第二讲 - 文本数据获取与预处理

张建章

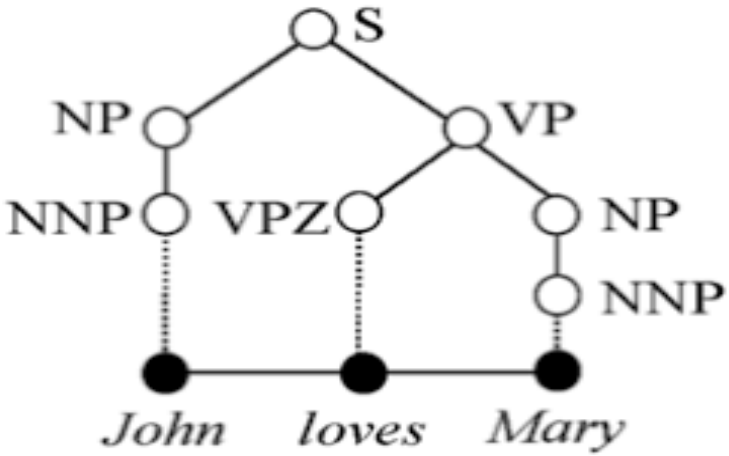
阿里巴巴商学院
杭州师范大学

2024-03-01

目录

精标语料库：语言和计算机专业人员依据严谨、详细的标注标准，使用规范的原始文本，在人工分析、理解的基础上制作而成的高质量语料库，为自然语言处理算法的发展奠定了高质量的数据基础。最具代表性的中英文精标语料库分别为人民日报语料库和宾州树库。

人民日报语料库 (新版)



Penn Treebank 3.0

这类语料库通常需要申请使用，可免费用于科研工作。

粗标语料库：大规模、粗粒度标注语料库，原始语料的广泛的来源有助于提高算法泛化性。

亚马逊商品评论数据集

千言数据集

这类语料包含的领域广泛，对应的 NLP 任务粒度相对较粗。既包含规范文本，也包含不规范文本，如，微博，论坛等。语料所用的原始文本年份较新。通常可免费下载，部分数据集需要注册申请下载，免费用于科研目的。

获取公开语料库的常见途径

- 科研论文公开的数据集，通常公布在 Github；
- 各类竞赛使用的数据集，通常公布在竞赛官网，如 Kaggle、阿里云天池、中文信息学会竞赛、SemEval；
- 互联网公司公开的大规模数据集，通常在公司的官网，可申请免费用于科研用途；
- 研究机构公开的数据集，通常在科研院所的主页中数据栏目下；
- 个人研究者整理收集的数据集，通常在个人 Github 或博客中，权威性较低。

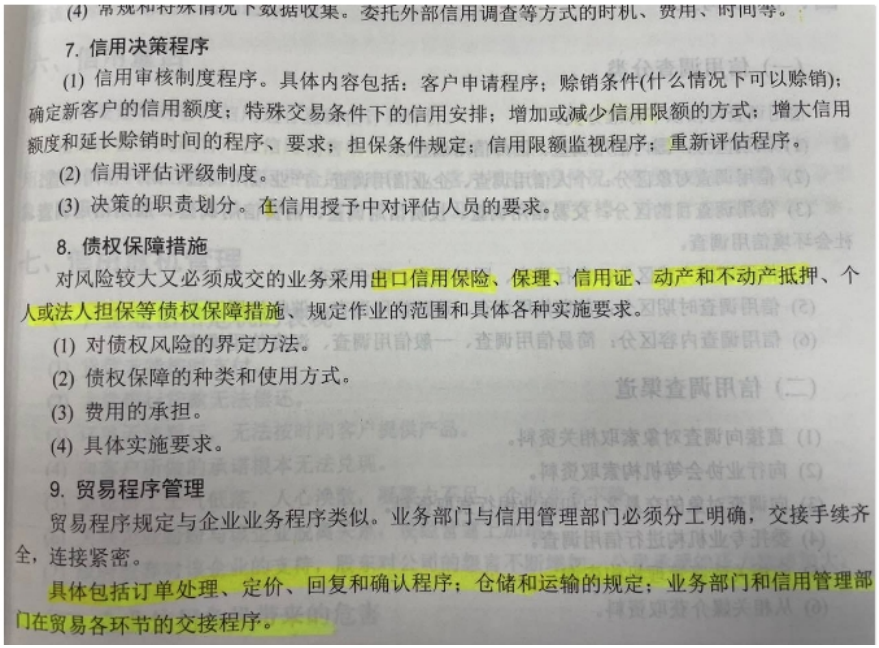
自建语料库：根据任务需求，收集语料进行标注，以训练针对特定任务需求的模型。相比使用公开的通用领域语料或相关领域预料，在自建语料库上训练的模型对解决特定任务具有更好的性能。自建语料库的常见步骤如下：

