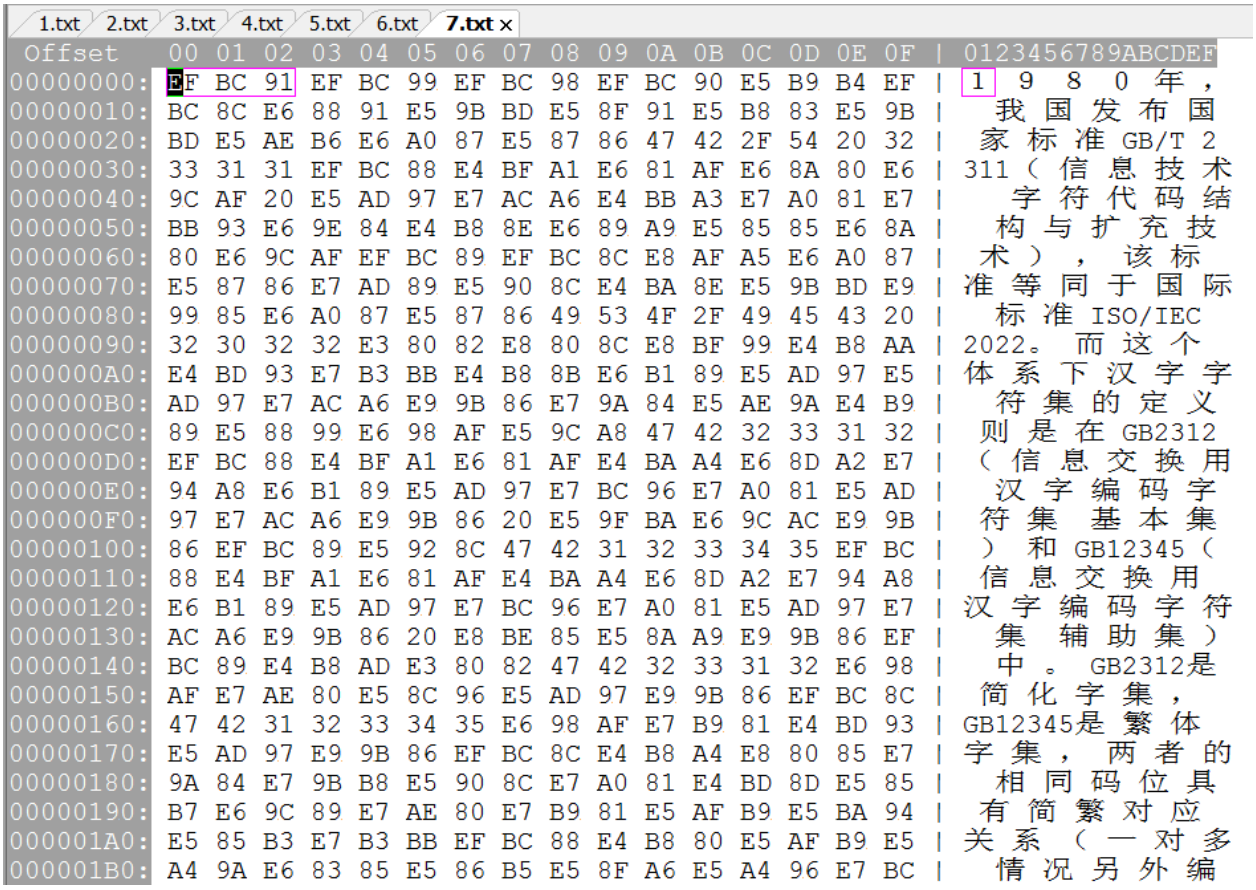
中文 3 字节，英文数字 1 字节，UTF8。BOM 为 EF BB BF。

1.txt 2.txt 3.txt 4.txt 5.txt **6.txt** × 7.txt

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000:	EF	BB	BF	EF	BC	91	EF	BC	99	EF	BC	98	EF	BC	90	E5	1 9 8 0 年
00000010:	B9	B4	EF	BC	8C	E6	88	91	E5	9B	BD	E5	8F	91	E5	B8	，我国发布
00000020:	83	E5	9B	BD	E5	AE	B6	E6	A0	87	E5	87	86	47	42	2F	国家标准GB/
00000030:	54	20	32	33	31	31	EF	BC	88	E4	BF	A1	E6	81	AF	E6	T 2311（信息技
00000040:	8A	80	E6	9C	AF	20	E5	AD	97	E7	AC	A6	E4	BB	A3	E7	术字符代码
00000050:	A0	81	E7	BB	93	E6	9E	84	E4	B8	8E	E6	89	A9	E5	85	结构与扩充
00000060:	85	E6	8A	80	E6	9C	AF	EF	BC	89	EF	BC	8C	E8	AF	A5	技术），该
00000070:	E6	A0	87	E5	87	86	E7	AD	89	E5	90	8C	E4	BA	8E	E5	标准等同于国
00000080:	9B	BD	E9	99	85	E6	A0	87	E5	87	86	49	53	4F	2F	49	际标准ISO/I
00000090:	45	43	20	32	30	32	32	E3	80	82	E8	80	8C	E8	BF	99	EC 2022。而这
000000A0:	E4	B8	AA	E4	BD	93	E7	B3	BB	E4	B8	8B	E6	B1	89	E5	个体系下汉字
000000B0:	AD	97	E5	AD	97	E7	AC	A6	E9	9B	86	E7	9A	84	E5	AE	字符集的定
