智能问答系统实践

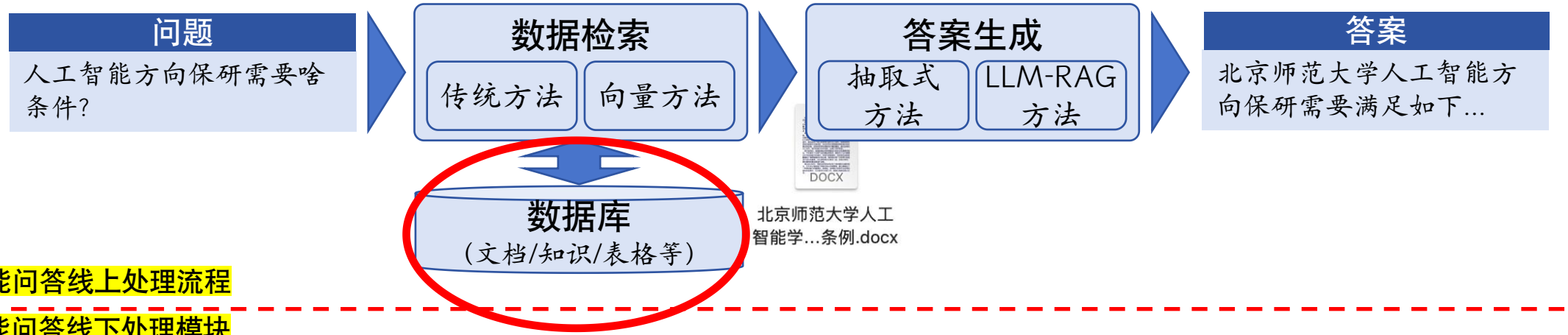
第一课：数据建库（词法分析和TF-IDF计算）

姜文斌
北京师范大学人工智能学院
2025.03.06

我的位置

智能问答系统

针对用户提出的自然语言问题，从数据库中检索相关信息，并依据相关信息作出回答



实验设置

基于传统方法的数据建库

- 中文词法分析工具使用
  - 文档的词语切分及虚词判定
- 词语-文档的TF-IDF计算
- 基于倒排索引的文档建库
  - 文档的关键词集合提取
  - 记录关键词在文档中的位置

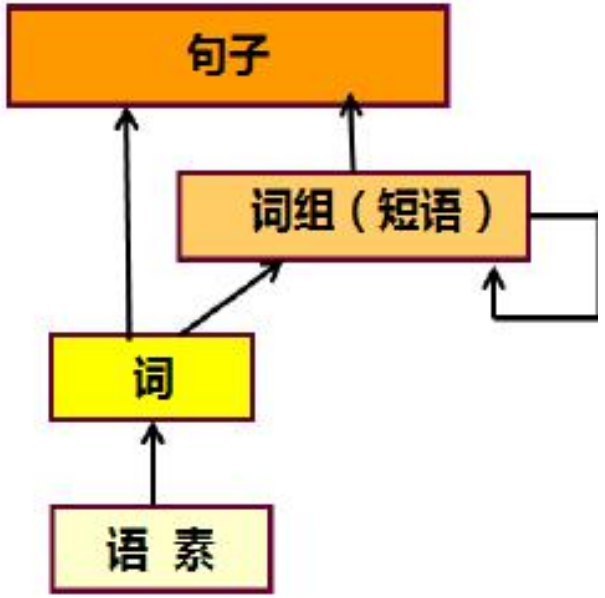
拓展：基于向量方法的数据建库

- 基于BERT的文本向量表示
  - 仅以文本模态展示向量化原理
- 基于K-Means算法的向量聚类
  - 给定聚类数目对文档向量进行聚类
- 基于HNSW向量索引的文档建库
  - 基于层次聚类建立层次化NSW图

目录

词法分析

- **词是形式和意义相结合的单位，是语言中具有独立意义的最小单位**
  - 词的处理是自然语言处理中重要的底层任务，是句法分析、文本分类、等任务的基础
  - 词通常是由语素构成，根据词在语言中用途的不同，词还可以被划分为实义词和功能词
  - 实义词主要包含名词、动词、形容词等，功能词主要包含代词、冠词、指示词等
- **词法分析是指将输入的句子字串转换成词序列并标记出各词的词性**
  - 词法根据词在句子中扮演的语法角色以及与周围词的关系对词进行分类
  - 词法分析将文本分解成更小的单元，通常是单词、短语或其他有意义的符号，然后对这些单元进行分类
  - 字串并不仅限于汉字，也可以指标点符号、外文字母、注音符号和阿拉伯数字等任何可能的文字符号
  - 词法分析的重点任务包括词语切分和词性标注



词语切分

- **词语切分是指将连续字序列转换为对应的词序列的过程**
  - 中文分词任务可以形式化表示为：输入中文句子c_1，$c_2 \ldots c_n$其中 c_i 为单个字符，输出词序列 w_1，$w_2 \ldots w_n$，其中 w_j 是中文单词
  - 例如：北京师范大学是中国人自主创办的一所高等院校
  - 分词结果：北京师范大学|是|中国人|自主|创办|的|一所|高等|院校|
- **汉语构词具有很大的灵活性，中文分词任务面临巨大挑战**
  - **分词规范**：中文因其自身语言特性的局限，字（词）的界限往往很模糊，关于字（词）的抽象定义和词边界的划定尚没有一个公认的、权威的标准
  - **切分歧义**：由于汉语构词方式的灵活性，使得同一个汉语句子很可能产生多个不同的分词结果，这些不同的分词结果也被称为切分歧义
  - **未登录词识别**：指在训练语料中没有出现或者词典当中没有，但是在测试数据中出现的词

