# 司法大数据

# 组队情况

本小组由三名成员组成。

201180024 陈灵灵 2833328717@qq.com 主要负责网页前端制作和前后端交互。

201250034 韩佳希 350039541@qq.com 主要负责文本分析、ppt制作和论文撰写。

201250056 胡雨翎 1286643955@qq.com 主要负责爬虫、数据库和论文撰写。

# 研究问题

## 研究背景

### 筛选有效信息

大数据技术的意义不在于掌握庞大的数据信息，而在于对这些含有意义的数据进行专业化处理，从海量数据中找到不容易昭示的规律和预测，从而使人受益。对于司法数据分析亦是如此。特别是近年来国家提出了建设“智慧法院”的意见，更加说明了大数据技术对服务人民群中、服务审判执行、服务司法管理的重要保障作用。虽然裁判文书网为我们提供了海量的文书信息，但缺少对关键信息标注、相似案例分析等功能，使得筛选有效信息有一定困难。

### 普及法律知识

法制宣传教育是社会主义法制建设的一项基础性工作，深入开展法制宣传教育，扎实推进依法治理，是实现科学发展的必然要求，也是全面贯彻落实科学发展观的重要举措。我们的项目成果能让普通大众方便地获取案件内容和案件结果，有利于普法活动的展开，帮助民众了解法律法规，做一个知法、懂法、守法的好公民。

## 研究目标

### 爬取目标文书

用户在页面中输入搜索关键字、开始日期、结束日期以及爬取数量后，自动下载目标文书到本地，并存入数据库。

### 显示分词结果，标注关键信息

通过手动复制或选择爬取下的文件上传至前端页面后，根据词性进行自动化分词，用户以勾选框的方式勾选案件当事人的姓名、性别、民族、籍贯、罪名、相关法院。且分词会优先显示正确率较高的词语，用户可以通过滑动栏、展开栏查看更多可能的分词结果。最后点击保存标注，将标注结果以json文件形式保存至本地。

### 特征要素提取

通过训练特征要素模型对案件进行解析，提取出案件特征要素，包括事故类型、罪名、主次责任、减刑因素、加刑因素和判决结果，显示在前端，作为对分词标注的补充。若提取出的特征要素不够准确，用户可以进行手动修改。

### 通过分析大数据寻找相似案例

用户上传目标案例后，点击寻找相似案例，通过存放在数据库中的大量案例及其特征的比较，可以找到最相似案例推荐给用户参考。

# 代码文件介绍



图表 1 源码文件目录

首先在大文件夹djangoProject1下，放入了数据库database.db和CRF模型测试的相关文件。数据库与网页的连接通过database\_process.py进行操作。获得特征要素的总方法是get\_data.py。模型测试过程中，template01是模型训练时用到的模版文件，ltp\_true.py是文本预处理的分词文件，并将测试文件写入output1文件中，crf\_test.exe用于运行模型测试,model是之前已经被训练好的模型，最后将结果写入result文件中。特征要素提取则需要feature.py文件来提取。

接下来是分块的前端页面功能。APP1，APP2，APP3分别对应着每部分功能的前端页面。APP1连接爬虫，APP2连接分词和特征要素，APP3连接本地保存按钮，APP3连接相似案例提取。

最后还有后端需要的工具包，包括用于处理相似案例的word2evc包，用于分词标注的分词包和特征提取的包。与此同时，分词和特征提取由于命令行的同级问题已经嵌入在上面前端页面的连接和响应代码中，并无调用的作用。

具体代码介绍见说明文档。

# 实现方法

## 过程综述



图表 2 总流程图

## 基础板块

### 前端及前后端交互

#### HTML，CSS，JavaScript编写前端页面

通过html基本的网页元素，CSS样式布局，以及JavaScript动态效果设计，在本地实现了一个可供用户传递数据到后端进行处理的前端网页。用户可以将时间，关键词和数量提交到后端爬虫进行爬取，选择本地文件上传进行分词，在前端由切换栏显示分词结果，选择关键信息并可以进行一定修改后传到后端保存到本地。

前端布局以及切换栏使用到较多CSS，分词完成后显示切换栏，切换栏里复选框的显示等事件使用到了精巧的JavaScript设计。

#### Django作为Web框架

在实现过程中，如何实现前后端数据传递是第一个难点，经过研讨后决定采用Django作为Web框架，前端设定一个url路径，即localhost:8000/appHTML/，发送数据到后端程序相应url路径。这里以分词为例，分词相应程序对应的路径为localhost:8000/APP2/，则在“上传进行分词”按钮中绑定点击事件，发送文件内容至后端，后端产生响应，在views.py中进行分词处理，处理后形成一个字典，通过return JsonResponse()以json的形式返回给前端，前端通过JavaScript将其展示在页面上。其他板块与此类似。