000000C0:	9A	E4	B9	89	E5	88	99	E6	98	AF	E5	9C	A8	47	42	32	义则是在GB2
000000D0:	33	31	32	EF	BC	88	E4	BF	A1	E6	81	AF	E4	BA	A4	E6	312（信息交换
000000E0:	8D	A2	E7	94	A8	E6	B1	89	E5	AD	97	E7	BC	96	E7	A0	用汉字编码
000000F0:	81	E5	AD	97	E7	AC	A6	E9	9B	86	20	E5	9F	BA	E6	9C	字符集 基本
00000100:	AC	E9	9B	86	EF	BC	89	E5	92	8C	47	42	31	32	33	34	集）和GB1234
00000110:	35	EF	BC	88	E4	BF	A1	E6	81	AF	E4	BA	A4	E6	8D	A2	5（信息交换
00000120:	E7	94	A8	E6	B1	89	E5	AD	97	E7	BC	96	E7	A0	81	E5	用汉字编码字
00000130:	AD	97	E7	AC	A6	E9	9B	86	20	E8	BE	85	E5	8A	A9	E9	符集 辅助集
00000140:	9B	86	EF	BC	89	E4	B8	AD	E3	80	82	47	42	32	33	31	）中。GB231
00000150:	32	E6	98	AF	E7	AE	80	E5	8C	96	E5	AD	97	E9	9B	86	2是简化字集
00000160:	EF	BC	8C	47	42	31	32	33	34	35	E6	98	AF	E7	B9	81	，GB12345是繁
00000170:	E4	BD	93	E5	AD	97	E9	9B	86	EF	BC	8C	E4	B8	A4	E8	体字集，两者
00000180:	80	85	E7	9A	84	E7	9B	B8	E5	90	8C	E7	A0	81	E4	BD	的相同码位
00000190:	8D	E5	85	B7	E6	9C	89	E7	AE	80	E7	B9	81	E5	AF	B9	具有简繁对
000001A0:	E5	BA	94	E5	85	B3	E7	B3	BB	EF	BC	88	E4	B8	80	E5	应关系（一对