词语切分-词典匹配

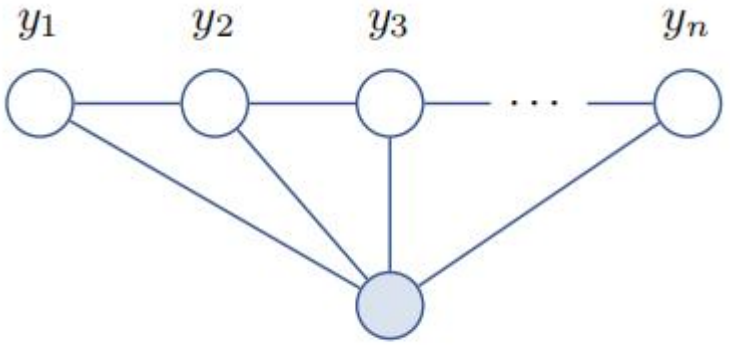
- 基于最大匹配的分词方法根据给定的词典，利用贪心搜索策略找到分词方案
  - 最大匹配分词算法主要包含前向最大匹配，后向最大匹配以及双向最大匹配
- 基本步骤
  - 从左向右扫描句子，选择当前位置与词典中最长的词进行匹配
  - 对于句子中的一个位置i,依次考虑子串$c_{[i:i+l-1]}$，$c_{[i:i+l-2]}$，．．．，$c_{[i:i]}$，其中$c_{[i:j]} \triangleq c_i c_{i+1} \ldots c_j$表示从第i个字到第j个字构成的字串,$l$表示词典中词的最大长度
  - 当某一个$c_{[i:j]}$能够对应字典中的一个词时，输出这个词并从j+1开始重复以上的过程直至整个句子被遍历完成

输入：他是研究生物化学的一位科学家

时间步	开始位置	候选匹配	输出
1	1	他是研究，他是研，他是，他	他
2	2	是研究生，是研究，是研，是	是
3	3	研究生物，研究生，研究，研	研究
4	5	生物化学，生物化，生物，生	生物化学
5	9	的一位科，的一位，的一，的	的
6	10	一位科学，一位科，一位，一	一
7	11	位科学家，位科学，位科，位	位
8	12	科学家，科学，科	科学家

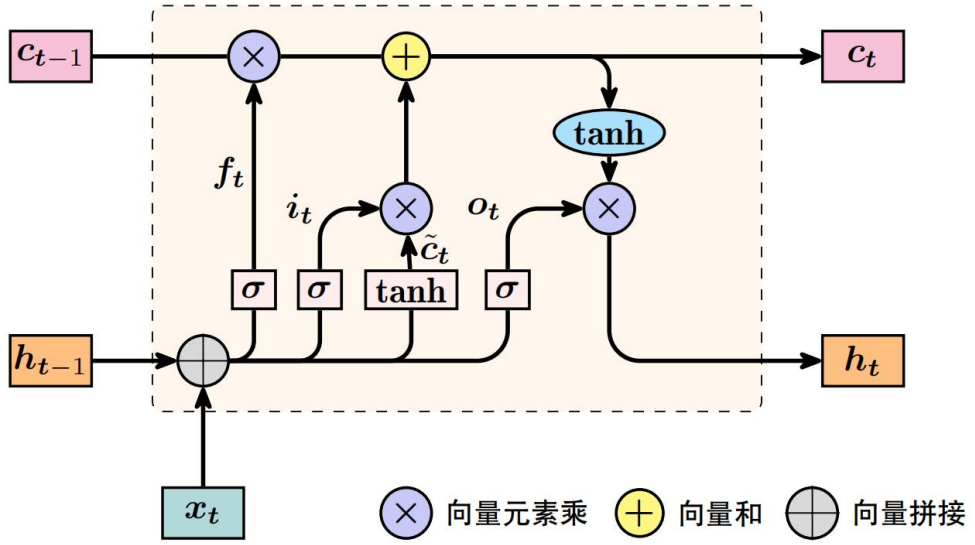
词语切分-条件随机场

- 条件随机场分词利用线性序列中相邻元素的依赖关系来对文本分词
  - 通过BIES标签可以将分词问题转换为字的分类问题
  - 对于输入句子中的每一个字c_i，根据它在分词结果中的位置赋予不同的标签
    - 输入句子：他是研究生物化学的一位科学家。
    - 分词结果：他 | 是 | 研究 | 生物化学 | 的 | 一 | 位 | 科学家 |。
    - 对应标记：他/S是/S研/B究/E生/B物/I化/I学/E的/S一/B位/E科/B学/I家/E。/S
- 条件随机场试图对多个变量在给定观测值后的条件概率进行建模
  - $x=\{x_1,x_2,\ldots,x_n\}$为观测序列，$y=\{y_1,y_2,\ldots,y_n\}$为对应的标记序列，条件随机场的目标是构建条件概率P(y|x)模型
  - x对应输入的字序列$\{c_1,c_2,\ldots,c_n\}$，标记序列为每个字对应BIES标签，条件随机场使用势函数和图结构上的团来定义条件概率P(y|x)



