7



数据预处理：字符串及文体处理

授课教师：[李阳源]

SL02:今日学习目标



- 理解文本数据(Text Data)的特性和重要性
- 掌握自然语言处理(NLP, Natural Language Processing)的基本概念
- 学习文本预处理(Text Preprocessing)的主要技术
- 理解正则表达式(Regular Expression)的基本语法和应用方法
- 认识形态分析(Morphological Analysis)的概念和重要性

SL03: 什么是文本数据?



- 用自然语言(Natural Language)编写的非结构化数据(Unstructured Data)
- 例如:
  - 社交媒体帖子(Social Media Posts)
  - 新闻文章(News Articles)
  - 客户评论(Customer Reviews)
  - 电子邮件(Emails)
  - 聊天对话(Chat Conversations)

SL04: 文本数据的特点



- 非结构化(Unstructured) - 不是表格形式
- 噪声多(Noisy) - 拼写错误、缩写、特殊字符等
- 存在与语言相关的特性(Language-specific features)
- 上下文依赖(Context-dependent) - 相同的词语在不同上下文中意义可能变化
- 维度非常高(High-dimensional) - 存在无数词语和表达方式

SL05: 文本数据分析的重要性



- 互联网上80%以上的数据是文本形式
- 企业的重要数据大多以文档形式存在
- 对了解客户意见和市场趋势至关重要
- 可以进行自动化信息提取和分类
- 通过情感分析(Sentiment Analysis)了解舆论

SL06: 什么是自然语言处理(NLP)?



- 计算机理解和处理人类语言的技术
- 主要应用领域:
  - 机器翻译(Machine Translation)
  - 情感分析(Sentiment Analysis)
  - 文档分类(Document Classification)
  - 信息提取(Information Extraction)
  - 问答系统(Question-Answering Systems)
  - 聊天机器人(Chatbots)

SL07: NLP流程(Pipeline)



1. 数据收集(Data Collection)
2. **文本预处理(Text Preprocessing)** ← 今天学习的部分
3. 特征提取(Feature Extraction)
4. 模型训练(Model Training)
5. 评估与应用(Evaluation & Deployment)

SL08: 文本预处理的必要性



- 计算机无法直接理解原始文本(Raw Text)
- 通过去除噪声提高分析精度
- 减少数据大小以提高处理效率
- 转换为一致的格式使模式识别更容易
- 改进机器学习算法的性能

SL09:文本预处理的主要步骤



1. 文本规范化(Text Normalization)
2. 分词(Tokenization)
3. 停用词去除(Stopword Removal)
4. 词干提取/词形还原(Stemming/Lemmatization)
5. 词性标注(POS Tagging)
6. 命名实体识别(Named Entity Recognition)

SL10: 文本规范化(Text Normalization)



- 目的: 将文本转换为一致的形式
- 主要工作:
  - 转换为小写(Lowercase conversion)
  - 去除特殊字符(Special character removal)
  - 处理数字(Number handling)
  - 去除HTML标签(HTML tag removal)
  - 处理电子邮件和URL(Email, URL cleaning)

```
# 示例代码
text = text.lower()  # 转换为小写
text = re.sub(r'[^\w\s]', '', text)  # 去除特殊字符
```

SL11: 分词(Tokenization)



- 将文本分解为更小单位(标记)的过程
- 标记的类型:
  - 词(Word)分词
  - 句子(Sentence)分词
  - 字符(Character)分词
  - N元语法(N-gram)分词

```python
# 词分词示例
from nltk.tokenize import word_tokenize
tokens = word_tokenize("你好。今天天气真好！")
# 结果: ['你好', '。', '今天', '天气', '真好', '！']
```

SL12: 中文分词的特点



- 比英语更复杂的结构
- 仅靠空格无法进行足够的分词
- 需要字符级别的分析
- 主要中文分词工具:
  - jieba（结巴分词）
  - THULAC（清华大学）
  - pkuseg（北京大学）
  - SnowNLP

```python
# jieba分词示例
import jieba
tokens = jieba.cut("中文分词是处理中文文本的基础")
# 结果: ['中文', '分词', '是', '处理', '中文', '文本', '的', '基础']
```

SL13: 停用词去除(Stopword Removal)



- 停用词(Stopwords): 在分析中没有太大意义的高频词
- 示例: 'a', 'the', 'is', 'and', '的', '了', '和', '在' 等
  - 去除目的:
  - 节省存储空间
  - 提高处理速度
  - 专注于重要词语

```
# 英文停用词去除示例
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word not in stop_words]
```