3.1.7 UTF8

中文 3 字节，英文数字 1 字节，UTF8。



1.txt 2.txt 3.txt 4.txt 5.txt 6.txt **7.txt** ×

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000:	EF	BC	91	EF	BC	99	EF	BC	98	EF	BC	90	E5	B9	B4	EF	１９８０年，
00000010:	BC	8C	E6	88	91	E5	9B	BD	E5	8F	91	E5	B8	83	E5	9B	我国发布国
00000020:	BD	E5	AE	B6	E6	A0	87	E5	87	86	47	42	2F	54	20	32	家标准GB/T 2
00000030:	33	31	31	EF	BC	88	E4	BF	A1	E6	81	AF	E6	8A	80	E6	311（信息技术
00000040:	9C	AF	20	E5	AD	97	E7	AC	A6	E4	BB	A3	E7	A0	81	E7	字符代码结
00000050:	BB	93	E6	9E	84	E4	B8	8E	E6	89	A9	E5	85	85	E6	8A	构与扩充技
00000060:	80	E6	9C	AF	EF	BC	89	EF	BC	8C	E8	AF	A5	E6	A0	87	术），该标
00000070:	E5	87	86	E7	AD	89	E5	90	8C	E4	BA	8E	E5	9B	BD	E9	准等同于国际
00000080:	99	85	E6	A0	87	E5	87	86	49	53	4F	2F	49	45	43	20	标准ISO/IEC
00000090:	32	30	32	32	E3	80	82	E8	80	8C	E8	BF	99	E4	B8	AA	2022。而这个
000000A0:	E4	BD	93	E7	B3	BB	E4	B8	8B	E6	B1	89	E5	AD	97	E5	体系下汉字字
000000B0:	AD	97	E7	AC	A6	E9	9B	86	E7	9A	84	E5	AE	9A	E4	B9	符集的定义
000000C0:	89	E5	88	99	E6	98	AF	E5	9C	A8	47	42	32	33	31	32	则是在GB2312
000000D0:	EF	BC	88	E4	BF	A1	E6	81	AF	E4	BA	A4	E6	8D	A2	E7	（信息交换用
000000E0:	94	A8	E6	B1	89	E5	AD	97	E7	BC	96	E7	A0	81	E5	AD	汉字编码字
000000F0:	97	E7	AC	A6	E9	9B	86	20	E5	9F	BA	E6	9C	AC	E9	9B	符集基本集
00000100:	86	EF	BC	89	E5	92	8C	47	42	31	32	33	34	35	EF	BC	）和GB12345（
00000110:	88	E4	BF	A1	E6	81	AF	E4	BA	A4	E6	8D	A2	E7	94	A8	信息交换用
00000120:	E6	B1	89	E5	AD	97	E7	BC	96	E7	A0	81	E5	AD	97	E7	汉字编码字符
00000130:	AC	A6	E9	9B	86	20	E8	BE	85	E5	8A	A9	E9	9B	86	EF	集辅助集）
00000140:	BC	89	E4	B8	AD	E3	80	82	47	42	32	33	31	32	E6	98	中。GB2312是
00000150:	AF	E7	AE	80	E5	8C	96	E5	AD	97	E9	9B	86	EF	BC	8C	简化字集，
00000160:	47	42	31	32	33	34	35	E6	98	AF	E7	B9	81	E4	BD	93	GB12345是繁体
00000170:	E5	AD	97	E9	9B	86	EF	BC	8C	E4	B8	A4	E8	80	85	E7	字集，两者的
00000180:	9A	84	E7	9B	B8	E5	90	8C	E7	A0	81	E4	BD	8D	E5	85	相同码位具
00000190:	B7	E6	9C	89	E7	AE	80	E7	B9	81	E5	AF	B9	E5	BA	94	有简繁对应
000001A0:	E5	85	B3	E7	B3	BB	EF	BC	88	E4	B8	80	E5	AF	B9	E5	关系（一对多
000001B0:	A4	9A	E6	83	85	E5	86	B5	E5	8F	A6	E5	A4	96	E7	BC	情况另外编