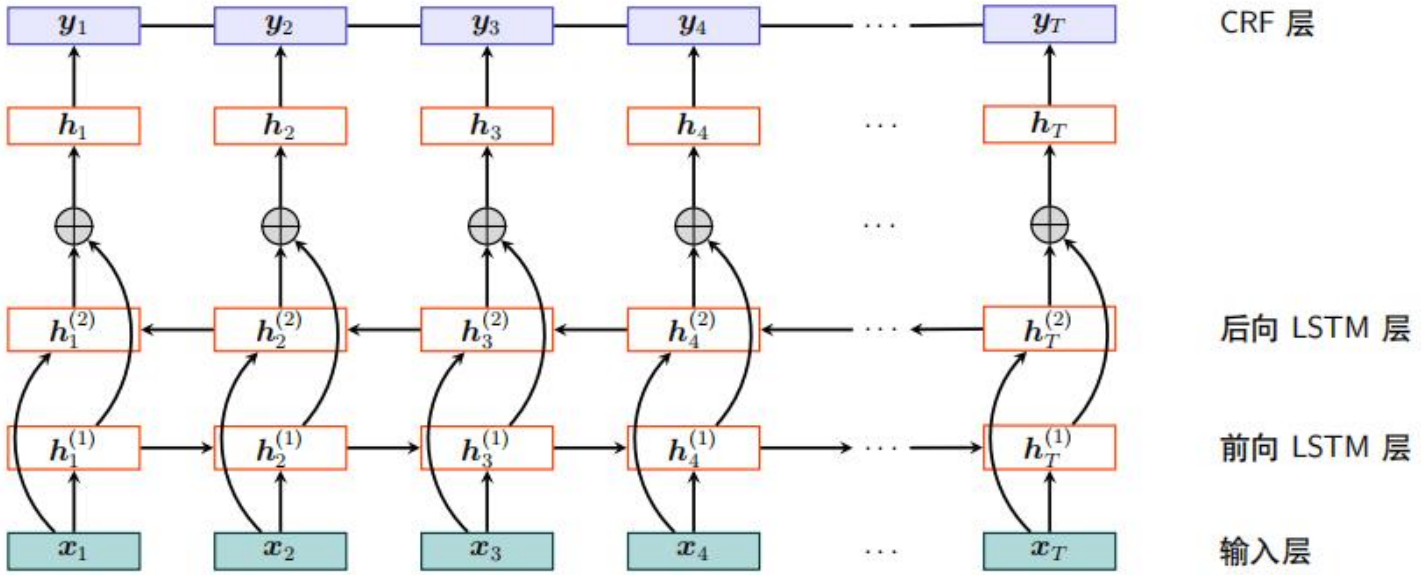
词语切分—BiLSTM

- 利用双向长短期记忆网络捕捉词组间双向依赖关系
  - 长短期记忆网络是一种具有自我门控机制的递归神经网络，能够有效解决传统RNN在处理长序列数据时的长期依赖问题
  - 长短期记忆网络在一定程度上缓解简单循环神经网络的梯度消失和梯度爆炸问题
- LSTM引入了新的内部状态$c_t \in R^D$，用来进行信息传递
  - LSTM引入了门控机制来控制信息传递路径
  - 遗忘门f_t控制上一个时刻的内部状态c_{t-1}需要遗忘多少信息
  - 输入门i_t用来控制当前时刻的候选状态c_t有多少信息需要保存
  - 输出门O_t控制当前时刻内部状态c_t有多少信息需要输出给外部状态h_t



词语切分-BiLSTM

- BiLSTM能够同时处理序列中的过去和未来信息，以捕捉双向依赖关系
  - BiLSTM是由两层长短期记忆网络组成，它们结构相同但是信息传递的方向不同
  - BiLSTM能够同时考虑序列中每个点的过去和未来信息，增强了模型对上下文的理解能力
- BiLSTM还可以结合条件随机场，有效利用结构化学习和神经网络的能力
  - **输入层**:将每个字转换为低维稠密的字向量
  - **BiLSTM层**:采用双向LSTM，其主要作用是提取句子特征提取句子特征
  - **CRF层**:学习标签之间的转移概率，优化序列标注任务中标签序列的预测



词性标注

- **词性标注是指在给定的语境中确定句子中各词的词性，是句法分析的基础**
  - 词性是词语的基本属性，根据其在句子中所扮演的语法角色以及与周围词的关系进行分类
  - 兼类词多为常用词，而且越是常用词，其用法就越多，例如，英语“like”具有动词、名词、介词等多种词性
  - 词性标注的主要难点在于歧义性，即一个词可能在不同的上下文中具有不同的词性，因此需要结合上下文来确定词在句子中所对应的词性
    - “book”可以表示名词“书”，也可以表示动词“预定”
    - “good”可以表示形容词“好”，也可以表示名词“货物”
- **中文尚无统一标注标准，常用的有北大词性标注集和宾州树库词性标注集等**
  - 由于词性表以及词性定义有许多不同的变种，词性标注的结果与这些标注密切相关
  - 在不同的语料集中所采用的划分粒度和标记符号也都不尽相同

词性标注-规则方法

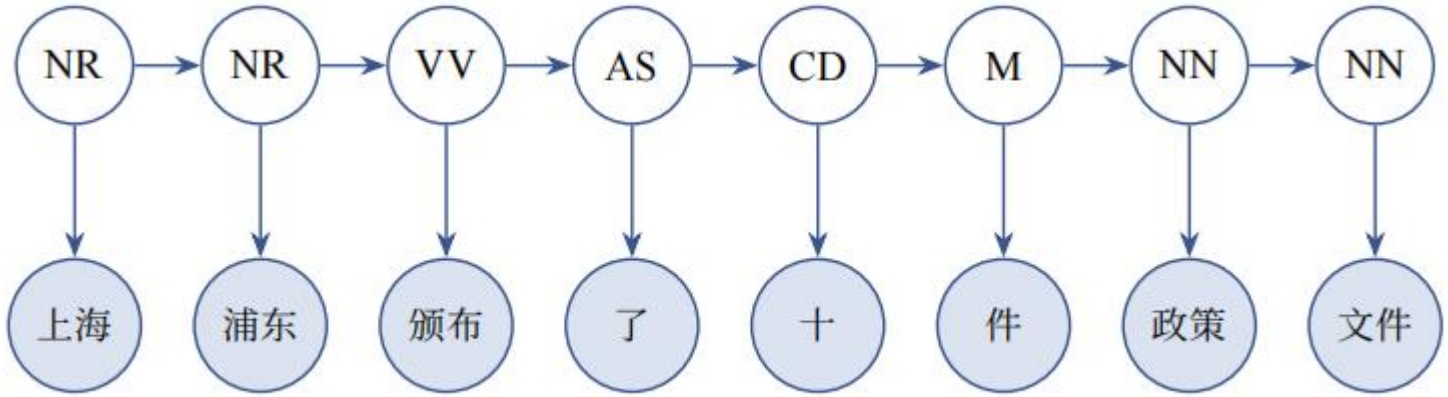
- 基于规则的词性标注算法利用词典和搭配规则针对词语和上下文进行分析
- Brill Tagger利用错误驱动方法，学习用于词性标注的转换规则
  - **初始化**
    - 对于词典中包含的词语，根据最常使用的词性设置初始值
    - 对于词典中没有的单词根据某种词性分析规则设置初始值
  - **转换规则**：根据转换规则对初始标注进行转换，对于某单词
    - 如果词性为a，并且其所在上下文为C，那么将其词性转换为b
    - 如果词性为a，并且其具有词汇属性P，那么将其词性转换为 b
    - 如果词性为a，并且其周边范围R内有一个词具有属性P，那么将其词性转换为 b
  - **规则学习**：根据错误驱动的策略学习转换规则
    - 首先根据现有的初始词典和转换模板对训练语料进行分析
    - 然后利用上述模板得到的转换规则进行校正，计算该规则可以修复的错误数
    - 修复错误数更高的转换规则，将获得更好的优先级

