预处理

原始文本 → 清洗 → 标准化 → 分词

网页
新闻
报告

无用标签
特殊符号
停用词

词干提取
词形还原
不定词处理
敏感词处理

数据收集

1、语料库：存放语言材料的数据库

类型：专用&通用 单语&多语 共时&历时 生语&熟语

2、词汇知识库(WordNet)：一个按语义关系网络组织的巨大语库

以同义词集合(sysnet)作为基本构建单位，将名词、动词、形容词、副词都组织到sysnet中，一个sysnet只包含一个注释，对一个sysnet中不同的词，分别用造当的例句加以区分

如：car auto automobile 都属于"汽车"这个sysnet

WordNet中的语义关系及标记符号

名词		动词		形容词		副词	
反义关系 Antonym	!	反义关系 Antonym	!	反义关系 Antonym	!	反义关系 Antonym	!
下位关系 Hyponym	~	下位关系 Troponym	~	近义关系 Similar	&	导出形式 Derived from	\
上位关系 Hypemym	@	上位关系 Hypemym	@	关系型形容词 Relational Adj.	\		
部分关系 Meronym	#	蕴涵关系 Entailment	*	又见 Also See	^		
整体关系 Holonym	%	致使关系 Cause	>	属性 Attribute	=		
属性 Attribute	=	又见 Also See	^				

上下位关系：上位：抽象包含性概念 如：animal ⟷ dog

下位：具体概念

整体部分关系：car ⟷ wheel

3、数据爬虫

一种自动提取网页的程序，传统爬虫从一个或若干初始网页的URL开始，获得初始网页上的URL，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队列，直到满足一定条件

步骤：① 获取网页：根据URL获取HTML数据

② 解析网页：解析HTML，获取HTML目标信息

③ 存储数据

西文字符编码

ASCII码（美国标准信息交换码）

ASCII可显示字符（共95个）

二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形
0010 0000	32	20	(space)	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(	0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29	)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	-	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s
0011 0100	52	34	4	0101 0100	84	54	T	0111 0100	116	74	t
0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u
0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	v
0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	w
0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	x
0011 1001	57	39	9	0101 1001	89	59	Y	0111 1001	121	79	y
0011 1010	58	3A	:	0101 1010	90	5A	Z	0111 1010	122	7A	z
0011 1011	59	3B	;	0101 1011	91	5B	[	0111 1011	123	7B	{
0011 1100	60	3C	<	0101 1100	92	5C	\	0111 1100	124	7C	|
0011 1101	61	3D	=	0101 1101	93	5D	]	0111 1101	125	7D	}
0011 1110	62	3E	>	0101 1110	94	5E	^	0111 1110	126	7E	~
0011 1111	63	3F	?	0101 1111	95	5F	_				

中文字符编码

GB2312-80：1980年发行的国标码，覆盖常用简体字

GBK：1995年发行，覆盖生僻字、繁体字，windows默认中文编码，国标扩展码

GB18030：覆盖少数民族文字，中国官方强制标准

GB13000/Unicode/ISO 10646：国际通用字符集标准

GB2312-80：用两个字节表示一个汉字，每个字节的ASCII码都在161(A1)～254(FF)间，共$94^2=8836$个码位，共7445个字符(汉字6763个)，1391个空位

首字节ASCII码	
161	标点、一般符号（202） 序号（60） 数字（22）、拉丁字母（52）
162	
163	
164	日文假名（169）
165	
166	希腊字母（48）
167	俄文字母（66）
168	汉语拼音符号（26） 汉语注音字母（37）
176~215	一级汉字（3755个） 按汉语拼音顺序排序
216~247	二级汉字（3008个） 按部首排列

一个字在方阵中的坐标称为其区位码

一级汉字：按拼音字母排序，同音字按笔形顺序排序（横直撇点折）

二级汉字：按部首顺序排序
同部首按笔划数排序
同笔划数按笔形顺序排序

Unicode：可视作ISO/IEC 10646标准的实践版

四维编码空间：
一个完整的UCS-4编码 = 4个8位字节，记作：
B_1（最高字节）B_2（次高字节）B_3（第3字节）B_4（最低字节）

- 根据最高位为0的最高字节分为$2^7=128$组（group）
  即：B_1 0xxxxxxx
- 每个组再根据次高字节B_2分为$2^8=256$个平面（plane）
- 每个平面根据第3字节B_3分为$2^8=256$行（row）
- 每行有2^8个码位（cell）

group 0的plane 0被称作BMP，是目前最实用的Unicode版本，剩下2字节（2^{16}）用于给字符编码。

Unicode的实现方式不同于编码方式，因为一个字符的Unicode编码为4字节，会浪费大量空间。

Unicode的实现方式称为Unicode转换格式(UTF)

