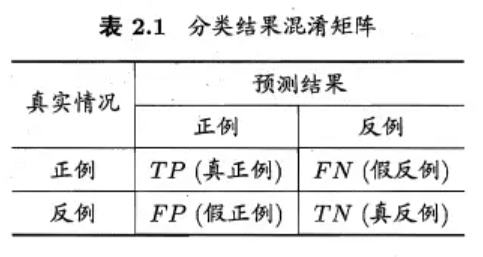
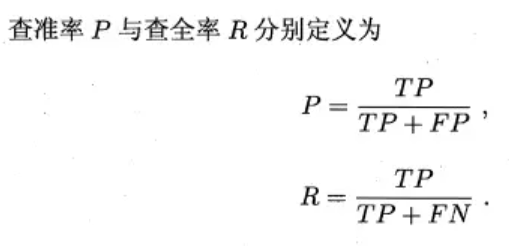
**基础评价指标：**

在场景文字检测任务中，有三个性能评价指标：

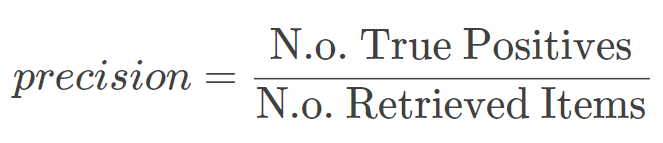
* 精度(P)
* 召回率(R)
* F度量
* 平均正确率
* IoU





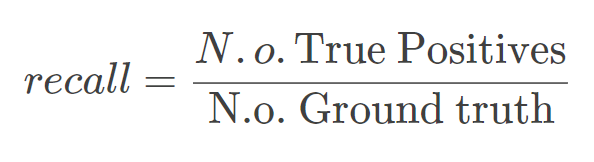
**精度：**

分类正确的样本数占总样本数的比例



**召回率：**

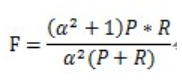
ground truth里的东西被取出来的比例



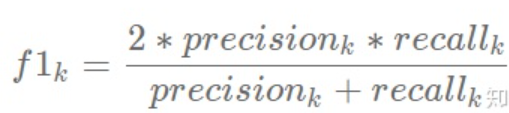
**综合评价指标（F-Measure）**

P和R指标有时候会出现的矛盾的情况，这样就需要综合考虑他们

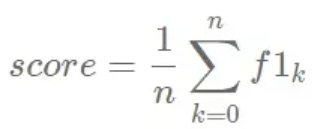
F-Measure是Precision和Recall加权调和平均



当α=1时：



如果类别为多类，则F1-score 求均值：

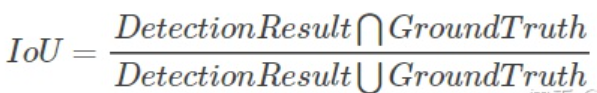


**平均正确率（Average Precision, AP）**

**IoU**

IoU这一值，可以理解为系统预测出来的框与原来图片中标记的框的重合程度。

计算方法即检测结果Detection Result与 Ground Truth 的交集比上它们的并集，即为检测的准确率：



**文本检测有三个挑战：**

* Challenges 1(Born-Digital)的数据来源于电脑制作的，而Challenges 2和Challenges 4(Real Scene)的数据来源于摄像机的拍摄。
* Challenges 2主要是来源于用户有意识的对焦拍摄的(focused text)比如一些翻译的场景，这些场景中文字基本是对焦好的且水平的
* Challenges 4主要来源也是用户拍摄的，但是这些照片的拍摄是比较随意的(incidental text)这样会导致图片里的文字角度、清晰度、大小等情况非常的多。

这些挑战都有对应的**评估方法**：

* Challenges 1和2使用的是叫做 **DetEval**的方法，该方法来自《Object Count / Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms》，ICDAR自己实现了一套Deteval方法

**评估方法链接：**

<https://rrc.cvc.uab.es/?ch=2&com=mymethods&task=1>

* Challenges 4使用的是简单的通过**IoU**来判定算法的recall、precision的。

**评估方法链接**：

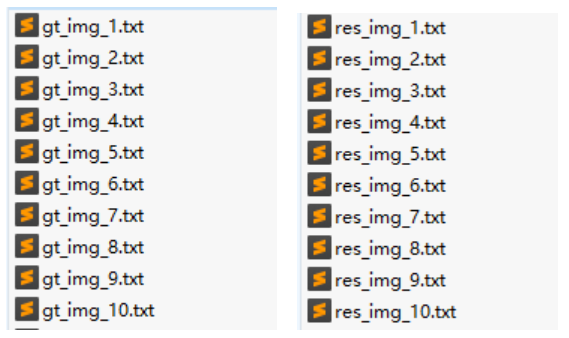
<https://github.com/liuheng92/OCR_EVALUATION/blob/master/Algorithm_IoU.py>

**文件准备**

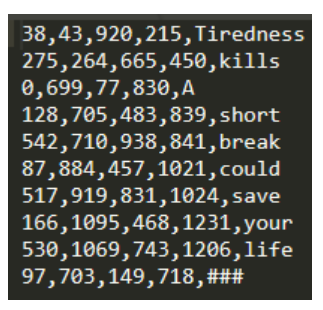
本文的代码是官方实现的Deteval方法即上面链接中的第一个。运行命令如下

python script.py –g=gt.zip –s=submit.zip –o=./ -p={\"IOU\_CONSTRAINT\":0.8}

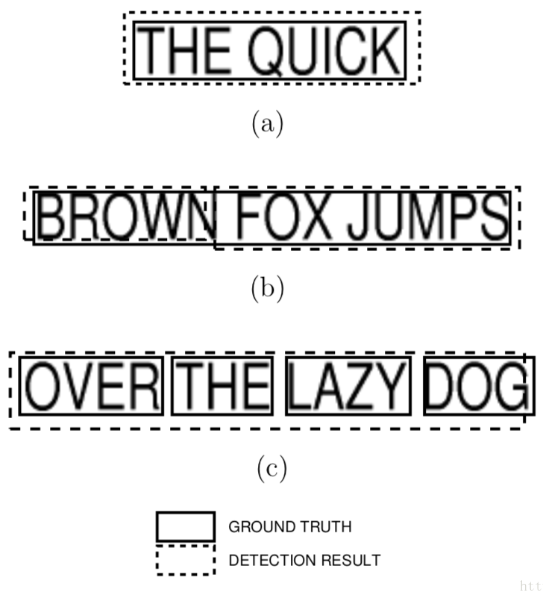
其中gt.zip是标签文件，submit.zip是模型的识别结果，-o是保存result.zip的路径，result.zip中是每张图片的评测结果，-p是评测中用到的一些阈值，在script.py中的default\_evaluation\_params函数中定义了默认值，需要改动也可以在代码中改。因为代码中是按正则的方式读取txt文件，因此txt的名称和内容格式都需要严格按照代码中设定的格式。最方便的方法还是将需要评测的gt.zip和submit.zip转换成代码中的格式而不是去改代码。其中gt.zip和submit.zip中的txt文件名称如下所示



txt内保存水平框的xmin,ymin,xmax,ymax，icdar提供的gt中还有文本的内容



评价识别结果考虑了三种情况：



**一对一**的match，如(a)所示

**一对多**的match，如(b)所示，ground truth 粒度大于detection 粒度时出现的情况。

**多对一**的match，如(c)所示，detection的粒度大于ground truth的粒度

当评测一个ground truth文本实例和一个检测结果实例时，recall是它们的Intersect Area除以GT框的面积，precision是Intersection Area除以检测框的面积。

代码见 .py 文件