- 从特定渠道收集原始数据；
- 制定规范统一的标注手册；
- 培训标注人员熟悉标注规范和标注工具的使用；
- 标注数据，众包形式或内部团队协作，需要一定的激励机制以确保标注结果的质量；
- 标注结果质量评估及完善修改。

关于自建语料库的更多细节，可参考《面向机器学习的自然语言标注》。

纸媒文本电子化

借助光学字符自动识别技术 (OCR)，将扫描版或图片版纸媒内容转化为电子文本内容。



(4) 常规和特殊情况下数据收集。委托外部信用调查等方式的时机、费用、时间等。

7. 信用决策程序

(1) 信用审核制度程序。具体内容包括：客户申请程序；赊销条件(什么情况下可以赊销)；确定新客户的信用额度；特殊交易条件下的信用安排；增加或减少信用限额的方式；增大信用额度和延长赊销时间的程序、要求；担保条件规定；信用限额监视程序；重新评估程序。

(2) 信用评估评级制度。

(3) 决策的职责划分。在信用授予中对评估人员的要求。

8. 债权保障措施

对风险较大又必须成交的业务采用出口信用保险、保理、信用证、动产和不动产抵押、个人或法人担保等债权保障措施、规定作业的范围和具体各种实施要求。

(1) 对债权风险的界定方法。

(2) 债权保障的种类和使用方式。

(3) 费用的承担。

(4) 具体实施要求。

9. 贸易程序管理

贸易程序规定与企业业务程序类似。业务部门与信用管理部门必须分工明确，交接手续齐全，连接紧密。

具体包括订单处理、定价、回复和确认程序；仓储和运输的规定；业务部门和信用管理部门在贸易各环节的交接程序。

(4) 常规和特殊情况下数据收集。委托外部信用调查等方式的时机、费用、时间等。7．信用决策程序

(1) 信用审核制度程序。具体内容包括：客户申请程序；赊销条件（什么情况下可以赊销）；确定新客户的信用额度；特殊交易条件下的信用安排；增加或减少信用限额的方式；增大信用额度和延长赊销时间的程序、要求；担保条件规定；信用限额监视程序；重新评估程序。

(2) 信用评估评级制度。

(3) 决策的职责划分。在信用授予中对评估人员的要求。8．债权保障措施

对风险较大又必须成交的业务采用出口信用保险、保理、信用证、动产和不动产抵押、个人或法人担保等债权保障措施、规定作业的范围和具体各种实施要求。

(1) 对债权风险的界定方法。

(2) 债权保障的种类和使用方式。 (3) 费用的承担。

(4) 具体实施要求。9．贸易程序管理

贸易程序规定与企业业务程序类似。业务部门与信用管理部门必须分工明确，交接手续齐全，连接紧密。

具体包括订单处理、定价、回复和确认程序；仓储和运输的规定；业务部门和信用管理部门在贸易各环节的交接程序。

图 1: OCR 示例 (在线工具白描 OCR)