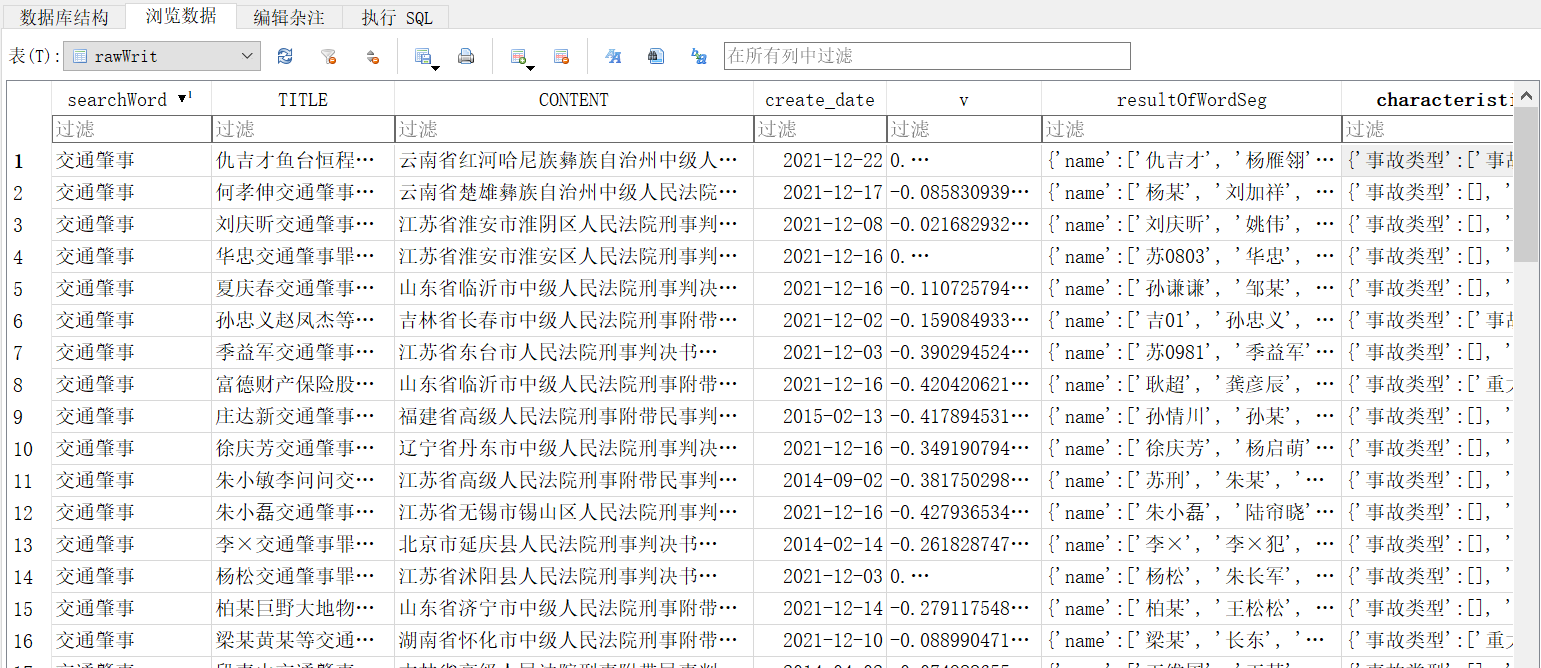
#### Ajax异步刷新

在进行前后端交互的过程中，考虑到需要进行多次数据传递，每次传递都需要刷新页面，造成使用上的不便，故提交时采用Ajax，以达到异步刷新的效果，这样每次提交完就不需要刷新页面了，使得整体前后连贯，流畅自然。在编写时引入了Jquery，使代码更清晰，更简洁。

### 数据库

#### 表头元素

每次下载新的文书之后，自动提取出**搜索关键字、文书的标题及内容**；利用正则表达式从文书内容中匹配出**文书日期**；对文书内容调用分词中conclude.py中的cut方法，并将分词结果从字典转化为字符串，获得**分词标注结果**；调用特征提取中get\_data.py中的word\_process\_text方法计算出文书的**特征要素**，同样将结果从字典转化为字符串形式；最后调用word2evc文件中word2evc\_method.py的方法计算出每一份文书的**词向量值**。以上结果全部存入数据库中。



图表 3 数据库展示

#### 主要功能

一是提高**搜索效率**。用户在搜索文书时，会根据搜索关键字和日期范围优先去数据库中搜索符合条件的文书。若符合条件的文书数量不足，则再去裁判文书网下载。从数据库中取数据的速度非常快，因此这样可以大大提高用户搜索文书时的效率。



图表 4 数据库结合爬虫

二是寻找**相似案例**。用户上传文书内容寻找与之相似的案例时，会先计算出此份文书的向量v1,然后遍历数据库中所有文书的向量，并且每次都会调用word2evc\_method.py中的vector\_similarity方法计算相似度，从而得到相似度最大的一份文书并反馈给用户。

三是提高**分词速度**。每次将文书存入数据库时，都会计算其分词结果和特征要素并一起存入。因此用户需要分词时可以直接从数据库中取出结果并进行修改，省去了等待后端计算的时间，提高了分词的速度。

## 爬虫

#### 初阶段方法

本来采用的是**selenium模拟用户登录+request爬取案例文书**的方式下载txt形式的文书到本地。但是在去年12月裁判文书网更新反爬策略之后，selenium失效。我们用request方法也可以登陆成功，但无法保存登陆后的cookie，经过多次尝试之后决定放弃此方法。

#### 修改方法

此后我们转换思路：selenium的基本思路是模拟用户在浏览器中进行操作，那我们也可以用python自带的**pyautogui**和**pyperclip**库**模拟鼠键操作**，结合文书网的高级检索功能直接下载文书，并且通过解压缩文件、将word文件转化为txt文件两步操作得到与之前相同的结果。