SL14: 中文停用词处理



- 与英语不同，缺乏标准化的停用词列表
- 研究人员/开发者需要根据目的自定义
- 常见中文停用词:
  - 助词: '的', '地', '得', '了', '着' 等
  - 连词: '和', '与', '以及', '但是' 等
  - 代词: '我', '你', '他', '它' 等

```
# 中文停用词去除示例
chinese_stopwords = ['的', '了', '在', '是', '我', '有', '和', '就', '不', '人']
filtered_tokens = [word for word in tokens if word not in chinese_stopwords]
```

SL15: 正则表达式(Regular Expression)



- 表达字符串模式的形式语言
- 用于复杂字符串的搜索、提取和替换
- 在Python中使用re模块
- 基本模式
  - ^ : 字符串开始
  - $: 字符串结束
  - . : 任意单个字符
  - * : 重复0次或更多次
  - + : 重复1次或更多次
  - ? : 0次或1次
  - [] : 字符类（包含的字符之一）
  - | : OR运算符

SL16: 正则表达式应用示例



```python
import re

# 提取电子邮件
emails = re.findall(r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}', text)

# 提取URL
urls = re.findall(r'https?://(?:[-\w.]|(?:%[\da-fA-F]{2}))+', text)

# 仅提取中文
chinese = re.sub(r'[^\u4e00-\u9fa5\s]', '', text)

# 去除数字
no_digits = re.sub(r'\d+', '', text)

# 去除HTML标签
clean_text = re.sub(r'<.*?>', '', html_text)
```

SL17: 形态分析(Morphological Analysis)



形态分析(Morphological Analysis)

- 语素(Morpheme): 具有含义的最小语言单位
- 例: '学校里' → '学校' + '里'
- 在中文分析中尤为重要
- 类型:
  - 实质语素: 具有词汇意义（名词、动词等）
  - 形式语素: 执行语法功能（助词、词缀等）

SL18: 形态分析器(Morphological Analyzer)



- 中文的形态分析工具: jieba: 支持词性标注
  - LTP（哈工大语言技术平台）: 提供完整的中文NLP功能
  - HanLP: 支持多种语言分析功能
  - THULAC: 清华大学开发的词法分析工具

```
# jieba词性标注示例
import jieba.posseg as pseg
words = pseg.cut("我爱北京天安门")
for word, flag in words:
    print(word, flag)
# 结果: 我 r, 爱 v, 北京 ns, 天安门 ns
```

SL19: 词性标注(POS Tagging)



- 为每个词赋予词性信息的过程
- 对区分同形异义词很重要
- 语义分析和句法分析的基础
- 中文主要词性标签:
  - n: 名词
  - v: 动词
  - a: 形容词
  - d: 副词
  - p: 介词

```python
# 词性标注示例
import jieba.posseg as pseg
words = pseg.cut("今天天气真好")
for word, flag in words:
    print(word, flag)
# 结果: 今天 t, 天气 n, 真 d, 好 a
```

SL20: 词干提取(Stemming)和词形还原(Lemmatization)



词干提取(Stemming)

- 从词中删除词缀并仅提取词干(stem)
- 快速但有时会生成非实际单词
- 例: "running" → "run", "jumps" → "jump"

词形还原(Lemmatization)

- 寻找词的基本形式(lemma)的过程
- 基于形态分析和词典
- 比词干提取更准确但需要更多计算
- 例: "better" → "good", "was" → "be"

SL21: 中文的词干提取和词形还原



- 结构比英语复杂，难度更高
- 主要依靠形态分析工具`

```python
# 中文词形还原示例
import jieba
import jieba.posseg as pseg

text = "我们正在学习自然语言处理技术"
words = pseg.cut(text)

# 提取动词原形
base_verbs = []
for word, flag in words:
    if flag.startswith('v'):
        # 简化示例，实际中文词形还原更复杂
        base_verbs.append(word)

print(base_verbs)
# 结果可能包含：['学习']
```

SL22: 命名实体识别(Named Entity Recognition, NER)



- 识别文本中的人名、组织、地点等专有名词
- 文本分析的高级阶段，用于提取重要信息
- 主要命名实体类型:
  - 人名(Person): 张三, John Smith
  - 地名(Location): 北京, New York
  - 组织名(Organization): 腾讯, Google
  - 时间(Time): 2023年, 明天
  - 数量(Quantity): 300万元, 50公斤

SL23:文本预处理实用技巧



1. 选择适合目标的预处理级别
2. 考虑领域特性（医学、法律、社交媒体等）
3. 自动化预处理过程并建立流程
4. 根据数据大小考虑高效处理方法
5. 理解各语言特性（中文、英语等）
6. 通过检查预处理前后的数据样本确保质量

