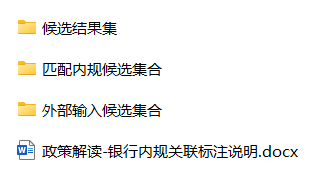
说明手册

1. 数据集说明

本次关联标注的源文档包括271篇银行内部规章制度（简称“内规”）和195篇政策解读（简称“外规”）。内规被保存在《匹配内规候选集合》中，政策解读被保存在《外部输入候选集合》中。关联关系标注表格被保存在《候选结果集》中如下图所示：



每一个关联关系表格是一个excel表格，以外规文件的名称命名，如下图所示。



**每个表格的内容是该外规与271个内规的关系，标注了每篇外规相关的内规有哪些。**具体格式如下所示：

表格

描述已自动生成

第一行的第一个单元格是政策解读的名称；第二个单元格是解读的具体内容

之后的271行的每一行代表该外规与内规的关联关系，格式如下：

**第一格：标注“该解读与对应内规是否相关”，相关标为1，不相关标为0；**

第二格：解读与内规的相似度（不重要）

第三格：内规id

第四格：内规所属部门分类，可在《匹配内规候选集合》中找到

第五格：内规标题

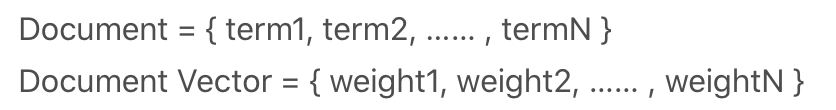
第六格：内规具体内容

1. 技术栈说明

本次示例样本采用基于TF-IDF算法的空间向量模型（[Vector](https://so.csdn.net/so/search?q=Vector&spm=1001.2101.3001.7020" \t "/Users/cyl/Documents\\x/_blank) Space Model，简称VSM）进行IR文本分析，得出文本相似度。

VSM表示通过向量的方式来表征文本。一个文档（Document）被描述为一系列关键词（Term）的向量。

首先我们希望将文档[向量化](https://so.csdn.net/so/search?q=%E5%90%91%E9%87%8F%E5%8C%96&spm=1001.2101.3001.7020" \t "/Users/cyl/Documents\\x/_blank)，可以将一篇文档表示成从该文章抽取的关键词及其权重所构成的向量。该向量由n个词Term组成，每个词都有一个权重（Term Weight），不同的词根据自己在文档中的权重来影响文档相关性的重要程度。



如何提取文章的关键词和他的权重？可以采用Hanlp、jieba分词等分词器对文章进行分词处理，得到文章的关键词，然后利用TF-IDF算法计算每个词的权重。

1、计算TF（词频）：两种标准

TF(t) = (t出现在文档中的次数) / (文档中的term总数)

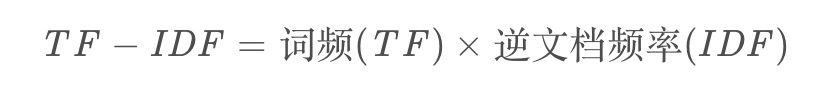
TF(t) = (t出现在文档中的次数) / (文档中出现最多的term)

2、计算IDF（逆文档频率）：用来判断每个词在该文章中的重要程度（权重），如果一个词越常见，那么分母就越大，逆文档频率就越小越接近0。log表示对得到的值取对数。

IDF(t) = log(语料库中的文档总数 / 含有该term的文档总数)

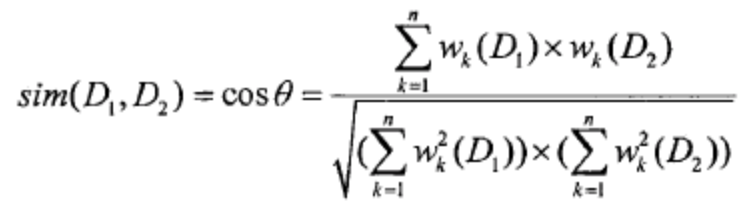
3、计算TF-IDF：这里计算得出的TF-IDF值是文章中一个分词的权重值。

TF-IDF与一个词在文档中的出现次数成正比，与该词在整个语料中的出现次数成反比



依次计算得到你想要查询的外规文档D1=(w1, w2, …, wn)共n个关键词的权重。对于你想要进行判断是否与D1相关的内规文档D2，采用相同的方法计算出D2=(q1, q2, …, qn)，然后计算D1和D2的相似度。