3.2 分词方法

3.2.1 基于词典

3.2.1.1 正向最大匹配

从前往后，每次匹配最大的词。

3.2.1.2 逆向最大匹配

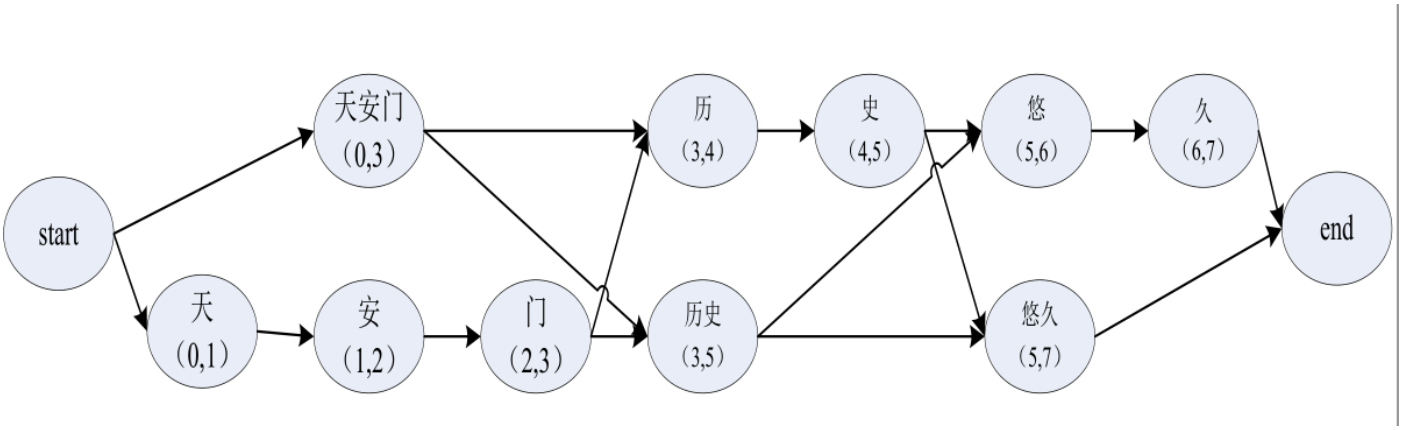
从后往前，每次匹配最大的词。

3.2.1.3 最短路径匹配

分词数最少。

3.2.1.4 全切分

找以每个字为首字的所有串，构造有向无环图 DAG。



3.2.1.5 最大概率分词

利用全切分的 DAG。每一条路径代表一种方案。

3.2.1.5.1 有向边概率

W_i到W_j的转移概率P_{ij}

3.2.1.5.2 路径概率

该分词方案下的句子生成概率，可近似估值为有向边的联合概率：

$$P(s) = \prod_{i=1}^{n} P(w_i|w_{i-1})$$

3.2.1.5.3 最佳路径

具有极大似然估计值的路径，即最大概率路径。此时的节点序列就是最优分词方案$W^{\#}$

$$W^{\#} = arg\max_{w} \prod_{i=1}^{n} P(w_i|w_{i-1})$$

3.2.1.5.4 二元语法建模

分词词典包含 n 个词条，建立$n \times n$的一阶马尔可夫转移概率矩阵$P = (P_{ij})$。

i, j 对应分词词典的第 i, j 个词条

$$P_{i,j} = P(w_j|w_i) = \frac{w_i, w_j 同时出现的次数}{w_i 出现的次数}$$

3.2.1.5.4.1 代价函数

将概率相乘转化为对数相加：

$$Cost(w_2|w_1) = -\log_2 P(w_2|w_1)$$

$$W^{\#} = arg\max_{w} \sum_{i=1}^{n} Cost(w_i|w_{i-1})$$

3.2.1.5.4.2 加权最短路径

Cost 为 DAG 边权。Dijkstra 算法。

3.2.2 启发式规则

对交集型歧义字段分别处理。

3.2.3 未登录词

- 尽量多地收集词汇
- 统计方法猜测
- 构词规则、上下文
- 分类处理

3.3 词性标注

3.3.1 CLAWS 算法

给定词序列 W，推断标记序列 T。

$$\arg\max_T P(T|W)$$

贝叶斯定理：$P(\mathrm{T}|\mathrm{W}) = \dfrac{P(T)P(W|T)}{P(W)}$

词序列已知，P(W)是常数。

$$\arg\max_T P(T|W) = \arg\max_T P(T)P(W|T)$$

所以，

$$\mathrm{T}^{\#} = \arg\max_T \prod_{i=1}^{n} P(t_i|t_{i-1})RTP(w_i, t_i)$$

3.3.1.1 转移概率矩阵

$$\mathrm{P_{i,j}} = P(t_j|t_i) = \frac{t_i与t_j同时出现次数}{t_i出现次数}$$

3.3.1.2 相对标注概率

$$\mathrm{RTP}(\mathrm{w_i}, t_i) = P(t_i|w_i) = \frac{w_i标记为t_i的次数}{w_i出现的次数}$$

3.3.2 隐马尔可夫模型（HMM: Hidden Markov Model）

给定词序列 W，推断标记序列 T。

$$\arg\max_T P(T|W)$$

$$\mathrm{P(T|W)} = \frac{P(T,W)}{P(W)} = \frac{P(T)P(W|T)}{P(W)} \approx P(T)P(W|T)$$