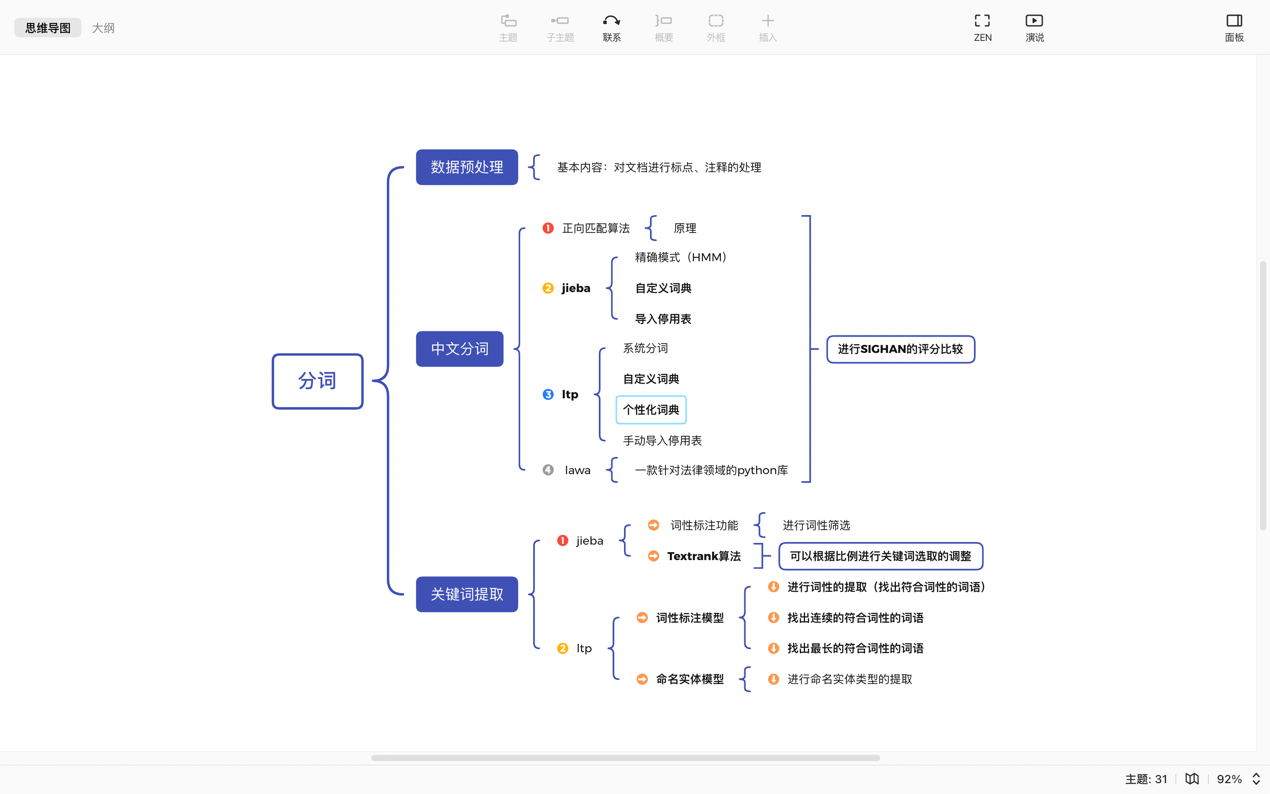
#### 效果比较

#### 经过比较之后，我们认为此种方法甚至比selenium+request更加稳定、更加快速。

## 文本分析

### 中文分词

#### 分词思路



图表 5 分词思路

#### 数据预处理

首先是对**括号与书名**号内的注释进行删除，过滤掉无用信息，其次是对**标点符号**进行删除，避免分词时标点单独成词，降低分词利用率。

此后发现，由于中文使用标点符号进行断句和分割，删除标点符号影响分词方法的准确率，因此对于中文分词，暂时不去掉标点符号。

#### 方法比较

方法1是**正向匹配算法**。正向最大匹配法，对于输入的一段文本从左至右、以贪心的方式切出当前位置上长度最大的词。根据参考文献[1][2],正向最大匹配法是基于词典的分词方法，其分词原理是:单词的颗粒度越大，所能表示的含义越确切。

方法2是调用python库里的**jieba**包。jieba分词的原理是利用一个中文词库，确定汉字之间的关联概率，汉字间概率大的组成词组，形成分词结果。首先jieba包自带**精确模式**，把一段文本精确地切分成若干个中文单词，若干个中文单词之间经过组合，就精确地还原为之前的文本且其中不存在冗余单词。此后为了提高分词准确率，我们在其中加入了**法律相关的自定义词典**，并且加入了**停用词表**，停用词表中包括了一些日常会使用到的转折词等等，精简了分词所得的结果。

方法3是调用python库里的**pyltp**包。pyltp 是哈工大自然语言工作组推出的一款基于Python 封装的自然语言处理工具，提供了分词，词性标注，命名实体识别，依存句法分析，语义角色标注的功能。为了提高准确率，我们也在其中导入了**法律相关的词典**和**停用词表**。

方法4是调用python库里的**lawa**包。Lawa是一个基于法律语料的中文分词工具，因此对于法律元素的提取会更加敏感。

经过SIGHAN评分后，结果显示ltp效果更好，如图所示。

图表 6 分词精确度

#### 信息提取

根据要求，我们需要提取出当事人、性别、民族、出生地、罪名、相关法院等信息。

本小组采用多层分析的形式，首先是对每一类的信息进行文本分析后的可能性最大的信息提取。其次是通过词性筛选的形式，展示特定词性的名词供勾选。最后还保留网页修改的功能，便于在结果还不满意的时候，进行用户的直接修改。

**一是文本分析后的信息提取。**由于当事人和出生地，都处于**命名实体**识别的范围，导入pyltp的ner.model更加精准的获得信息。而罪名和相关法院等信息更偏向于法律相关，因此在jieba中导入**相关词典**获取对应词性的方法更为便捷。至于性别和民族，范围小，种类少，便于提取。

**二是词性分析后的信息提取。**初步提取未必能得到最精确的答案，由于所需的信息都是名词类别，我们将名词信息传入前端，供用户勾选。初步分词也很有可能将原本的完整词块切割成多个连续的名词，因此我们在后端进行了包括**单个名词，连续多个名词，去重的多个名词在内的三种名词类别**，在用户不满意初始分析时进行**勾选框**的选择。

三是手动修改。若是结果仍不满意，我们将采取手动修改的模式进行信息的保存。

### 特征要素

#### 前期探索

首先想到的便是jieba包自带的**TextRank算法**和**TF-IDF算法**结合获得关键词的方法。

TextRank算法从一段给定的文本中自动抽取出若干有意义的词语或词组，利用局部词汇之间关系对后续关键词进行排序，直接从文本本身抽取。

TF-IDF算法是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性与出现的次数成正比，但同时会随着它在语料库中出现的频率成反比下降。

但是根据后期文本测试发现，通过将两类算法调整概率比例而获得的关键词，往往会因为词频达不到要求而出现数量少、提取无效的情况，故转而采用**条件随机场CRF预测算法**。

#### CRF原理

根据参考文献[3][4][5][6],首先假设文本案例符合马尔科夫随机场，即状态xi的状态满足马尔可夫过程，i时刻的状态xi的条件分布，仅与其前一个状态xi-1相关，即假设观测序列中各个状态仅取决于它所对应的隐状态。

以线性链条件随机场为例，马尔可夫随机场的参数化形式如下。



图表 7 CRF欲实现功能

对于序列标注，我们需定义特征函数集合，用这个特征函数集合来为标注序列打分，并据此选出最合适的序列。而对于CRF条件随机场，条件指的是，对于输入序列X，根据条件概率来计算输出序列Y。随机场指的是，给若干位置按照某种分布赋予一个值。故我们借助CRF随机场，对于文段中的特征要素，先训练出有效的特征函数和条件概率分布，以此预测其对应的标注序列，最后按照得到的标注序列进行要素提取，也就是模型学习问题和模型预测问题。