SL24: 理解测试



1. 以下关于停用词(Stopword)的描述哪一个是正确的?

a) 在文本中具有重要意义的核心词语

b) 频率高但对意义分析贡献不大的词语

c) 在文档中只出现一次的罕见词语

d) 两个或更多词组合的复合词

SL24: 理解测试



2. 正则表达式 ₩d+ 表示什么?
    a) 所有字符
    b) 一个或多个数字
    c) 空白字符
    d) 字母和数字

SL24: 理解测试



3. 以下哪项不是形态分析的主要目的?
    a) 确定词的词性
    b) 文本压缩
    c) 提取有意义的最小单位
    d) 理解语法结构

SL24: 理解测试



4. 关于分词(Tokenization)的描述，哪一项是正确的?

a) 将文本分解为更小的意义单位的过程

b) 从文本中删除不必要符号的过程

c) 将词转换为原形的过程

d) 文本加密的过程

SL25: 理解测试



5. 在中文处理中，与其他语言相比特别重要的步骤是?

a) 大小写转换

b) 拼写校正

c) 形态分析

d) URL删除

SL26: 理解测试答案



1. b) 频率高但对意义分析贡献不大的词语
2. b) 一个或多个数字
3. b) 文本压缩
4. a) 将文本分解为更小的意义单位的过程
5. c) 形态分析

SL27: 要点总结 (Wrap-up)



要点总结 (Wrap-up)

- 文本数据是非结构化数据，需要特殊处理
- 文本预处理是NLP流程的核心阶段
- 主要预处理步骤: 规范化、分词、停用词去除、形态分析等
- 正则表达式是基于模式的字符串处理的有用工具
- 中文作为特殊语言，形态分析尤为重要
- 预处理质量对后续分析的性能有重大影响

下次课预告

- 实践第7周: 字符串及文本处理实践
- 正则表达式应用及中文形态分析器实践

PL01: 字符串及文本处理实践 (String and Text Processing Practice)



7

课程目标:

- 掌握Pandas中的字符串处理方法
- 学习使用正则表达式处理文本
- 了解基本的文本形态分析

PL02: 今日大纲 (Outline)



1. 字符串基础 (String Basics)
2. Pandas中的.str方法 (Pandas .str Methods)
3. 正则表达式简介 (Regular Expression Introduction)
4. 文本清洗实践 (Text Cleaning Practice)
5. 简单形态分析 (Simple Morphological Analysis)
6. 案例实践 (Case Study)

•re.findall(): 패턴과 일치하는 모든 부분 찾기

•re.sub(): 패턴과 일치하는 부분을 다른 문자열로 교체

•re.split(): 패턴을 기준으로 문자열 분할

PL03: 准备工作 (Preparation)



开发环境：PyCharm IDE

1. **所需库**：

```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
import seaborn as sns
```

Regular Expression module

2. **数据集**：

◆news_headlines.csv - 新闻标题数据集

◆product_reviews.csv - 产品评论数据集

PL04: 字符串基础 (String Basics)



- Python字符串是不可变的字符序列，具有多种内置方法：
- 字符串连接 (Concatenation): +
- 字符串重复 (Repetition): *
- 索引 (Indexing): s[i]
- 切片 (Slicing): s[i:j]
- 长度 (Length): len(s)
- 查找 (Find): s.find('text')
- 替换 (Replace): s.replace('old', 'new')

PL05: 字符串方法演示 (String Methods Demo)



- 文件名: string_basics.py

PL06: Pandas中的字符串处理 (String Processing in Pandas)



- 在Pandas中，.str访问器允许在Series上应用字符串方法：
  - .str.lower() - 转换为小写
  - .str.upper() - 转换为大写
  - .str.len() - 计算字符串长度
  - .str.contains(pattern) - 检查是否包含模式
  - .str.replace(old, new) - 替换内容
  - .str.split(delimiter) - 分割字符串
  - .str.strip() - 去除前后空白字符

PL07: 加载示例数据 (Loading Sample Data)



1. 文件名: pandas_str_demo.py

PL08: 应用字符串方法 (Applying String Methods)



1. 文件名: pandas_str_demo.py

PL09: 字符串提取和替换 (String Extraction and Replacement)



1. 文件名: pandas_str_demo.py

PL10: 正则表达式简介 (Regular Expression Introduction)



