# 题目查重报告——智慧学伴

1. 背景

当前导出的题目数据中存在相同题目及相似题目，需要筛选去除掉重复的题目。数据中数学学科题型主要分为：单选题、填空题、作图题、解答题四种题型，题目重复的标准分别是题干、选项两部分完全相同、题干相同、题干及问题均相同等，因此这里仅考虑题干内容，以此作为题目重复的依据。

因为当前没有标注数据，需要利用一些现有的文本相似度计算题干相似度，通过人工评价的方式选取最合适的比较方法，然后通过两两比较，筛选重复的题目。

1. 数据预处理

源数据：split.xls 使用其中 “正常” sheet共6678条。

数据特征：单选题题干内容长度较小，题干内容很多为中英文混合出现，数学公式文本表现中含有很多转义符如反斜杠，整体不利于字符串比较。

题目类型共三种：单选、填空、主观题

文本预处理：

1. 去除题干内容含有“\_\_img”的数据，这部分为非文本，图片类型png|jpg|jpeg|bmp|gif
2. 去除不合法字符如空格 换行 空括号 下划线 “.” 等
3. 去除【教师题库】情况
4. 去除（2018九下广东中考）情况
5. 去除题干处理后过短的情况，当前去除文本长度小于2的文本。

补充：对于以上情况4，如果不去掉，其中的信息会产生干扰，对于题干信息的判定会产生较大的影响；如果去掉，有的题干内容很简单，如“下列计算正确的是”，查看数据发现两个题干虽然一样，但是不是同一个题。

综合分析，对于单选题的处理较差，题干一般很短，包含信息很少，题干内容为“如下正确的说法是”，需要讨论是否将单选题类型单独处理。

1. 不适宜去掉 “下列XXX”的信息，尤其对于单选题，去掉后题干长度过短，如去掉下列说法，很多单选题仅剩余“正确的是”、“错误的是”，效果不好。

生成数据：

文本预处理后获取到2947条数据

1. 相似度比较：

这里选取了几种不同的文本相似度比较算法

一、基于字符比较：

* Jaccard通过两个字符串的交集并集的比例来计算相似度。取值区间[0,1]越接近1越相似。
* Levenshtein编辑距离 通过比较两个字符串的编辑距离进行比较。取值区间x>=0越小越相似。
* BM25 将句子s1分词后，对每个词与句子s2计算相关性，最后求和计算两个句子的相关性。取值区间x>=0越大越相似。
* Difflib 计算两个句子的相似度 取值区间[0,1]越接近1越相似。

二、基于向量比较：取值区间[0,1]越接近1越相似

* Word2vec + cosine
* Tf-idf + cosine
* Bert + cosine（结果没跑完）

比较结论：

两两比较的结果，生成文件地址在服务器：

172.24.227.247/home/liujiefei/pythonworkspace/cal\_ques\_similary/result

整体来讲，通过向量计算相似度算法时间复杂度高，耗时长，效果并不是很好。

1，Word2vec + cosine比较效果较差，结果依赖于向量表示。由于数学题干中存在很多公式表示，因此分词效果较差， 转换为向量后进行比较的效果也较差，且转换为向量后对以下情况处理较差。

0.9999954104423523, 497 下列函数中的偶函数是<--->905 下列函数中不是偶函数的是

2，tf-idf +cosine依赖于分词后统计的tf-idf，但效果就我个人来看，此方法相对其他转换向量的方法，结果更精确些。

0.9995223879814148,975 下列说法正确的是<--->944 在同一平面内，有三条直线a、b、c，下列说法中正确的是  
0.9995216727256775,628 下列说法正确的是<--->360 集合A＝{2，－1}，B＝{m²－m，－1}，且A＝B，则实数m＝()