在模型学习问题中，我们需要将已知的观测序列Y,求解使得该观测序列概率最大的模型参数，包括隐状态序列、隐状态间的转移概率分布和从隐状态到观测状态的概率分布。

而在预测问题中，我们将学习后获得的模型所有参数和观测序列Y，计算最可能的隐状态序列X。其中常用维特比算法，通过动态规划寻找最优的观测路径。

#### 数据预处理

同相似案件的数据预处理方式，通过截取判决书和分词的方法，关注事件的判决过程。

#### 语料训练

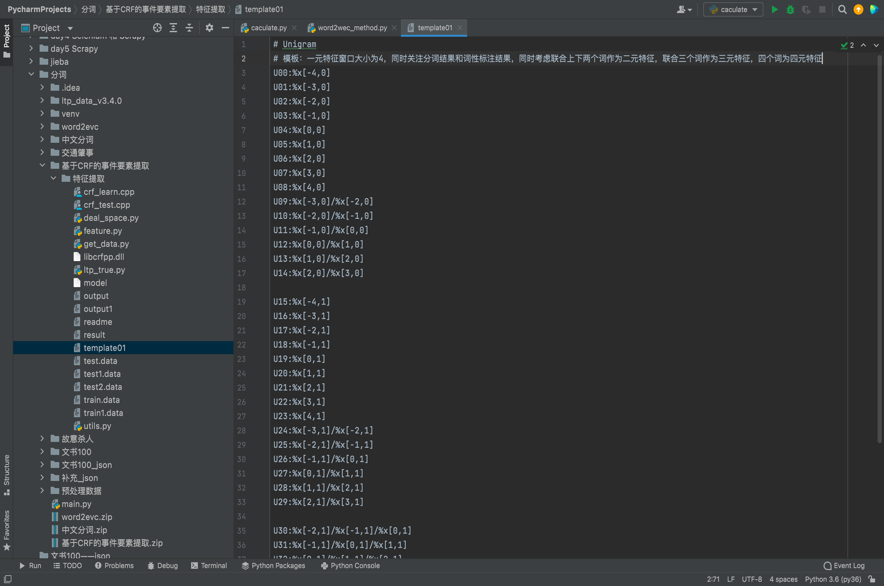
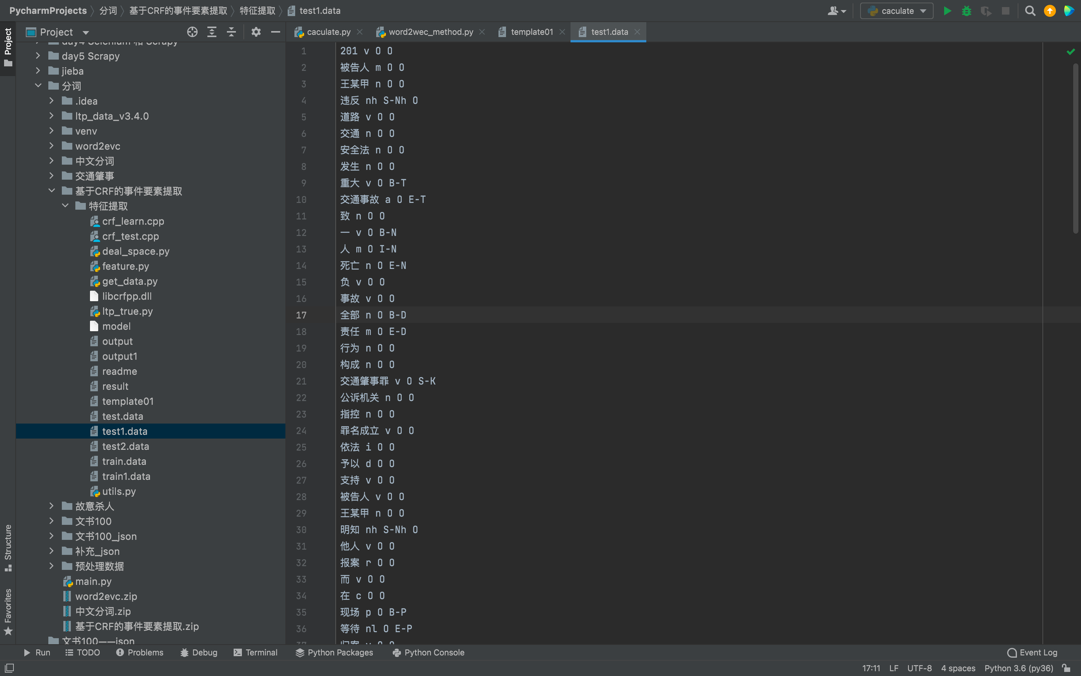
下载CRF++包，通过**crf\_learn**方法，放入模版文件和训练文件，获得model。

以交通肇事案件为例，我们将所需要的事件要素分为事件要素（T）、罪名（K）、主次责任（D）、积极因素（P）、消极因素（N）、判决结果（R）六类。

对训练文件中的每一行分词，都进行了pyltp的**词性标注**和**命名实体标注**。最后一列放入了**BIS标签**对应的特征要素（不存在标为0）。对于BIS标签的标注，我们规定的模版是“词语类别-特征要素类别”，单独词语（S），连续词语分为开始（B）、中间（I）、结束（E）三类。

而在模版文件中，采用了**Unigram**板块的**四元特征训练**，从上下两个特征到相邻的四个特征都进行了考虑。

对于**crf\_learn**方法，设置使用属性的出现次数不少于1，设置1.5为代价参数，防止过大会过度拟合。

图表 8 训练文件截图 图表 9 训练文件截图

#### 案例测试

对需要进行测试的案例文本预处理后，通过**crf\_test**方法，获得输出文件。对输出文件中的每一行，若有特征要素对应的字符出现，就加入对应的字典集，从而获得特征要素的归纳。