OCR 技术目前的准确率已经非常高，已达到实用阶段，在票据自动识别和录入、书籍电子化等领域广泛应用。

互联网文本自动收集

借助网络爬虫技术已自动化方式获取和存储互联网上丰富的文本语料。应用爬虫技术大规模获取数据需要遵守相关法律法规，否则可能会涉嫌以非法方式入侵计算机系统罪。

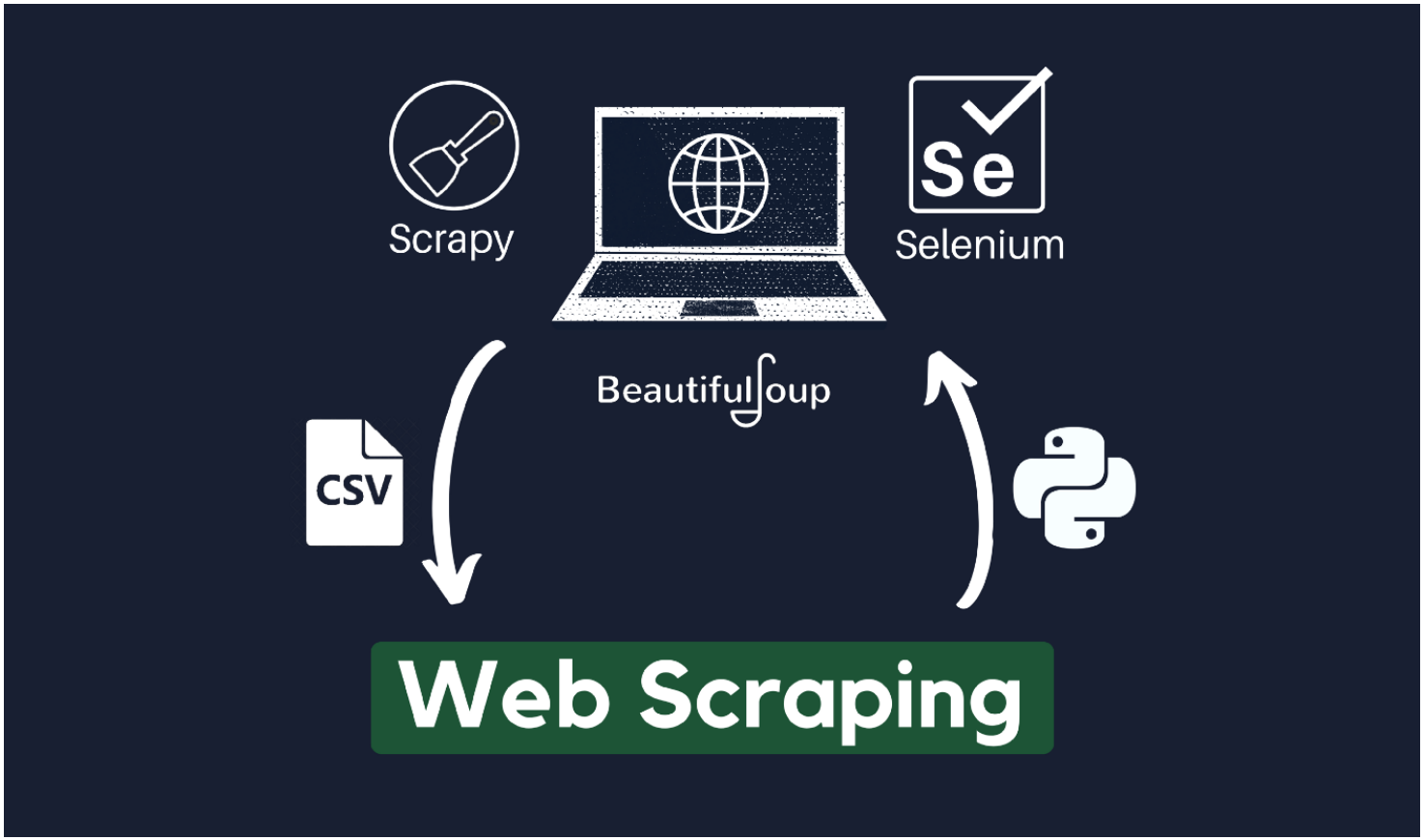


图 2: 网络爬虫

问题介绍

APP 隐私政策自动收集

以百度应用市场为数据源，获取网络购物类别 APP 的隐私政策，将隐私政策保存到本地 html 文件中。

隐私政策

文件保存

示例代码

```
import requests
# 获取APP隐私政策网址
url = 'https://mobile.baidu.com/api/board?boardid=board_101_0316
↪  &boardId=34612&pn=0'
response = requests.get(url)
cont_str = response.content.decode('UTF-8')
cont_dict = eval(cont_str.replace('true','True'))
data = cont_dict['data']['data']
privacy_url = data[0]['privacyUrl']
# 根据隐私政策网址，访问并保存隐私政策到本地html文件
response = requests.get(privacy_url)
html = response.content.decode('UTF-8')
with open('./demo.html','w+') as f:
  f.write(html)
```