计算两篇文章间的相似度就通过两个向量的余弦夹角cos来描述。文本D1和D2的相似性公式如下：



其中分子表示两个向量的点乘积，分母表示两个向量的模的积。

**总结一下文本相似度分析的步骤**：

1、读取一篇外规文档和要进行相似度匹配的内规文档

2、对两篇篇文档进行分词

3、对分词结果进行预处理（过滤停用词等）

4、统计两篇文档所有的分词

5、分别计算两篇文档对应分词的词频TF

6、计算每个分词的IDF

7、分别计算两篇文档每个分词的TF-IDF值，构成文档的词向量空间

8、带入VSM向量模型公式，得到相似度

9、对所有内规计算完相似度后，依据相似度降序排列，得到一篇外规的相关内规列表

**实例解释**：

句子A：我喜欢看电视，不喜欢看电影。

句子B：我不喜欢看电视，也不喜欢看电影。

第一步，分词。

　　句子A：我/喜欢/看/电视，不/喜欢/看/电影。

　　句子B：我/不/喜欢/看/电视，也/不/喜欢/看/电影。

 第二步，列出两句话所有的分词。

我，喜欢，看，电视，电影，不，也。

第三步，计算词频。

　　句子A：我 1，喜欢 2，看 2，电视 1，电影 1，不 1，也 0。

　　句子B：我 1，喜欢 2，看 2，电视 1，电影 1，不 2，也 1。

 第四步，写出每个词频TF-IDF值向量。

　　句子A：[..., ..., ..., ..., ..., ..., ...]

　　句子B：[..., ..., ..., ..., ..., ..., ...]

第五步，使用余弦公式，我们就可以得到，句子A与句子B的夹角的余弦。

余弦值越接近1，就表明夹角越接近0度，也就是两个向量越相似。

参考：

1. VSM：<https://blog.csdn.net/weixin_43758551/article/details/113918690>
2. HanLP：<https://hanlp.hankcs.com/demos/tok.html>
3. 具体流程
4. 对内规、外规文本进行预处理，主要是冗余信息去除、中文分词和停用词去除。冗余信息去除部分，我们主要排除了文本信息中附加的网址、多余的空白字符、前缀信息等。中文分词我们使用HanLP、jieba分词等进行。（更近一步可以考虑构建特定于内规语料库的字典，用于优化分词）
5. 计算和排序。基于步骤（2）所得的结果，采用VSM模型进行IR文本分析，计算所需查询的外规语句以及内规语料库文本中每个关键词的TF-IDF值，然后将得到的结果代入VSM模型中，得到一篇外规和一条内规条例之间的相似程度，即余弦距离。最后，按照相似度得分高低对这些内规条例进行降序排序。

经过上述步骤，会针对每一篇外规的查询，生成一份内规的排序列表。

1. 评估指标
2. AP(Average Precision)。
3. MAP(Mean Average Precision)。

我们以AP（Average Precision）和MAP（Mean Average Precision）来度量我们文本分析和匹配的结果。关于AP和MAP的介绍可以参考[目标检测中的AP，mAP](https://zhuanlan.zhihu.com/p/88896868)

**更新：**其中，AP的计算我们使用以下公式：

公式中n为一个候选列表中被标记为正确的候选结果数目，k表示找到的第k个被标记为正确的候选结果，k=1,2,3……,n。其中，指得是找到第k个标记时，已经找到的标记个数，即=k；指的是第k个标记在候选列表中的位置。即候选列表的前=i个项中，正确候选结果数量k（即，领域专家标注为“相关”的结果）与i的比值。

例如，一个长度为10候选列表中有3个被标记为1的值，分别位于列表的第1、4、9三个位置，那么：

AP = (1/1 + 2/4 + 3/9) / 3 = 11/18

我们取每个类别查询得到的AP的平均数作为MAP，即

我们认为，每一个政策解读输入是一个类别查询，因此计算AP时，我们需要将每个政策解读查询得到的候选内规序列合并并排序。计算MAP时，我们需要对每个政策解读输入查询得到的候选内规序列分别计算AP，再将AP平均。

**例如**：有三篇政策解读输入1，2，3；内规库中有三篇内规a，b，c；

输入1查询得到序列：a，b，c

输入2查询得到序列：b，c，a

输入3查询得到序列：c，a，b

其中1与a，c相关，2与b相关，3与a，b，c相关。我们以1a表示通过1查询得到的a。

基于上述表示方法，假设合并得到的序列为：1a，1b，2b，2c，1c，3c，2a，3a，3b

由查询和内规的关系可知，关联关系的标注为：1，0，1，0，1，1，0，1，1

则AP = (1/1 + 2/3 + 3/5 + 4/6 + 5/8 + 6/9) / 6

AP\_1 = (1/1 + 2/3) / 2

AP\_2 = (1/1) / 1

AP\_3 = (1/1 + 2/2 + 3/3) / 3

MAP = (AP\_1 + AP\_2 + AP\_3) / 3

我们以这两项指标来度量匹配的效果。AP的意义在于其在度量中代表着相关结果在候选列表中的平均位置，当所有相关结果都在候选列表的最前方时，AP=1；都在最后方时，AP最低。而MAP则代表每个类别的AP的平均值，在此处意味着相关结果在每个输入的政策解读的候选列表中的平均位置。

1. 开发环境说明

推荐使用Java或python，编译器可以采用IntelliJ IDEA、PyCharm、VS Code。

**最终项目的部署需要放在华为云上**，具体可见moodle上提供的相关文档：  