#### 补充提取

最后考虑到CRF可能无法完全提取到所有的要素，再对文本进行遍历，对**相关类型的特定词语进行捕捉**，若存在，则加入特征要素。

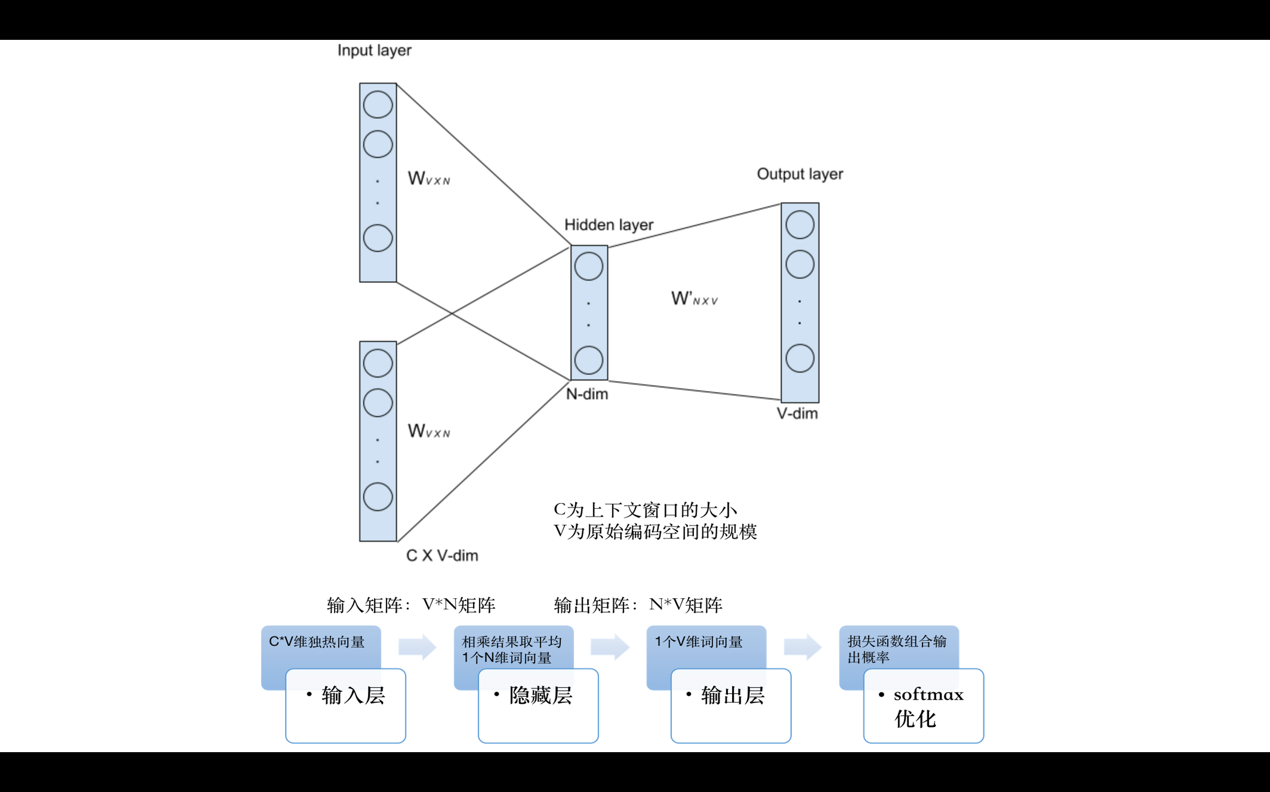
以交通肇事案件为例，我们将“无证驾驶”、“醉酒驾驶”等等常见情况进行了细致的对照，以防出现遗漏。

### 相似案件

#### Word2evc原理

首先是将单个词语从**独热表示**到**词嵌入**模型的转换。独热表示存在无法结合上下文与容易造成维度灾难的问题，因此采用词嵌入模型更好地贴合上下文语境。

其次是**CBOW模型**和skip-gram模型之间的取舍。CBOW模型根据某个中心词前后A个连续的词，来计算该中心词出现的概率，即用上下文预测目标词。鉴于CBOW模型在多个相邻词对一个中心词的预测作用，效率更高，速度更快，并且更加符合此次案例分析中对文段平均词向量的需求，故采取CBOW模型。

得到输出层后我们采用**层次softmax(hierarchical softmax)**来对结果进行优化。得到输出层后采用层次softmax方法来对结果进行优化。该方法引入二叉树中的哈夫曼树，在哈夫曼树中从根结点到叶结点的过程，是二分类问题，这样就把输出层的N分类问题变成 log(N)次二分类问题，通过概率计算和极大似然估计可以获得最终优化好的词向量。

图表 10 CBOW+Softmax原理算法

#### 数据预处理

此时我们所需的文本内容并不是完整的案件，不需要特定的当事人及其身份信息，关注点将集中于事件的**审判过程**，例如案由、审判结果等等，因此我们需要截取案件中的**判决部分**。同时为了词向量的训练和计算，采用**jieba**分词的方法获得分词好的文本内容。

#### 模型训练

借用数据库中处理过的大量案件信息，我们调用python库中的gensim模块，利用其中的**word2evc方法**，建立模型。由于基准语料库较小，我们最小出现词频min\_count为1, 上下文词window为5，词向量维度vector\_size为100。