通过字符来计算的方法：

时间复杂度较小，比较结果符合我们判断题干内容是否相同的需求，

1，Jaccard方法最为简单，效果较好，根据结果展示，阈值可以设置为0.75-0.8。

2，Difflib方法效果也比较好，根据结果，阈值可以设置为0.8

3，BM25算法效果最差，推测是由于中英文混合效果不好，还未实际验证；

4，编辑距离效果不符合我们比较相似题目的要求，且阈值不好设置。

1. 结论

推荐使用Difflib 方法或者Jaccard方法来比较题干内容的相似度，结果符合我们的需求。

1. 备注

* 对于数据的处理，需要考虑将单选题于其他三种题目分开处理，单选题的题干内容很多为“下列正确的是：”这种情况，属于无意义描述，需要结合选项内容进行比较。
* 对于题干中包含的公式字符，需要考虑将公式做统一表示，便于字符串进行比较，如果无法实现，在文本预处理中需要单独处理公式部分
* 对于题干数据，当前仅处理文本部分，对于图片部分未考虑。

# 对于编辑距离与Difflib效果的比较

根据上面比较的结论，我们单独对于difflib方法与Levenshtein\_dist编辑距离两种方法做评估，比较在筛选相同问题上的效果

由于题目类型不同，需要区别对待处理，特别是单选题，题干信息一般较短，无法提供有效信息，因此额外引入单选题的答案。

将数据区分为三种类型：

单选题：1140

填空题：396

主观题：1070

对于编辑距离，由于Levenshtein\_dist取值为[1,正无穷)，如果严格比较，认为只有两个题目字符串完全一致才可认为是相同题，那么设置编辑距离为0，difflib为1.0即可。

这里需要将题目比较认为相同题目的情况，可以允许有少量字符查别。因此这里采用difflib与Levenshtein\_ratio值的比较，取值都在[0,,1]，越接近1越相似。

效果比较过程：分别对三种不同题型比较采用不同方法的效果，依次按照Levenshtein\_ratio Levenshtein\_dist difflib计算的相似结果做排序，与另外方法的值的比较

**结果地址服务器/home/liujiefei/pythonworkspace/cal\_ques\_similary/difflib2leven**

**在单选题的比较中，我们发现Levenshtein\_ratio对字符变化更加敏感，便于区分两个字符串的差别，如果我们认为下面两个题目题干相同，选项顺序不同为同一个题，那么我们可以采用difflib的方式来做筛选，如果认为这是两个不同的题，可以采用编辑举例的方式**

**举例1：**

Levenshtein\_ratio：.896551724137931,difflib\_value: 1.0

2073 下列各选项中，线段的长度能组成直角三角形的是（）（）A、1，√2，√3B、2，3，4C、√2，√3，5D、3，5，6

2229 下列各选项中，线段的长度能组成直角三角形的是（）（）A、2，3，4B、1，√2，√3C、√2，√3，5D、3，5，6

举例2：

Levenshtein\_ratio：0.8074534161490683,difflib\_value: 0.8571428571428571

2459 在长为18cm的线段AB上任取一点M，并以线段AM为边作正方形，则这个正方形的面积介于36cm2与81cm2之间的概率为A、5/6B、1/2C、1/3D、1/6

2888 在长为10cm的线段AB上任取一点P，并以线段AP为边作正方形，这个正方形的面积介于25cm2与49cm2之间的概率为A、3/10B、1/5　C、2/5D、4/5

举例3：

Levenshtein\_ratio：0.7681159420289855,difflib\_value: 0.9565217391304348 1376 下列条件中，不能判定一个四边形是平行四边形的是（）A、两组对边分别平行B、两组对边分别相等C、两组对角分别相等D、一组对边平行且另一组对边相等

2211 下列条件中，不能判定四边形是平行四边形的是A、两组对边分别平行B、一组对边平行，另一组对边相等C、两组对边分别相等D、两组对角分别相等

同样在填空题与主观题中也是相同情况

**结论：推荐使用difflib的方式来做题目查重。**

备注：

在实际题目去重过程中，除了题干字符串的比较还可以引入【级主题代码 一级主题 二级主题代码 二级主题 核心概念代码 核心概念】等信息做辅助判别。