词性标注-隐马模型

- 隐马尔可夫模型学习状态转移概率和发射概率，预测最可能的词性标注序列
  - 隐马尔可夫模型是用来描述一个含有隐含未知参数的马尔可夫过程
  - 一个隐马尔可夫模型可用如下 5 个参数定义：N 状态数、M 观察值数、π 初始状态概率、A 状态转移概率矩阵、B 观测概率矩阵
- 隐马尔可夫模型的三个主要问题
  - **观测概率计算**：在给定模型的情况下，如何根据观测序列计算观测概率 P，即在给定模型情况下，如何计算观测序列的概率
  - **状态序列预测**：在给定模型和观测序列的情况下，如何得到与该观测序列最匹配的状态序列，即如何根据观测序列推断出隐藏的状态序列
  - **模型参数学习**：在给定观测序列情况下，如何调整模型参数使得该序列的P最大，即如何训练模型使其能最好地建模观测序列

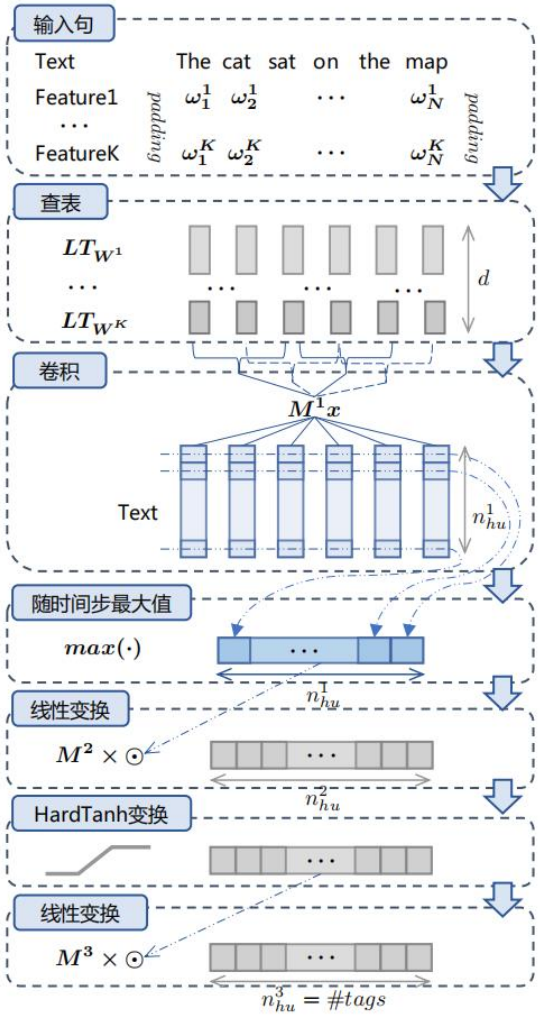
词性标注-隐马模型

- 基于隐马尔可夫模型，可以按照如下方式构建和学习模型
  - **输入**：给定句子W = $w_1,w_2,...,w_T$，即观测序列O ={$o_1,o_2,...,o_T$}，o_i为句子中第i个单词w_i
  - **输出**：状态序列Q ={$q_1,q_2,...,q_T$}则表示输入句子中单词对应的词性
  - **学习**：使用最大似然估计方法可以得到模型参数即状态转移概率和发射概率
  - **解码**：在此基础上，针对输入句子可以利用维特比算法应用动态规划求解状态路径



词性标注-卷积网络

- 利用CNN捕捉局部上下文信息，以提取词语特征表示
  - **模型**
    - 首先通过查找表（Lookup Table）LT_W将单词转换为词向量，d_{wrd}表示词向量的维度
    - 卷积层根据所设置的窗口大小d_{win}，将每个单词周边的单词拼接起来构成具有$d_{wrd}d_{win}$维度的向量
    - 针对通过池化层计算得到的向量，利用公式进行非线性变换，再叠加新的线性层后完成特征提取工作
  - **训练**：基于最大化对数似然目标，可以根据标注语料训练得到模型参数θ，根据模型参数
  - **解码**：使用维特比算法可以获得任意句子中每个词的词性



目录

jieba下载与安装

全自动下载

- 终端输入“pip install jieba”/“pip3 install jieba”（支持python2，3）

```
(.venv) PS D:\pythonProject6> pip install jieba
```

```
(.venv) PS D:\pythonProject6> pip3 install jieba
```