其中,

$$\mathrm{P(T)} = \mathrm{P(t_1}|t_0)P(t_2|t_1,t_0)\dots P(t_i|t_{i-1},t_{i-2},\dots) \approx P(t_1|t_0)P(t_2|t_1)\dots P(t_i|t_{i-1})$$

$$\mathrm{P(t_i}|t_{i-1}) = \frac{语料中t_i出现在t_{i-1}之后的次数}{语料中t_{i-1}出现次数}$$

$$\mathrm{P(W|T)} = \mathrm{P(w_1}|t_1)P(w_2|t_2,t_1,w_2,w_1)\dots P(w_i|t_i,t_{i-1},\dots t_1,w_i,w_{i-1},\dots,w_1) \approx P(w_1|t_1)P(w_2|t_2)\dots P(w_i|t_i)$$

$$\mathrm{P(w_i}|t_i) = \frac{语料中w_i词性被标记为t_i的次数}{语料中t_i出现的次数}$$

3.3.2.1 与 CLAWS 的不同点

3.3.2.1.1 CLAWS 的发射概率

$$\mathrm{RTP(w_i},t_i) = P(t_i|w_i) = \frac{w_i标记为t_i的次数}{w_i出现的次数}$$

词性相对该词的概率。

3.3.2.1.2 HMM 的发射概率

$$\mathrm{P(w_i}|t_i) = \frac{语料中w_i词性被标记为t_i的次数}{语料中t_i出现的次数}$$

该词相对当前状态的词性的概率。

3.3.3 基于规则

- 按兼类词搭配关系
- 按词语结构

3.4 中文命名实体识别

3.4.1 层叠隐马尔可夫模型

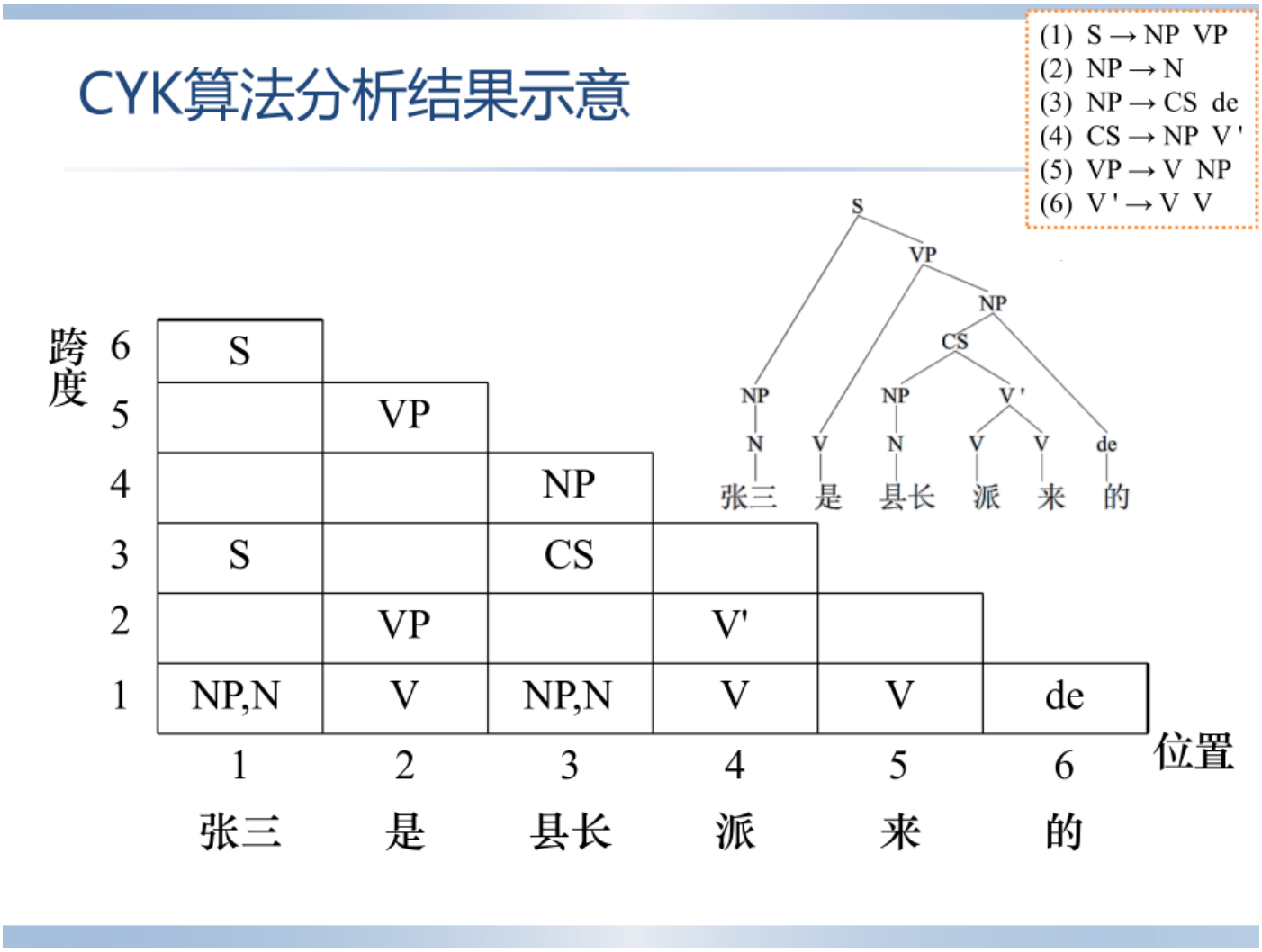
$$\mathrm{P(w_i}|t_i) = \frac{w_i作为角色t_i出现的次数}{角色t_i出现的次数}$$