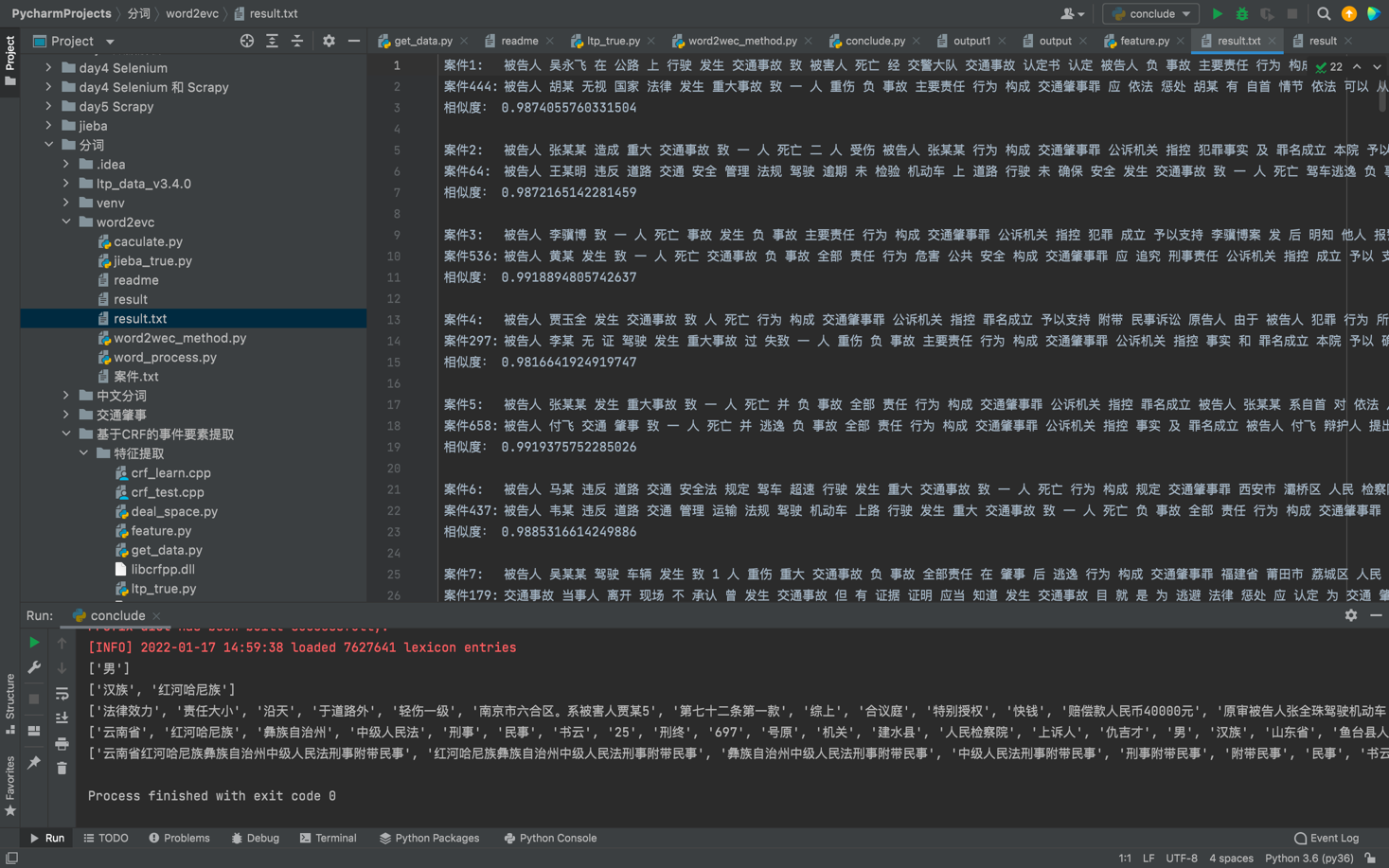
#### 案例计算

传入文本内容，进行数据预处理后，根据模型计算文段**平均词向量**，并且与案件其他数据一起进行存储，公式如下。

得到词向量后，遍历数据库中的案件，使用numpy库的**dot()函数**进行两向量的**内积与夹角计算**，得到的浮点数值则为**相似度大小**，公式如下。

#### 有效性分析

#### 对于数据库中的案例，目前我们认为计算案件所得相似度高，同时匹配效果也较好。



图表 11 相似度分析结果

# 案例分析

## 选取文件

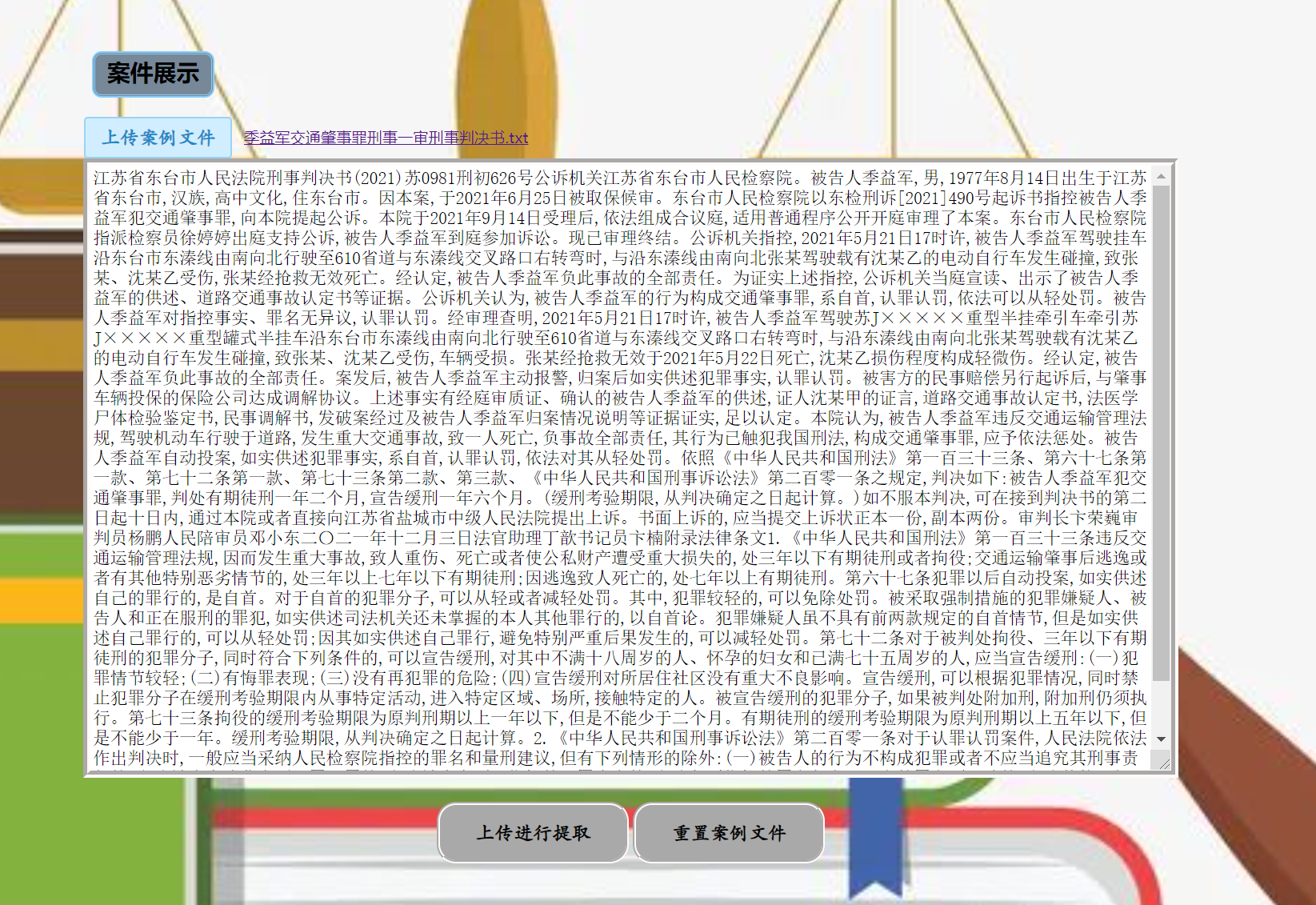
在前端页面选择开始日期及结束日期，关键字为交通肇事，爬取数量为50份。点击爬取案件后，会在目标文件夹中创建一个新的文件夹，命名为关键字+爬取时间，里面存放了爬取到的txt形式的50份文书。



图表 12 选取文件

## 上传并展示文件

点击上传案例文件，从本地选择需要进行分析的文件：季益军交通肇事罪刑事一审刑事判决书，展示在下方文本框内。



图表 13 显示案件

## 勾选分词

点击上传进行提取，展示分词结果。用户以勾选框的方式勾选案件当事人的姓名、性别、民族、籍贯、罪名、相关法院。在这里当事人勾选季益军，性别勾选男等等，完成基本信息的标注。



