**文本检测中的两个挑战：**

* 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

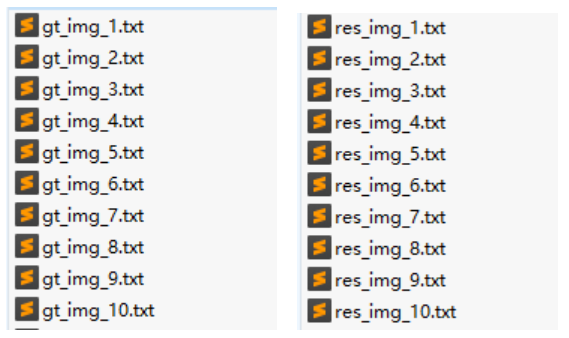
对应的**评估方法**：

使用**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

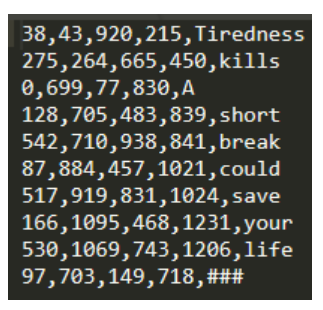
官方链接：<https://rrc.cvc.uab.es/?ch=2&com=mymethods&task=1>

**文件准备**

gt.zip是标签文件，submit.zip是模型的识别结果, 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。gt.zip和submit.zip中的txt文件名称如下所示

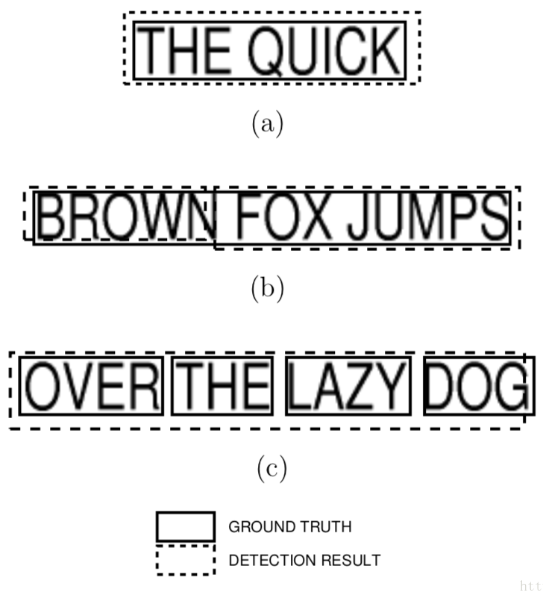


txt内保存水平框的xmin,ymin,xmax,ymax，icdar提供的gt中还有文本的内容



目前我们使用的是**PPOCR**格式的txt，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（如上图所示）而不是去改代码。

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

**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。

代码中 **default\_evaluation\_params**是用到的一些阈值和参数，**validate\_data**是检测传入的gt.zip和submit.zip是否按照要求的格式，**evaluate\_method**则是评价检测结果的函数

**具体做法：**

1. 创建n × m大小的两个矩阵分别叫做recallMat、precisionMat，其中n为标定框(ground truth)的个数，m为检测到的框个数



1. recallMat中存储的是每个检测框的召回率，计算方法为=

其中area()函数表示求矩形的面积，inter()函数表示求两个矩形的交集，gti表示

第i个标定框，detj 表示第j个标定框

b) precisionMat中存储的是每个检测框的准确率，计算方法为



1. 在考虑三种情况之前，先要定义两个域值这里称为r和p，r表示判断召回率的阈值这里设为0.8，p表示判断准确率的阈值这里设为0.4，recall用来存储召回率，precision用来存储准确率
2. **one-to-one matches**，如果在recallMat和precisionMat中的i行只有一个值大于阈值，j列中也只有一个值大于阈值，且这个值在第i行第j列，那么就认为gti与deti是one-to-one matches。如果gti与deti满足一定的条件就将recall和precision加1

a) 这里说的一定条件是指两个框的中心点距离与两个框对角线平均值的比例要小于阈值1

1. **one-to-many matches**，对于precisionMat中如果i行中有多个值大于p，将对应于

recallMat位置的值相加(比如precisionMatij大于p则将recallMatij相加)，用最后的和与r比较，如果大于r就符合one-to-many matches的条件，one就是gti，many就是i行中所有大于p值的列对应的框。如果满足one-to-many matches就将recall加0.8，precision加0.8 × num，num表示对应与gti匹配的所有many框的个数(其实就是many的具体值)

​

1. **many-to-one matches**，这里与one-to-many matches类似，只是先判断recallMall中j列，如果j列中有大于r的值，则将对应于precisionMat位置的值相加，用最后的和与p比较，如果大于p就认为符合many-to-one matches的条件，one就是指detj，many值的是j列中所有大于r值对应的框。如果满足many-to-one matches就将recall加1\*y，(与上述x相同，y表示many的具体值)，precision加1

​

1. 最后用recall除以所有的gt个数(这个个数不是n, n代表该张图片所有标定框的个数，但是计算的时候会将文本标定为###的框去除)，同理precision也会除以所有的det的个数(这个个数也不为m，如果有检测框检测到###区域，这个检测框也认为无效)。f-score也就是hmean算法为recall和precision的调和平均数。

注：上面说明的是一张图片的recall和precision计算方法，如果是整个数据集也是类似，只是先将第5步求出的recall和precision相加最后除以整个数据集的gt个数和det个数。