1. 正则表达式(Regular Expression)是描述字符串模式的强大工具。
2. 基本语法:
   - . - 匹配任意字符（除了换行符）
   - ^ - 匹配字符串开头
   - $ - 匹配字符串结尾
   - * - 匹配前面的元素0次或多次
   - + - 匹配前面的元素1次或多次
   - ? - 匹配前面的元素0次或1次
   - \d - 匹配数字
   - \w - 匹配字母、数字或下划线
   - \s - 匹配空白字符

PL11: 正则表达式实例 (Regex Examples)



1. 文件名: regex_basics.py

PL12: 正则表达式在Pandas中的应用 (Regex in Pandas)



1. Pandas的.str访问器支持正则表达式:
   - .str.contains(pattern, regex=True) - 检查是否包含模式
   - .str.match(pattern) - 从字符串开头匹配模式
   - .str.findall(pattern) - 查找所有匹配项
   - .str.replace(pattern, repl, regex=True) - 替换匹配项
   - .str.extract(pattern) - 提取第一个匹配组

PL13: 正则表达式在Pandas中的实例 (Regex in Pandas Examples)



1. 文件名: pandas_regex.py

PL14: 应用正则表达式 (Applying Regex)



1. 文件名: pandas_regex.py

PL15: 文本清洗实践 (Text Cleaning Practice)



文本清洗是文本分析的关键步骤，通常包括：

1. 删除特殊字符 (Remove Special Characters)
2. 去除HTML标签 (Remove HTML Tags)
3. 转换为小写 (Convert to Lowercase)
4. 删除停用词 (Remove Stopwords)
5. 去除多余空格 (Remove Extra Spaces)
6. 标准化文本 (Normalize Text)

PL16: 加载新闻标题数据集 (Loading News Headlines Dataset)



1. 文件名: text_cleaning.py

PL17: 文本清洗步骤 (Text Cleaning Steps)



1. 文件名: text_cleaning.py

PL18: 查看清洗结果 (Viewing Cleaning Results)



1. 文件名: text_cleaning.py

PL19: 简单形态分析 (Simple Morphological Analysis)



形态分析(Morphological Analysis)研究词语的结构和形式。
在处理中文时，可以进行:

- 词频统计 (Word Frequency Analysis)
- 词长分析 (Word Length Analysis)
- 词性标注 (POS Tagging) - 需要jieba等库

注意: 完整的中文形态分析通常需要专业的NLP库。

PL20: 词频分析 (Word Frequency Analysis)



1. 文件名: word_analysis.py

PL21: 可视化词频 (Visualizing Word Frequency)



1. 文件名: word_analysis.py (续)

PL22:产品评论分析 (Case Study: Product Reviews Analysis)



1. 文件名: review_analysis.py
2. 数据集: product_reviews.csv

PL23: 评论文本清洗 (Review Text Cleaning)



1. 文件名: : review_analysis.py (续)

PL24: 评论长度分析 (Review Length Analysis)



1. 文件名: : review_analysis.py (续)

PL25:评论情感词分析 (Review Sentiment Words Analysis)



1. 文件名: : review_analysis.py (续)

PL26:正负面词汇与评分关系 (Relationship Between Sentiment Words and Ratings)



1. 文件名: review_analysis.py (续)

PL27:正则表达式查找特定模式 (Finding Specific Patterns with Regex)



1. 文件名: pattern_extraction.py

PL28:查看模式提取结果 (Viewing Pattern Extraction Results)



1. 文件名: pattern_extraction.py (续)

PL29: 课堂练习 (Class Exercise)



1. 打开PyCharm，创建一个新的Python文件 text_processing_exercise.py
2. 使用以下代码创建一个简单的数据框:

```
import pandas as pd

data = {
    'customer_comment': [
        '产品质量不错，价格也合理。',
        '送货太慢了，等了一周才到！',
        '客服态度很好，解决了我的问题。',
        '包装有点破损，但是商品完好。',
        '这个型号ABC-123性能很棒，推荐购买！'
    ]
}

comments_df = pd.DataFrame(data)
```

1. 任务:
   - 创建一个函数清洗评论文本
   - 检测评论是否包含"质量"、"价格"、"客服"等关键词
   - 尝试识别型号信息
   - 计算评论的字符长度

PL30: 总结 (Summary)



本次课程我们学习了:

1. Python和Pandas中的字符串处理方法
2. 正则表达式的基本语法和应用
3. 文本清洗技术
4. 基本的文本形态分析方法
5. 如何提取文本中的特定模式

下周预告: 日期型数据预处理 (Date Data Preprocessing)

作业:

- 完成文本处理练习，将清洗和分析结果提交到学习平台
- 尝试对一个真实数据集进行文本清洗和模式提取

数据预处理专业术语