从混杂着网页源代码、多媒体等“噪音”的原始语料中抽取出纯文本内容 (plain text)。下面借助 `lxml` 库的 `Cleaner` 类清洗标签。

```
# 导入所需包
from bs4 import BeautifulSoup
from lxml.html.clean import Cleaner
# 从本地导入html文件
with open('./app_privacy/com.xunmeng.pinduoduo.html') as f:
  html = f.read()

def remove_invisible_chars(text):
  """移除所有不可见字符，除\r\t\n和空格外"""
  result = ''
  for char in text:
    if char in ['\r','\t','\n',' '] or char.isprintable():
      result += char
  return result
```

```
print(html)

# <!DOCTYPE html><html
↪  lang="zh-Han.....我们可能会收集您的相关信息，
↪  为.....d'));</script></body></html>
cleaner = Cleaner(style=True,scripts=True,javascript=True,)

html_notag = cleaner.clean_html(html)
soup = BeautifulSoup(html_notag, 'html.parser')
raw_text = soup.get_text()
clean_text = remove_invisible_chars(raw_text)
print(clean_text[:100] + ' ... ... ' + clean_text[-100:])

# 拼多多隐私政策拼多多隐私政策（V3.4.1）更新日期:
↪  2023年2月3日特别提示：拼多多（以下或称“我们”）非常注重保护用户
↪  （以下或称“您”）的个人信息.....包括Pinduoduo
↪  Inc最新上市公司年报披露的拼多多服务提供者的关联公司。\n
```

将文档内容以句子为单位进行分割。传统方法基于规则，根据标点符号、词性、词语的位置等信息来判断句子的结束。但是，此方法比较依赖于语言的特定规则，难以应对不同语境下的复杂情况和不规范文本。

```
import re
sample_doc = '''您可以通过我们为您提供的评价、
↪  拼小圈分享及其他信息发布功能，选择发表评价，
↪  公开发布图文/视频以及分享购买过的商品/服务的信息。
↪  我们尊重并保护您对相关信息是否发布以及发布范围的设置。'''
# 划分句子并保留标点
re.split('(。|？|！)', sample_doc)
'''['您可以通过我们为您提供的评价、拼小圈分享及其他信息发布功能，
↪  选择发表评价，公开发布图文/视频以及分享购买过的商品/服务的信息',
'。',
'我们尊重并保护您对相关信息是否发布以及发布范围的设置',
'。',
'']'''
```

常见的复杂情况和不规范文本的例子如下：

不规范文本：我家养了一只名叫“小黄”的猫，因为它很可爱，所以经常被我们抱起来玩。

社交媒体文本：# 购物日记 # 今天去逛街了，买了一堆东西，感觉好开心！

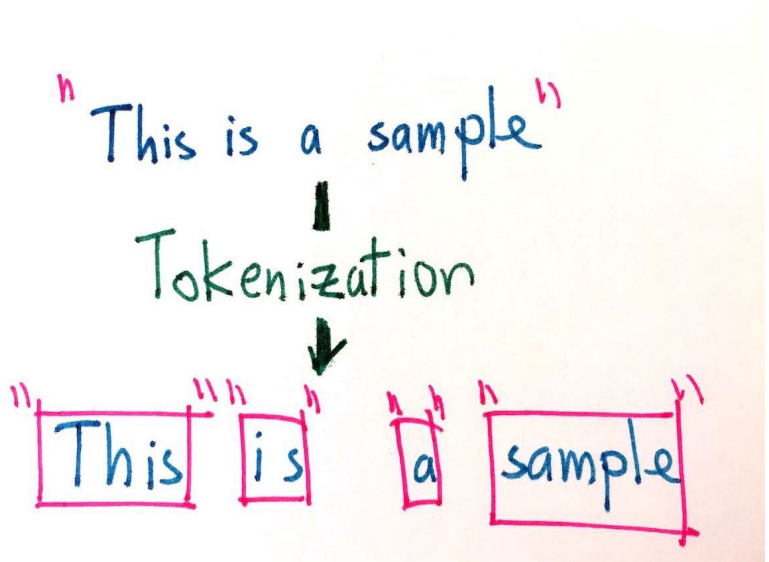
缩略词：I'm sorry, but I can't make it to the 2 p.m. meeting tomorrow.