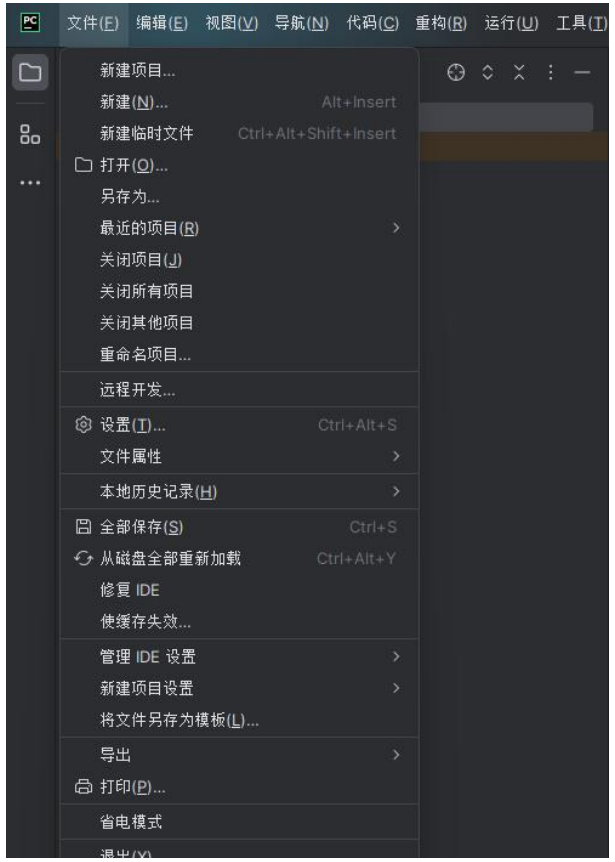
- 如果下载很慢，切换下载源，最好使用清华源（其命令行为“pip install jieba -i https://pypi.tuna.tsinghua.edu.cn/simple”）

```
(.venv) PS D:\pythonProject6> pip install jieba -i https://pypi.tuna.tsinghua.edu.cn/simple
```

jieba下载与安装

手动下载

- 打开pycharm中“文件”