图表 14 初步勾选分词

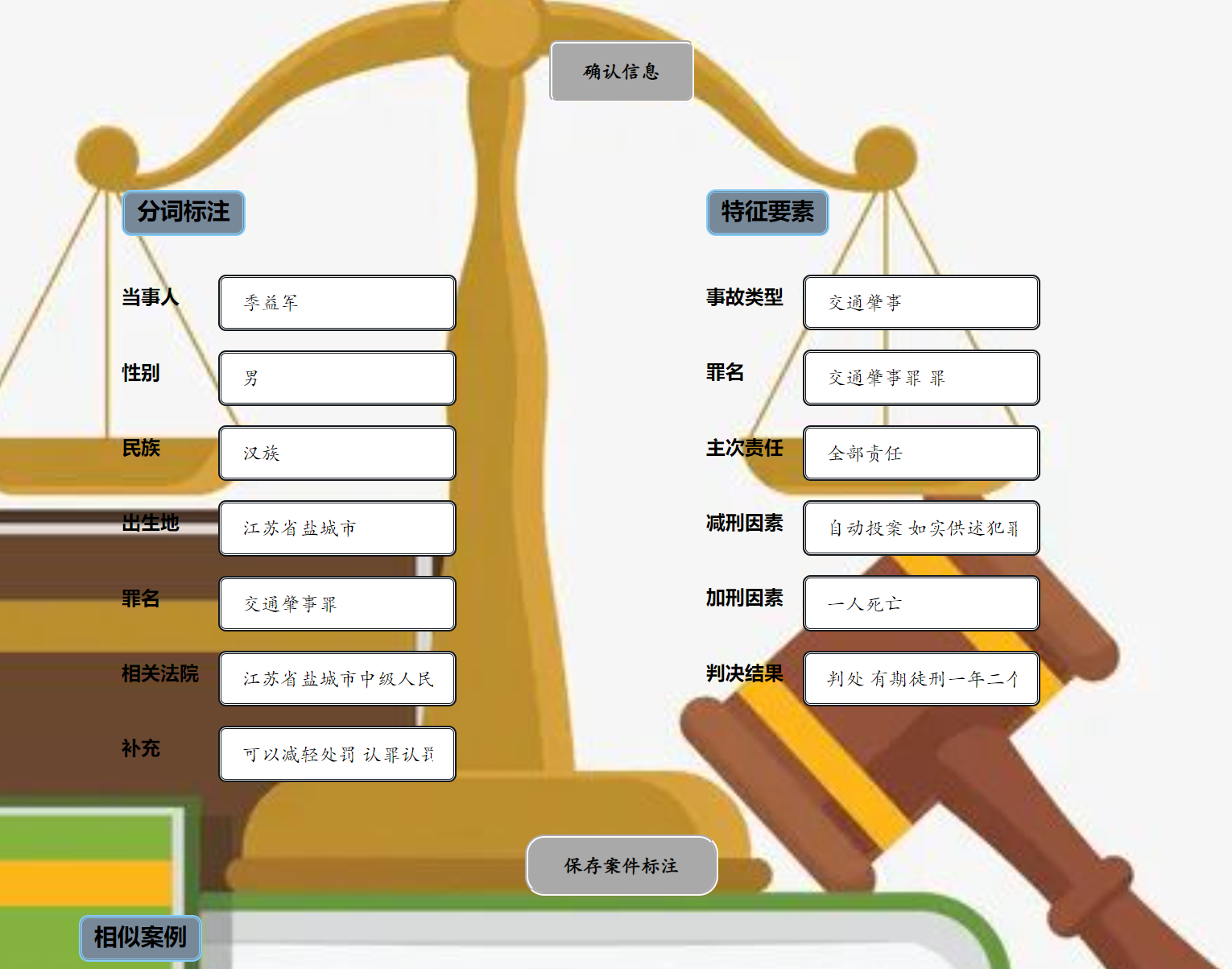
点击点击展开更多的链接，可以查看根据词性分类的更多分词结果并勾选，作为对初步分词的补充。



图表 15 补充分词选项

## 确认分词和特征要素

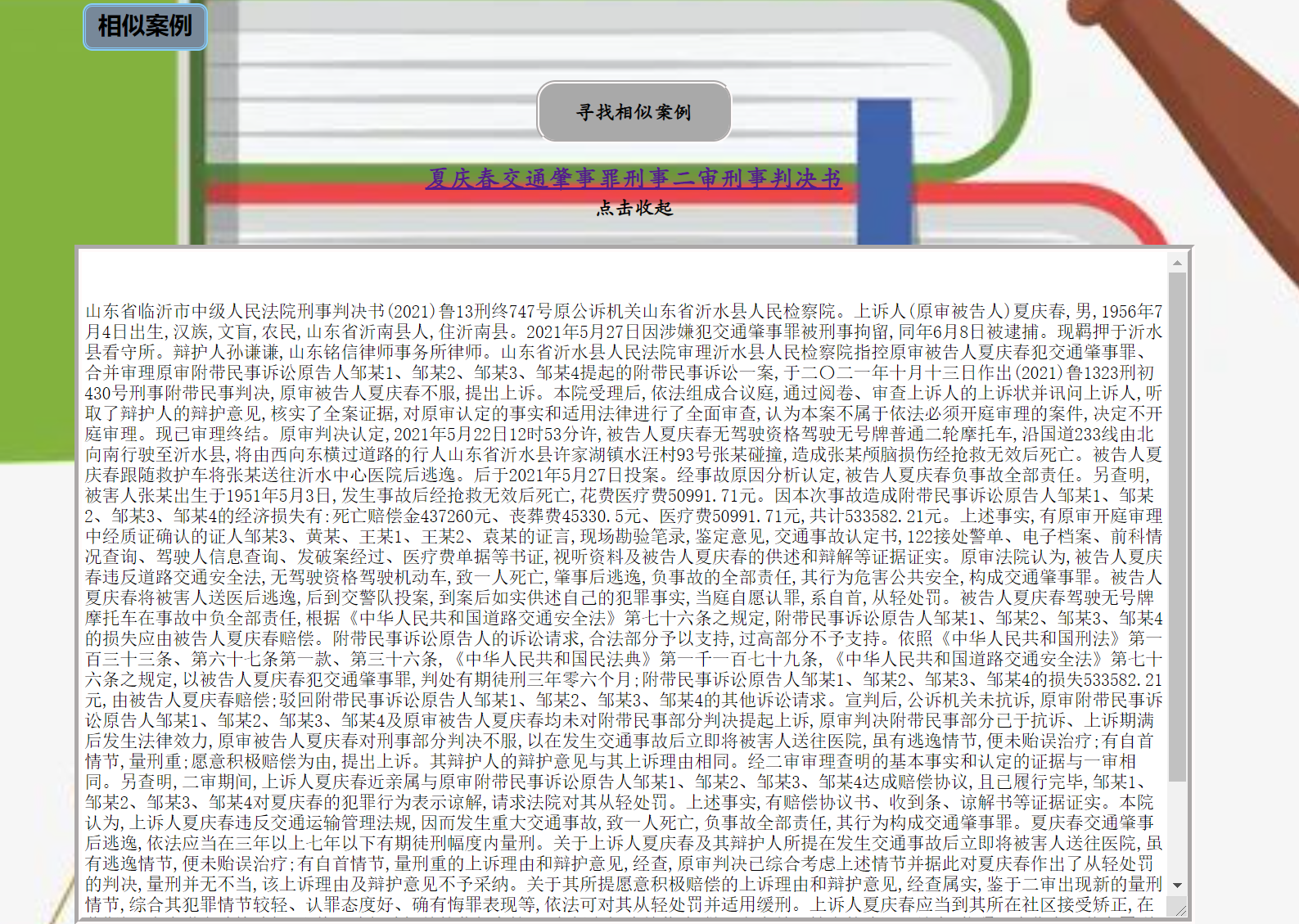
点击确认信息，可以看到分词标注和特征要素的分词结果。若对分词结果不满意，可以进行手动修改。在此例中我们可以将特征要素中罪名中的第二个“罪”字删除，然后点击保存案件标注。案件标注以json文件的形式保存至目标文件夹中。