文内引用：据该公司发言人表示：“我们真的没有回避！相反，我们正在积极解决客户投诉，收集产品问题。”

英文中还存在复杂的从句嵌套结构等较难处理的情况。通常，各类 NLP 工具中包含了较为详尽的分句规则，能够正确处理多数情形。

将文档以词语为基本单位进行分割，既可以在分句后进行，也可以不分句直接分词。中文等东亚语言文本 (中日韩文本) 没有明确的词汇边界，分词是一项比较困难的任务，英文等拉丁语系，词语间的空格可作为天然分隔符，正确处理大多数分词场景。

牙牙学语 (从词开始)



分词示例

文本分词示例

```
import jieba
from nltk import word_tokenize
eraw = """'When I'M a Duchess,' she said to herself, (not in a
↪  very hopeful tone though), 'I won't have any pepper in my
↪  kitchen AT ALL. Soup does very well without--Maybe it's
↪  always pepper that makes people hot-tempered,'..."""

craw = '我们尊重并保护您对相关信息是否发布以及发布范围的设置'

eraw.split()
print(word_tokenize(eraw))

for word in jieba.cut(craw):
  print(word)
```

为分词之后的每个词语分配一个词性标签，如动词、名词、代词等。常用的中文词性标签集合有 PKU、863、Universal Dependencies (详情)。

代码	名称	代码	名称	代码	名称	代码	名称
Ag	形语素	g	语素	ns	地名	u	助词
a	形容词	h	前接成分	nt	机构团体	Vg	动语素
ad	副形词	i	成语	nz	其他专名	v	动词
an	名形词	j	简称略语	o	拟声词	vd	副动词
b	区别词	k	后接成分	p	介词	vn	名动词
c	连词	l	习用语	q	量词	w	标点符号
Dg	副语素	m	数词	r	代词	x	非语素字
d	副词	Ng	名语素	s	处所词	y	语气词
e	叹词	n	名词	Tg	时语素	z	状态词
f	方位词	nr	人名	t	时间词		

图 3: PKU 词性标签含义

分词和词性标注既可以依次进行，也可以同时进行 (即联合建模两任务)。

常用的英文词性标签集合
有Penn Treebank 标签集和Universal Tag Set (粒度较粗)。

Tag	Meaning	English Examples
ADJ	adjective	new, good, high, special, big, local
ADP	adposition	on, of, at, with, by, into, under
ADV	adverb	really, already, still, early, now
CONJ	conjunction	and, or, but, if, while, although
DET	determiner, article	the, a, some, most, every, no, which
NOUN	noun	year, home, costs, time, Africa
NUM	numeral	twenty-four, fourth, 1991, 14:24
PRT	particle	at, on, out, over per, that, up, with
PRON	pronoun	he, their, her, its, my, I, us
VERB	verb	is, say, told, given, playing, would
.	punctuation marks	. , ; !
X	other	ersatz, esprit, dunno, gr8, univeristy

图 4: Universal Tag Set 词性标签含义

词性标注示例

```
import nltk
import jieba.posseg
from nltk import word_tokenize
eraw = """'When I'M a Duchess,' she said to herself, (not in a
↪  very hopeful tone though), 'I won't have any pepper in my
↪  kitchen AT ALL. Soup does very well without--Maybe it's
↪  always pepper that makes people hot-tempered,'..."""

craw = '我们尊重并保护您对相关信息是否发布以及发布范围的设置'

print(nltk.pos_tag(word_tokenize(raw)))

for word,pos in jieba.posseg.cut(craw):
  print(word,pos)
```

停用词：在一个文本中频繁出现但对理解文本语义贡献很小的词语。这些词通常是虚词、介词、连词、代词等，例如：“的”、“和”、“在”、“是”、“他” 等等。

除通用领域停用词表外，对于特定的应用领域，还可以添加领域停用词 (具体问题，具体分析)。例如，影评情感分析任务中，片名可能会频繁出现，但对情感极性判断的指示性较低，因此可能会被视为停用词。

```
# 整合常见中文停用词表, https://github.com/goto456/stopwords
with open('./stopwords.txt') as f:
  cstopwords = f.read().strip().split()

# 加载NLTK中的英文停用词表
from nltk.corpus import stopwords
estopwords = set(stopwords.words('english'))
```

中英文停用词过滤实例

```
eraw = """'When I'M a Duchess,' she said to herself, (not in a
↪  very hopeful tone though), 'I won't have any pepper in my
↪  kitchen AT ALL. Soup does very well without--Maybe it's
↪  always pepper that makes people hot-tempered,'..."""
craw = '我们尊重并保护您对相关信息是否发布以及发布范围的设置'

for word,pos in jieba.posseg.cut(craw):
  if word not in cstopwords:
    print(word,pos)

for item in nltk.pos_tag(word_tokenize(raw)):
  if item[0].lower() not in estopwords:
    print(item)
```

It's Your Turn: Improve the above codes to further remove punctuations.