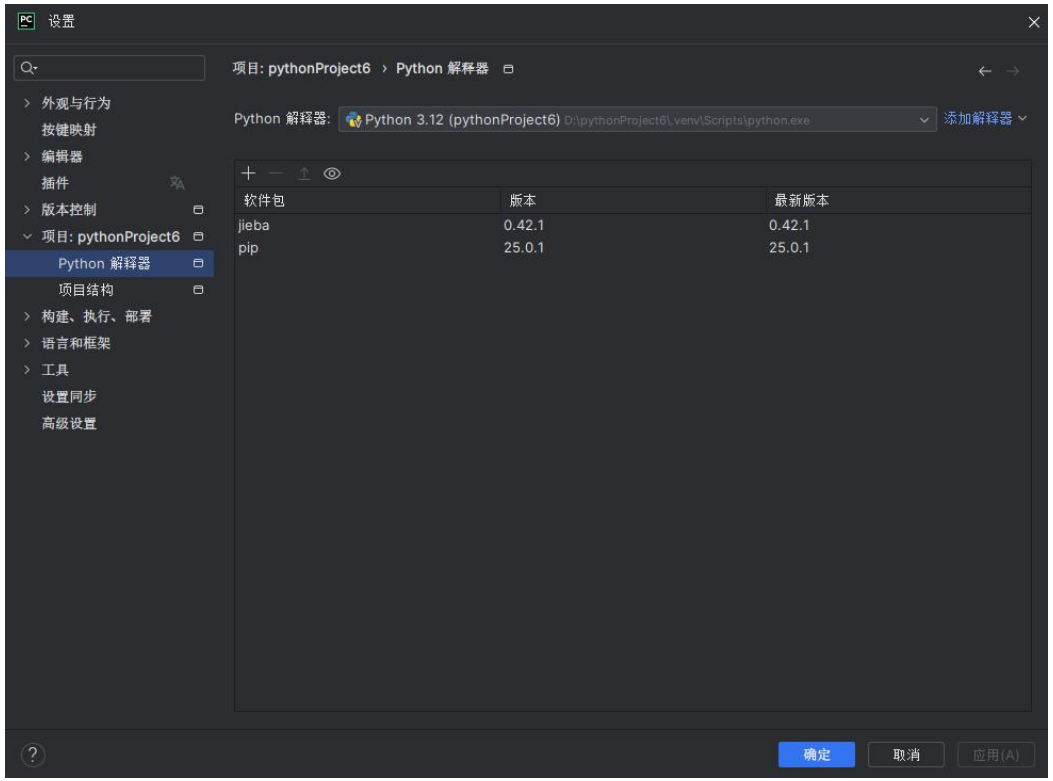


- 打开设置，点击项目

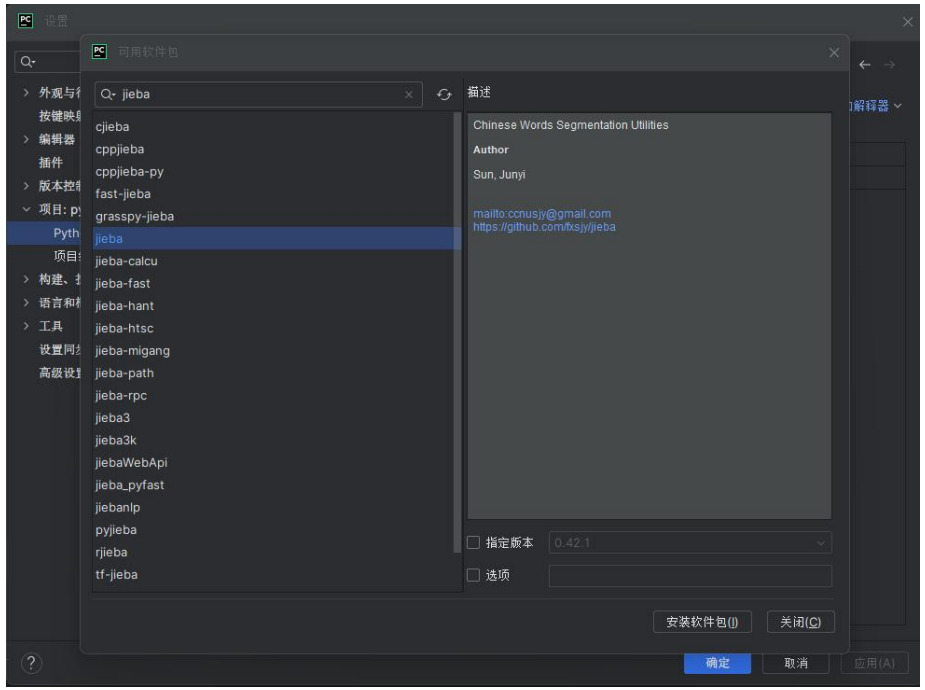
jieba下载与安装

手动下载

- 点击python编译器



- 点击文字“软件包”上方的“+”输入“jieba”再点击“安装软件包”



jieba常用函数

常见函数

函数	描述
jieba.cut(s)	精确模式，返回一个可迭代的数据类型(默认cut_all=False)
jieba.cut(s, cut_all=True)	全模式，输出文本s中所有可能单词
jieba.cut_for_search(s)	搜索引擎模式，适合搜索引擎建立索引的分词结果
jieba.lcut(s)	精确模式，返回一个列表类型，建议使用
jieba.lcut(s, cut_all=True)	全模式，返回一个列表类型，建议使用
jieba.lcut_for_search(s)	搜索引擎模式，返回一个列表类型，建议使用
jieba.add_word(w)	向分词词典中增加新词W，建议使用
jieba.del_word(x)	向分词词典中删除词语x，建议使用

- s：待分词的中文文本
- cut_all：是否采用全模式进行分词，默认为False

jieba三种使用模式

精确模式

- 最基本的分词模式，将句子切分成最小的词语单元，不存在冗余词语

```python
import jieba
sentence = "我爱北京师范大学"
words = jieba.cut(sentence,cut_all=False)
print("/".join(words))
```

```
运行  main
D:\pythonProject6\.venv\Scripts\python.exe D:\pythonProject6\main.py
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\dzq\AppData\Local\Temp\jieba.cache
Loading model cost 0.337 seconds.
Prefix dict has been built successfully.
我/爱/北京师范大学

进程已结束，退出代码为 0
```

jieba三种使用模式

全模式（全文扫描切分）

- 全模式将句子中所有可能的词语都扫描出来，可能包含一些无意义或重复的词语，有冗余，即在文本中从不同角度分词

main.py

```
1  import jieba
2  sentence = "我爱北京师范大学"
3  words = jieba.cut(sentence,cut_all=True)
4  print("/".join(words))
```

运行 main

```
D:\pythonProject6\.venv\Scripts\python.exe D:\pythonProject6\main.py
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\dzq\AppData\Local\Temp\jieba.cache
Loading model cost 0.311 seconds.
Prefix dict has been built successfully.
我/爱/北京/北京师范大学/京师/师范/师范大学/大学

进程已结束，退出代码为 0
```

jieba三种使用模式

搜索引擎模式

- 在精确模式的基础上进行了进一步的切分，对一些长词接下来再次切分，得到更细粒度的词语

main.py

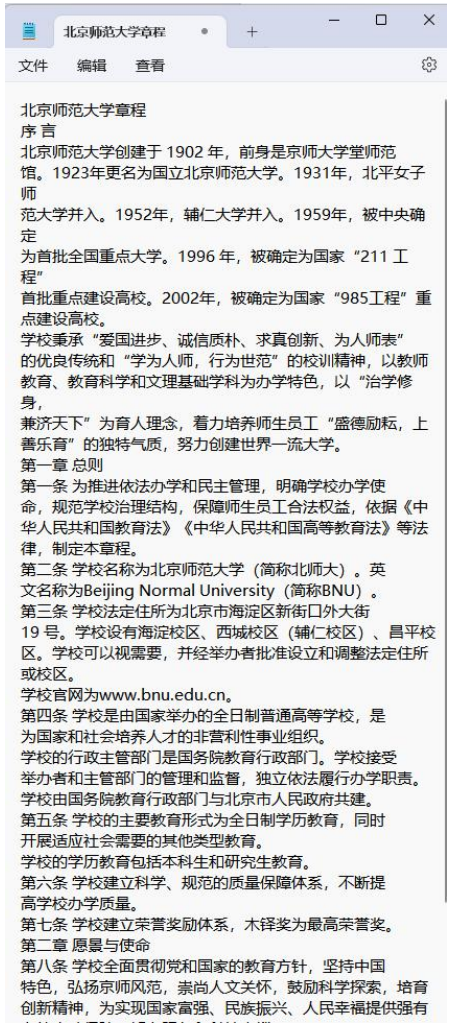
```
1  import jieba
2  sentence = "我爱北京师范大学"
3  words = jieba.lcut_for_search(sentence)
4  print("/".join(words))
```