$$\mathrm{P(t_i}|t_{i-1}) = \frac{角色t_{i-1}下一个角色是t_i出现的次数}{角色t_i出现的次数}$$

3.5 CFG（CONTEXT-FREE GRAMMAR）分析

3.5.1 CYK 算法

产生式全部转换为 Chomsky 范式，即只含 A->BC A->x。



跨度为 1 的一行即原词性，从跨度 2 开始填。

位置i，跨度j，则看i~(i + j)的词性是否可以规约，如可以，则填空规约后的非终结符。

3.5.2 PCFG（Probabilistic CFG）概率句法分析

选择概率最大的句法分析树。

3.5.3 依存语法

以词为节点。

- 分布策略
  - 先分析依存骨架树（依存概率、动态规划）
  - 再判别每条依存弧的关系类型（多元分类）
- 一体化策略

- 同时确定依存骨架及其关系类型（启发式算法）

3.6 文本分析

3.6.1 特征选择

3.6.1.1 文档频率法（DF: Document Frequency）

$$\mathrm{DF(t)} = \frac{dc(t)}{|d|}$$

特征词 t 出现的文本数占文档总数的频率。

3.6.1.2 信息增益法（IG: Information Gain）

$$\mathrm{IG(T)} = \mathrm{H(C)} - \mathrm{H(C|T)}$$

3.6.1.3 卡方检验法

$$\chi^2 = \frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(C+D)}$$

3.6.2 权重计算

3.6.2.1 布尔取值

$$\mathrm{w_{i,j}} = \begin{cases} 1, & t_j \text{在文档} d_i \text{出现} \\ 0, & t_j \text{未在文档} d_i \text{出现} \end{cases}$$

3.6.2.2 特征词频（TF: Term Frequency）

$$\mathrm{w_{i,j}} = tf_{i,j}$$

特征在文档中出现的次数。

3.6.2.3 倒排文档频（IDF: Inverse Document Frequency）

$$\mathrm{w_{i,j}} = \log_2 \frac{N}{n_j}$$

N是文档集中的文档总数，$\mathrm{n_j}$是包含词$\mathrm{t_j}$的文档数。

3.6.2.4 TF-IDF

$$\mathrm{w_{i,j}} = TF \cdot IDF(d_i, t_j) = tf_{ij} \times \log_2 \frac{N}{n_j}$$

权重与特征的频次成正比，与在整个文档中出现的文档数目成反比。

3.6.3 简单 TF-IDF 算法

- 用向量空间模型表示文本，采用 TF-IDF 法计算特征词权重，在此基础上进行类内叠加。
- 具体分类时，用余弦法计算文本向量。