UTF-8：以元符为8位整数作编码单位

类似的有UTF-16、UTF-32（32就是Unicode编码）

GBK：兼容GB2312与Unicode

第一字节ASCII码：129~254

第二字节ASCII码：64~254（127除外）

字符编辑距离

定义：在两个单词(w_1, w_2)间由其中一个单词w_1转换为另一个单词w_2所需的最少单字符编辑操作次数（插入，删除，替换）

英文字符串编辑距离：动态规划

字符串$a[0\cdots m]$与$b[0\cdots n]$，则d_{ij}表示将$a[0\cdots i-1]$转换为$b[0\cdots j-1]$的最短编辑距离

递推公式：

$$d_{ij}=\begin{cases} d_{i-1,j-1} & a_{i-1}=b_{j-1} \\ \min\begin{cases} d_{i-1,j}+cost\_del & 删a[i-1] \\ d_{i,j-1}+cost\_ins & 插b[j-1] \\ d_{i-1,j-1}+cost\_rep & 替 \end{cases} & a_{i-1}\neq b_{j-1} \end{cases}$$

汉字编辑距离：由汉字笔顺的数字编码得来，如"狼"与"狠"

"狼"：3534511534　"狠"：353511534

即该数字串编辑距离

正则表达式

通常用于检索、替换符合某个模式（规则）的文本

1、普通字符：直接匹配自身，如"123"匹配"123"

2、元字符

. ：匹配任意单个字符　如"a.b"匹配"axb""a#b""a1b"

^ ：匹配字符串开头(多行模式下匹配每行开头)

如：^hello 匹配"hello world"但不匹配"1hello"

$ ：匹配字符串结尾(或每行结尾)

如：world$ 匹配"hello world"

* ：匹配前面的元素0或很多次

如：ab* 匹配 a 、ab、abb …

+ ：匹配前面的元素1或很多次

如：ab* 匹配ab、abb、…

? ：匹配前面的元素0或1次

如：ab? 匹配a和ab

{n} ：匹配前面的元素n次(n为非负整数)

如：a{3} 匹配 aaa

{n,} ：匹配前面的元素至少n次

如：a{2,} 匹配 aa 、aaa 、…

{n,m} ：匹配前面的元素n到m次

如：a{2,3} 匹配 aa、aaa

| ：“或”，优先级最低的元字符，匹配左边或右边的模式

如：ab|cd 匹配ab或cd

a(b|c)d 匹配 abd或acd

( ) ：分组，将多个字符看作一个整体，同时捕获匹配结果
(可以通过反向引用使用)

分组：(ab)+ 匹配ab，abab …

捕获与引用：(\d)\1　　\d匹配一个数字，捕获为\1

\1再使用一次刚捕获的数字

匹配 11 、22 、33 …

\：转义字符，取消元字符的特殊含义，或表示预定义字符

如：a\.b 匹配 a.b

[]：字符类，匹配括号内任意字符，支持范围表示（[A-Z]，[0-9]），加上^表否定

如：[abc]匹配 a、b、c

[^abc]匹配任意非 abc 的字符

[-]：字符类内的范围符，表示连续字符集

如[a-z][0-9][a-zA-Z0-9]

3、预定义字符类

\d：等价于[0-9]

\D：等价于[^0-9]

\w：等价于[a-zA-Z0-9_]（注意这里有个下划线）

\W：等价于[^a-zA-Z0-9_]

\s：等价于[\t\n\r\f\v] 匹配任意空白字符

\S：等价于[^\t\n\r\f\v]