运行 main

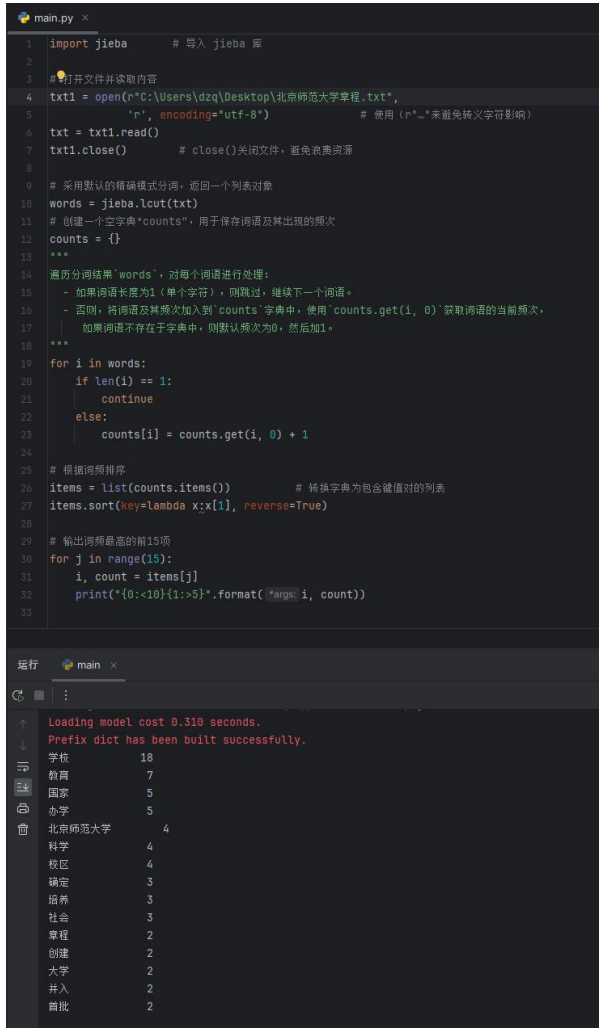
```
D:\pythonProject6\.venv\Scripts\python.exe D:\pythonProject6\main.py
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\dzq\AppData\Local\Temp\jieba.cache
Loading model cost 0.339 seconds.
Prefix dict has been built successfully.
我/爱/北京/京师/师范/大学/北京师范大学

进程已结束，退出代码为 0
```

jieba中文词频统计

- 创建一个中文内容的文本文件



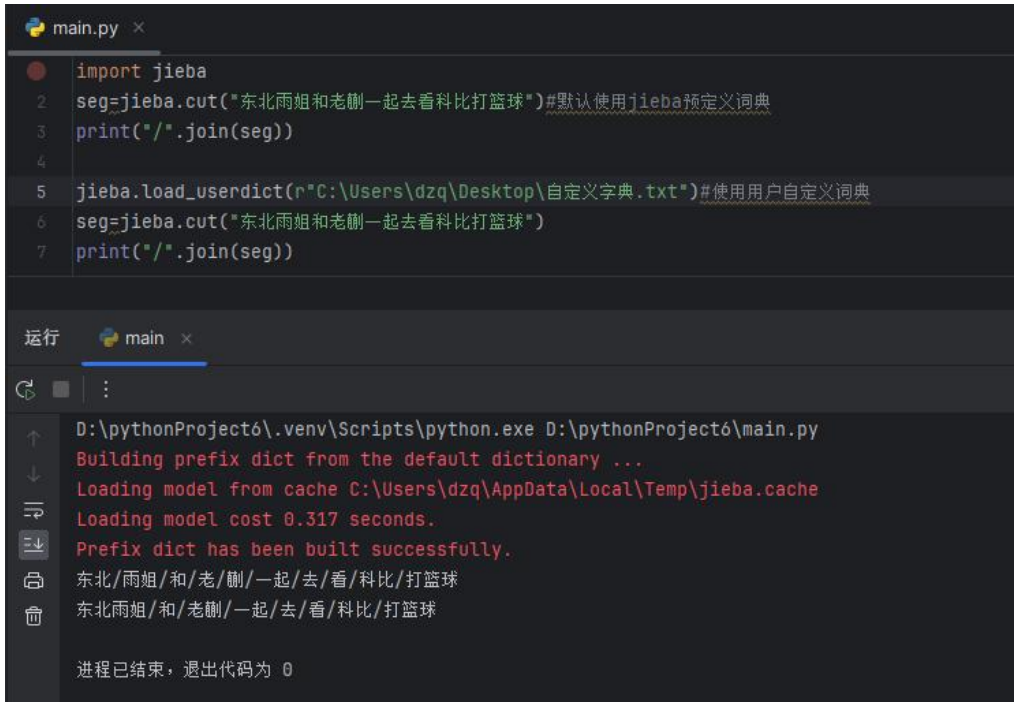
- 代码测试



jieba自定义词典

- 上诉流程中代码使用的是jieba中自定义的词典，我们也可以使用自己定义的词典
  - 使用jieba.load_userdict（“filepath”）方法获取自定义词典（数字为词频（可忽略），n为词性（可忽略，文件可为txt，或者csv格式）

  - 编译结果



目录

TD-IDF

- **概念**
  - TF-IDF（Term Frequency-Inverse Document Frequency） 是一种用于信息检索的统计方法，用于评估一个词语在文档中的重要程度
- **核心思想**
  - 一个词在某个文档中出现的频率越高（TF 高），且在语料库中包含该词的文档越少（IDF 高），则该词对当前文档的重要性越高
  - 综合局部重要性和全局稀缺性
- **计算公式**
  - TF-IDF = TF（文档词频） × IDF（逆文档频率）

TD-IDF

词语频率（Term Frequency, TF）

- 表示词语在文档中的出现频率
- 计算公式：

$$\mathrm{TF}(t,d) = \frac{\text{词 } t \text{ 在文档 } d \text{ 中的出现次数}}{\text{文档 } d \text{ 的总词数}}$$

逆文档频率（Inverse Document Frequency, IDF）

- 表示词在整个语料库中的稀缺性
- 计算公式：

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{语料库中文档总数 } N}{\text{包含词 } t \text{ 的文档数 } + 1}\right)$$

- IDF 分母中的 +1 是为了避免除零错误（例如，未出现的词）
- 作用：惩罚高频词（如“的”、“是”），提升稀缺词的权重

TD-IDF

代码

- 简化版中文停用词表，预处理防止干扰结果

```
# 中文停用词列表（过滤无意义词）
STOP_WORDS = {"的", "和", "是", "在", "了", "都", "有"}
```

- 计算TF

```
def calculate_tf(text):
    """计算词频（TF）"""
    words = preprocess(text)
    word_count = Counter(words)
    total_words = len(words)
    return {word: count / total_words for word, count in word_count.items()}
```