正则表达式：由一系列字符和特殊字符组成的表达式，可以用来匹配字符串中的特定模式。在文本处理、数据清洗、数据抽取等任务中，正则表达式是一种非常有用的工具，可以大大提高处理文本数据的效率和准确度。

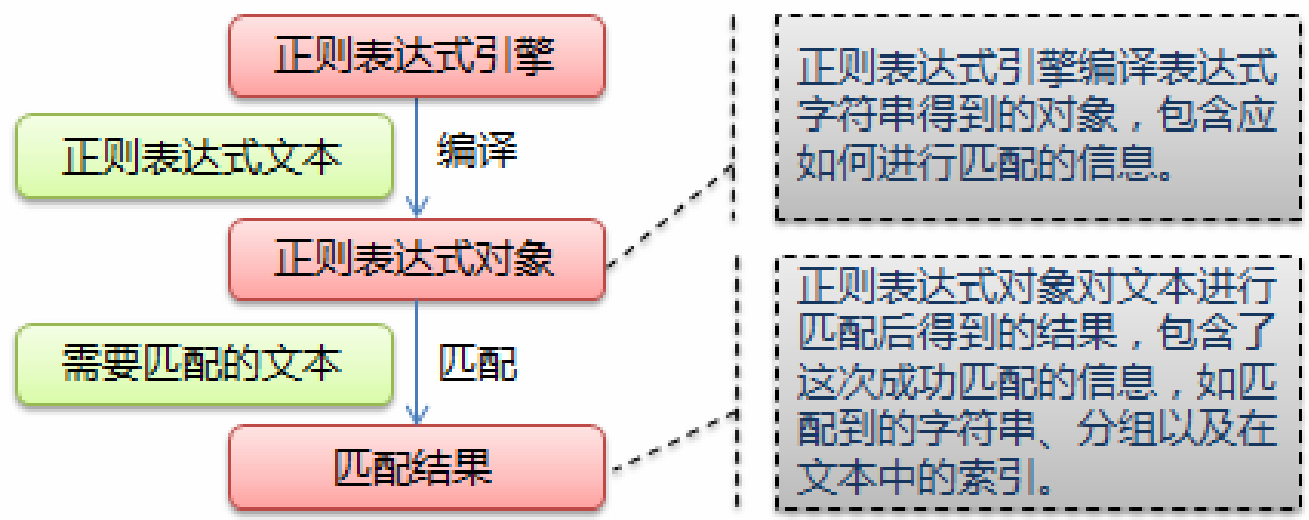


图 5: 正则表达式工作流程

Python 中使用正则表达式需要借助 `re` 包。虽然正则表达式功能强大，但也需要一定的学习成本 (相关资料，速查表)。

正则表达式实例

```
# 导入Python内置的re包
import re

raw = """'When I'M a Duchess,' she said to herself, (not in a
↪  very hopeful tone though), 'I won't have any pepper in my
↪  kitchen AT ALL. Soup does very well without--Maybe it's
↪  always pepper that makes people hot-tempered,'..."""

# 编译正则表达式，一次编译后可多次用于匹配，效率高
spliter = re.compile(r'\W+')

# 匹配，以非单词字符分割输入文本
re.split(spliter, raw)

# 编译+匹配，每次匹配都需要编译，效率低
re.split(r'\W+', raw)

# ['', 'When', 'I', ..... , 'tempered', '']
```

正则表达式实例

```
# 匹配数字
pattern1 = r'\d+'
text1 = '阿里巴巴商学院成立于2008年10月31日'
match1 = re.findall(pattern1, text1)
print(match1)  # ['2008', '10', '31']

# 匹配中文
match2 = re.findall(r'[\u4e00-\u9fa5]', text1)
print(match2) # ['阿', '里', ....., '日']
# 匹配邮箱用户名和域名
emails = ['Jhon@gmail.com', 'peng_zhang@hznu.edu.com',
↪  'jianzhang.zhang@foxmail.com']
for email in emails:
  match = re.match(r'^([\w\.]+)@([\w\.]{2,})$', email)
    if match:
      username = match.group(1)
      domain = match.group(2)
    print(f'Username: {username}, Domain: {domain}')
```

1. 读取给定的两个文件（2007 年和 2008 年政府工作报告）的内容，使用正则表达式识别两个文本内容中的社会消费品零售总额，并计算 2008 年比 2007 年增加多少（结果以百分比呈现）。

2. 复习程序设计基础课程中的字符串相关操作。

THE END