\b：匹配单词边界（单词与非单词分隔处，如hello world中间空格两侧）

如：\bcat\b 匹配 cat

\B：匹配非单词边界

如：cat\B 不匹配 cat，但会匹配catch等

\n：匹配换行符（部分语言支持匹配\r\n）

\t：匹配制表符

核心特性

1. 贪婪匹配

默认情况下，正则的量词（* + ? {n,m}）是贪婪的，即总是尽可能匹配更多字符，在量词后面加？则反之

如：匹配字符串"aabbaabcc"

a.*b：匹配aabbaab

a.*?b：匹配aab

a.+c：匹配aabbaabcc

a.+?c：匹配aabbaabc

2、分组与捕获

前面说()会捕获匹配到的内容，捕获的内容按捕获顺序(\1，\2，…)编号，可以被反向引用

如：(\w+)\s+\1：()会捕获\w+匹配的字符串(这里是一个单词)并存入\1，\s+是空白字符，\1即引用\1里存的字符串，如："hello hello"

(\d{4})-(\d{2})-(\d{2})：匹配2026-04-07

\1存"2026" \2存"04" \3存"07"

捕获自然会提升机器负担，若不要捕获但要分组，可以用：(?:)，如：(?:ab)

如果要为捕获组命名，则用(?<name>)，但只有部分语言支持(Py，JS，Java)

3. 零宽断言

①先行断言：判断当前位置后面是否满足条件

·前向先行：(?=pattern) 后面必须是pattern

·负向先行：(?!pattern) 后面必须不是pattern

如：[a-z](?=\d) 匹配后面跟着数字的字母"a1"的"a"

^(?=.*\d).{6,12}$：这是密码的正则表达式，要求这个密码长度在6~12个字符之间，必须包含一个数字。"^"强制匹配从字符串开头开始；(?=.*\d)表示至少包含一个数字的字符串，.{6,12}规定匹配长度在6~12之间，$标记字符串结尾

注：.*\d若直接作为正则表达式，那它会匹配"最长的以数字结尾的字符串"，如"a1b2c"里的"a1b2"
但是作为前向先行里的pattern，它就稍微不一样了，它的意思就变成了只要.*\d能去匹配这个pattern就说明条件成立，如pattern = ab3cd，.*\d能匹配ab3，则条件成立。换句话说只要pattern含数字就行

② 后行断言：判断当前位置前面是否满足条件
- 正向后行：(?<=pattern) 前面必须是pattern
- 负向后行：(?<!pattern) 前面必须不是pattern

如：(?<=\d)[a-z] 前面是数字的字母 如"1a"的"a"

常用组合

.*：匹配任意不换行字符串

[\s\S]：匹配任意字符串

[^x]*：匹配到x为止，不包含x

[^"]*：匹配双引号内的内容

\b\w+\b：匹配任意完整单词

\S+：匹配一串非空字符

Pattern	Matches	
0\d{2,3}-\d{7,8}	电话号码	0785-2557392
^1[3|4|5|7|8]\d{9}$	手机号码	15602527361
(^\d{15}$)\|(^\d{17}(\d\|X\|x)$)	身份证号码	44063719971204258X
^(\w+)@([0-9a-zA-Z]+)(\.[a-zA-Z]{2,4}){1,2}$	电子邮件	694431056@qq.com
^[\u4e00-\u9fa5]{0,}$	检验汉字	你我他
\n\s*\r	空白行	
[1-9][0-9]{4,}	QQ号	从10000开始

Pattern	Matches
网址（URL）	[a-zA-z]+://[^\s]*
IP地址(IP Address)	((2[0-4]\d|25[0-5]|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5]|[01]?\d\d?)
日期(年-月-日)	(\d{4}|\d{2})-((1[0-2])|(0?[1-9]))-(([12][0-9])|(3[01])|(0?[1-9]))
日期(月/日/年)	((1[0-2])|(0?[1-9]))/(([12][0-9])|(3[01])|(0?[1-9]))/(\d{4}|\d{2})
时间(小时:分钟, 24小时制)	((1|0?)[0-9]|2[0-3]):([0-5][0-9])
中文及全角标点符号(字符)	[\u3000-\u301e\ufe10-\ufe19\ufe30-\ufe44\ufe50-\ufe6b\uff01-\uffee]
中国大陆邮政编码	[1-9]\d{5}
整数	-?\d+
小数	(-?\d+)(\.\d+)?
不包含abc的单词	\b((?!abc)\w)+\b