

3.1. 某个IR系统的某次查询返回了13篇相关文档和5篇不相关文档。在文档集中总共有40篇相关文档。问：

a). 计算本次搜索的正确率P和召回率R

正确率：13 / (13 + 5) = 0.72

召回率：13 / 40 = 0.325

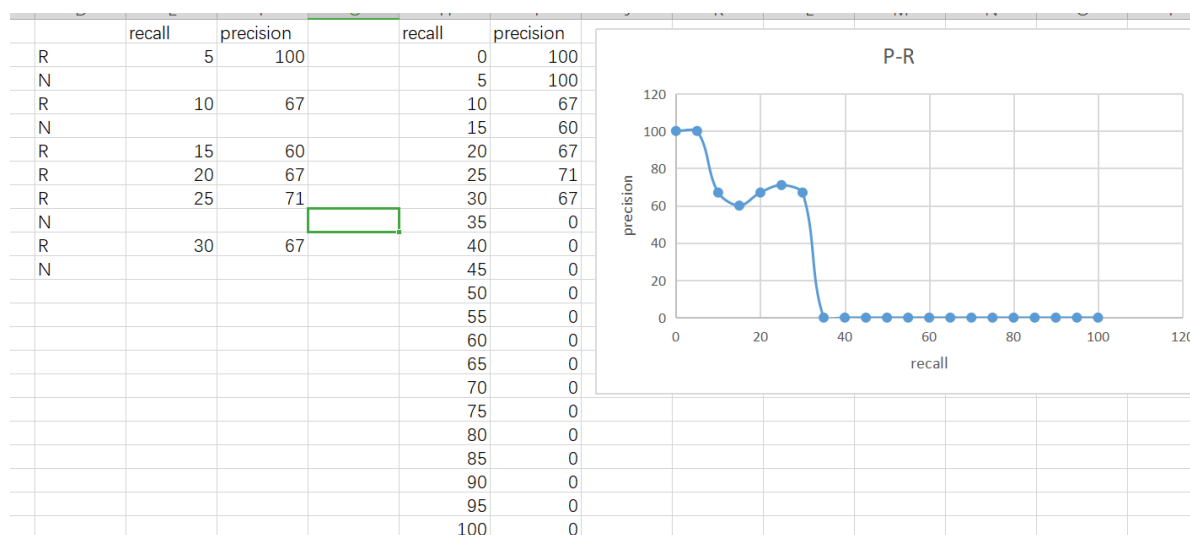
b). 试计算 a=0.2 与 a=0.25 时的F值

$$F = \frac{PR}{aR + (1-a)P}$$

a = 0.2: F = 0.365

a = 0.25: F = 0.377

3.2. 已知有一文档集中共2000篇文档，一个top K查询按顺序返回了10篇评分最高的文档，这10篇文档的相关性情况为：RNRNRRRRNRN，R表示相关文章，N表示不相关文档，从左往右按评分降序排列，已知对于此查询文档集中共有20篇相关文档，试绘制插值P-R（正确率-召回率）曲线。



3.3. 最近，Alice和Bob迷上了吃核桃，但是开核桃并非易事，于是二人上Google搜索砸核桃的方法，输入原始查询：crack nut之后，Bing返回的其中三篇文档标题分别为：Cracked Nut Butter，How to Crack Pecan Nut以及How to Crack Nut Without a Nutcracker，Alice认为第一篇文档不相关，而后两篇相关，此时Bob突发奇想，他想知道原始查询向量，以及使用了Rocchio算法修改后的查询向量是多少。虽然Alice和Bob在玩博弈的时候，总是无限聪明的，但是对IR却是门外汉，于是他们便向你求助，为了简化问题，权重计算时直接使用tf（词项频率，不需要归一化），处理词项时只需要统一大小写即可，并且令 $\alpha=1$, $\beta=2$, $\gamma=0.75$ 。

原始：

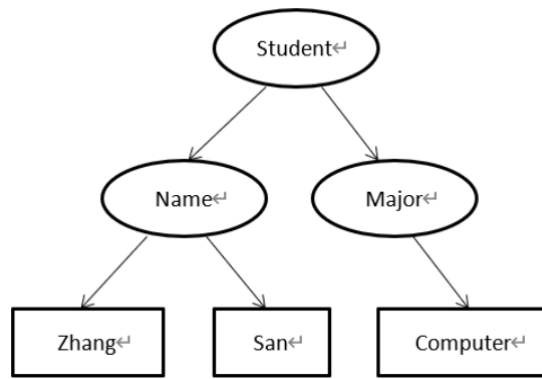
	query	doc1	doc2	doc3
crack	1	0	1	1
nut	1	1	1	1
cracked	0	1	0	0
butter	0	1	0	0
how	0	0	1	1
to	0	0	1	1
pecan	0	0	1	0
without	0	0	0	1
a	0	0	0	1
nutcracker	0	0	0	1

原始查询向量：(1, 1, 0, 0, 0, 0, 0, 0, 0, 0)

$$\begin{aligned}
 \text{Rocchio: } \vec{q}_m &= \alpha \vec{q}_0 + \beta \mu(D_r) - \gamma \mu(D_{nr}) \\
 &= 1 \times (1, 1, 0, 0, 0, 0, 0, 0, 0, 0) + 2 \frac{(2, 2, 0, 0, 2, 2, 1, 1, 1, 1)}{2} - 0.75 \times (0, 1, 1, 1, 0, 0, 0, 0, 0, 0) \\
 &= (3, 2.25, -0.75, -0.75, 2, 2, 1, 1, 1, 1)
 \end{aligned}$$

修改后的查询向量：(3, 2.25, -0.75, -0.75, 2, 2, 1, 1, 1, 1)

3.4. 有XML文档如下（方框为词项），请问：

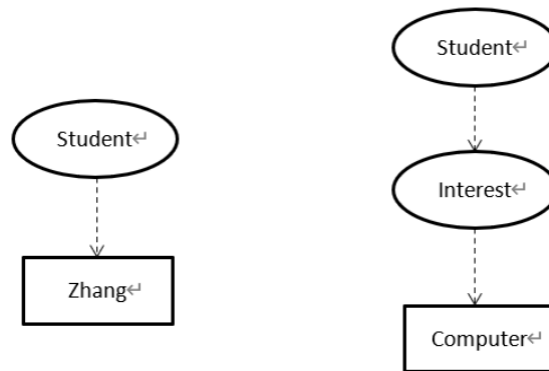


a). 写出其所有结构化词项

总共有9个结构化词项（使用教材的表达形式）：

1. "Zhang"
2. "San"
3. "Computer"
4. /Name#"Zhang"
5. /Major#"San"
6. /Major#"Computer"
7. /Student/Name#"Zhang"
8. /Student/Major#"San"
9. /Student/Major#"Computer"

b). 对于以下两个查询，分别找出与其上下文相似度最高的结构化词项，并给出对应的上下文相似度



/Student#"Zhang":

- 只可以通过插入节点与第七个结构化词项匹配，因此与其相似度最高的结构化词项是 /Student/Name#"Zhang"
- 相似度: $(1+2) / (1+3) = 0.75$

/Student/Interest#"Computer":

- 不可以通过插入节点与任意结构化词项匹配，因此不存在与其相似的结构化词项
- 所有相似度均为0