# 题目去重方法比较报告——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，需要筛选去除重复的题目。

因为当前没有标注数据，不知道那些题目算相似题目，因此需要将数据在常用的文本相似度方法上计算题目相似度，排序后通过人工评价的方式比较各种文本相似度方法的效果，便于找到最合适的方法用于题目去重的任务。

1. **数据处理**

数据来源：基于源数据解析，选取其中单行标注及非套题等情况的题目，共6678条。

数据特征：题目类型共三种：单选、填空、主观题。

其中单选题题干内容长度较小，通常包含信息较少，如单选题干一般为“下列正确的是”，因此对于单选题，将其答案选项的内容添加到题目信息中一起比较；

填空题与主观题题干内容长度一般较长，但混合中英文，数学公式表示。

文本预处理：

1. 去除图片表示的题目（存储信息为图片地址）。
2. 去除不合法字符如空格、换行、下划线、“□” 等。
3. 去除题目中【教师题库】及（2018九下广东中考）这种不利于区分的字符串。
4. 去除题目描述过短的文本，。

数据处理结果：将数据分为三种类型，其中：单选题：1140；填空题：396；主观题：1070。

对于题目文本比较采用文本相似度比较的算法，主要有两种思路，一种基于字符进行比较，另一种基于向量比较。

下面两部分，分别是从七种相似度计算方法中比较，选取效果较好的两种；另外一部分是比较这两种方法的效果对比。

1. **七种相似度算法比较：**

**结论：经过比较发现基于字符的方法整体比基于向量的方法效果较好。其中基于字符的方法中比较好的两种方法是：基于LCS的difflib方法与基于编辑距离的Levenshtein方法较好。**

这里比较了七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越接近0越相似，后面为了便于比较采用Levenshtein ratio来评估。
* 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 基于LCS的思路计算字符匹配， 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

实验结果记录：

一、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。

相似度：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，但效果也不好。

相似度：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＝()

二、通过字符来计算相似度的方法：时间复杂度较小，效率高。比较结果符合我们判断题干内容是否相同的需求。

1. Difflib方法与编辑距离方法效果较好，下面会比较这两种方法。
2. Jaccard方法最为简单，但效果不好，由于将字符串拆开一一比较，对于混合了公式的数据，表现较差。

3，BM25算法效果最差，基于统计的思想，结果不符合我们对相似题目认定的需求。

1. **基于LCS的difflib方法与基于编辑距离的Levenshtein\_ratio的效果比较**

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，比较在三种不同题目类型数据上的效果。

对于题目描述字符串完全一样的情况，两种方法计算的相似值均为1，没有差别，主要比较在题目描述字符串中有少量字符差异的情况，两种方法计算的文本相似度的值的比较。这里采用difflib与Levenshtein\_ratio值做评估，两种方法计算结果的取值都在[0,1]，越接近1认为两个字符串越相似。

效果比较过程：分别对三种不同题型数据采用两种方法计算相似度的值。

**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，比如例子1中，题干内容一样，选项内容一样，但是选项顺序不一样，Levenshtein方法认为这两个题目的相似度为0.89，difflib方法认为两个字符串的相似度为1，也就是认为两个题目相同。**

**同样在例2 例3中，同样可以发现，difflib对于字符的变化不敏感，对于题目相似度的评估均比Levenshtein高。同样在填空题与主观题数据中也是相同情况。**

例1：

Levenshtein\_ratio：0.896551724137931,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

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

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

**下一步工作：**

**需要具体讨论题目重复的标准，判断题目字符差异多大的情况仍然可以认为是相似题目来处理。**

**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其他维度信息如【一级主题代码 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。