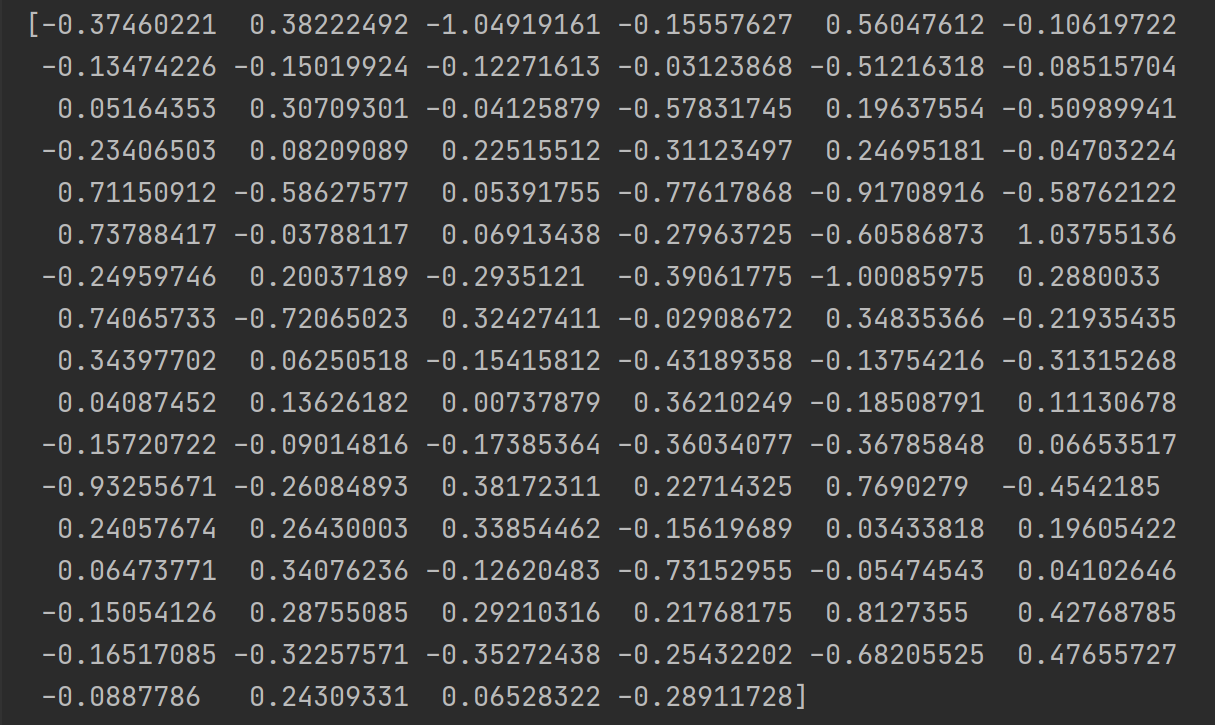
图表 16 确认分词和特征要素

## 获取相似案件

点击寻找相似案例后，下方会弹出数据库中寻找到的最相似案例的标题：夏庆春交通肇事刑事二审刑事判决书，再点击展开，可以将文书内容展示在下方文本框内。



图表 17 展示相似案件



图表 18 季益军交通肇事罪刑事一审刑事判决书的向量值

# 未来展望

1. 希望裁判文书网可以更加稳定和快速，进一步提高爬取文书的速度。

2. 扩大数据库规模，存入数量更多、类型更广泛的文书,提高最相似案例的相似度。

3. 增加训练语料容量，完善模型文件，提高CRF模型预测准确率。

4. 增加扩展功能，例如回归预测案件结果，为审判人员提供参考依据。

# 课程建议

1. 希望课后能够布置对应的书面和编程练习，以此巩固强化学习效率。

2. 希望书后的习题能够提供答案和讲解。

# 参考文献

[1]凤丽洲, 杨贵军, 徐雪等. 基于N-gram的双向匹配中文分词方法[J]. 数理统计与管理, 2020, 39(4):11.

[2]高天蒙. (2017). 微博文本的事件抽取与可视化. (Doctoral dissertation, 东南大学).

[3]张开旭, 夏云庆, 宇航. 基于条件随机场的古汉语自动断句与标点方法[J]. 清华大学学报：自然科学版, 2009(10):4.

[4] 刘稳,王锦,李锐,游景扬,陈建峡,法院判决书关键信息抽取系统设计与实现,湖北工业大学学报,第33卷第2期

[5]熊佳茜. 基于CRF的中文微博交通信息事件抽取. (Doctoral dissertation, 上海交通大学).

[6] Jiang Y , Zhao T , Chai Y , et al. Bidirectional LSTM-CRF models for keyword extraction in Chinese sport news[C]// MIPPR 2019: Pattern Recognition and Computer Vision. 2020.