- 计算IDF

```
def calculate_idf(documents):
    """计算逆文档频率（IDF）"""
    N = len(documents)
    word_doc_count = {}
    for doc in documents:
        words = set(preprocess(doc))
        for word in words:
            word_doc_count[word] = word_doc_count.get(word, 0) + 1
    return {word: math.log(N / (count + 1)) for word, count in word_doc_count.items()}
```

TD-IDF

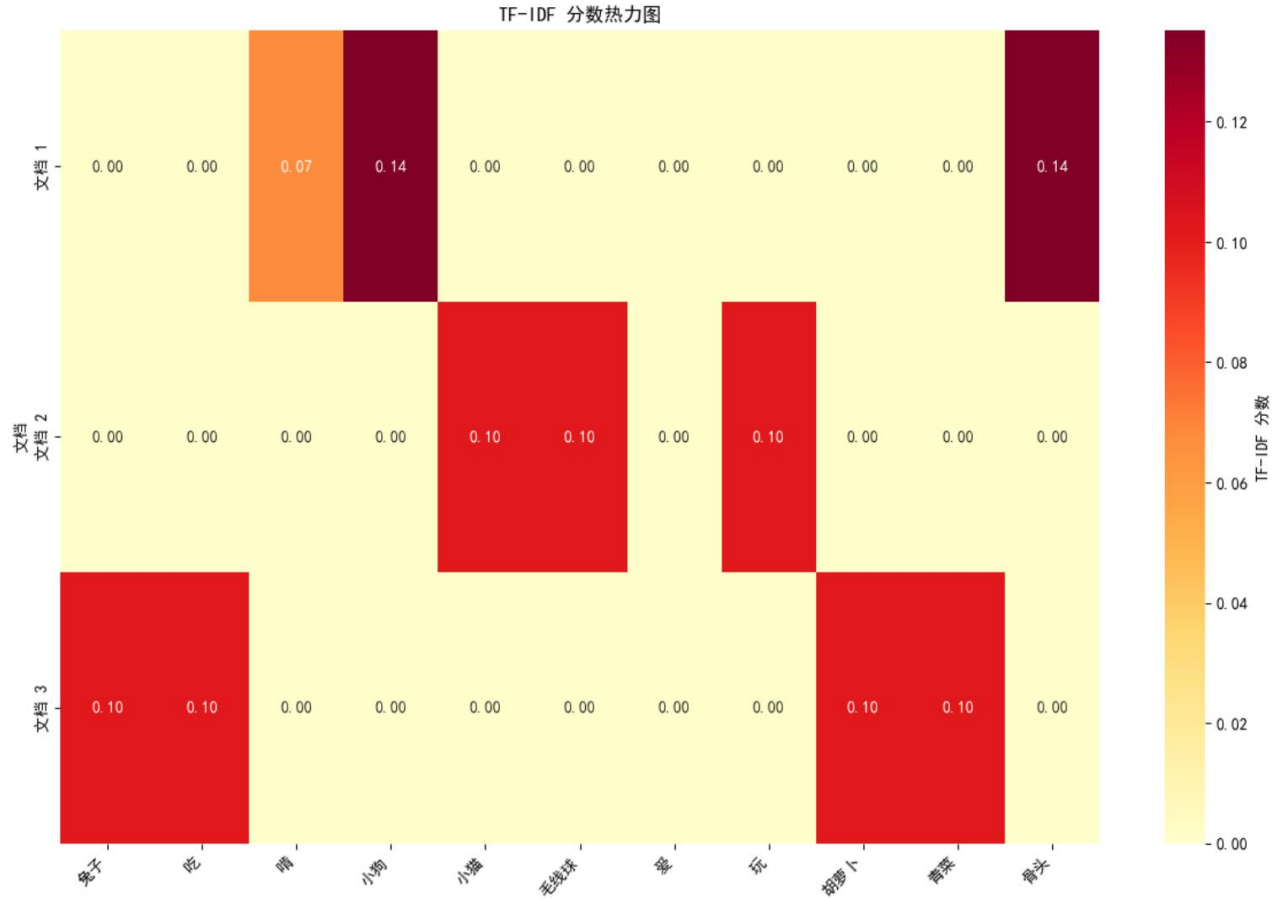
结果分析

- ‘爱’在两个文档中重复出现

$$\mathrm{IDF}(爱) = \log\left(\frac{3}{2+1}\right) = \log(1) = 0$$

- 文档1小狗和骨头重复出现
- 小猫和毛线球只在文档2出现
- 和属于无意义词（预处理）

```
documents = [
    "小狗 小狗 爱 啃 骨头 骨头",   # 文档1
    "小猫 爱 玩 毛线球",           # 文档2
    "兔子 吃 胡萝卜 和 青菜"       # 文档3
]
```



附：数据集介绍

- TXT 格式说明
  - [TEXT]: 阅读理解题目的段落内容
  - [QUESTION]: 该段落的问题
  - [ANSWER]: 该问题的答案，多个问题及其对应答案配对排列
- 示例

```
[TEXT] 范廷颂枢机是越南罗马天主教枢机。1963年被任为主教；1990年被擢升为天主教河内总教区宗座署理。
[QUESTION] 范廷颂是什么时候被任为主教的？
[ANSWER] 1963年
[QUESTION] 1990年，范廷颂担任什么职务？
[ANSWER] 天主教河内总教区宗座署理

[TEXT] 安雅·罗素法，来自俄罗斯圣彼得堡的模特儿。她是《全美超级模特儿新秀大赛》第十季的亚军。
[QUESTION] 安雅·罗素法参加了什么比赛获得了亚军？
[ANSWER] 《全美超级模特儿新秀大赛》第十季
[QUESTION] Russell Tanoue 对安雅·罗素法的评价是什么？
[ANSWER] 有前途的新面孔
```

谢谢大家！