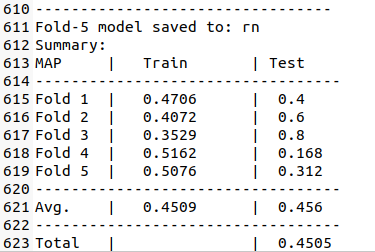
java -jar RankLib-2.18.jar -train 49\_SBFL/SBFL\_chart\_new.txt -ranker 1 -kcv 5 -kcvmd 49\_SBFL\_result/chart/ -kcvmn rn -metric2t MAP -metric2T MAP -tvs 0.8 >> chart\_RN\_train.txt

結果如下

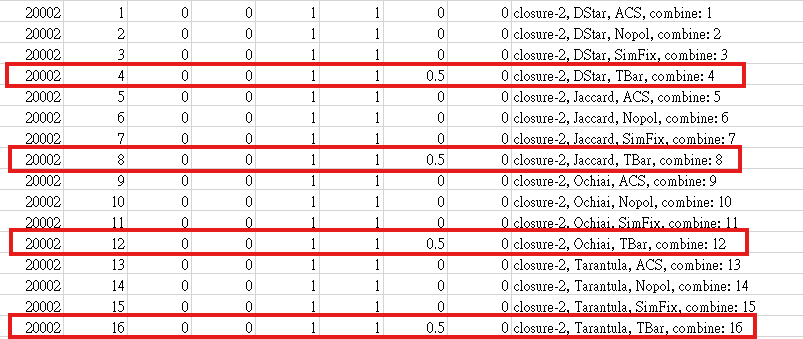


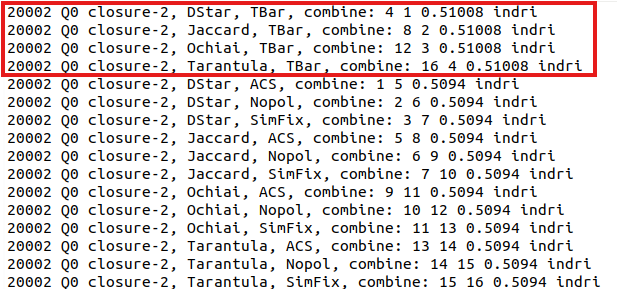
將模型用於closure討論，使用的資料為尚未擴增的16種組合的closure。

java -jar RankLib-2.18.jar -load 49\_SBFL\_result/chart/f4.rn -rank normalize/SBFL\_closure.txt -indri chart\_f2.rn\_in\_closure\_result.txt

根據原始資料，closure-2沒有正確patch，但是有plausible patch，所以應該要排在前面。

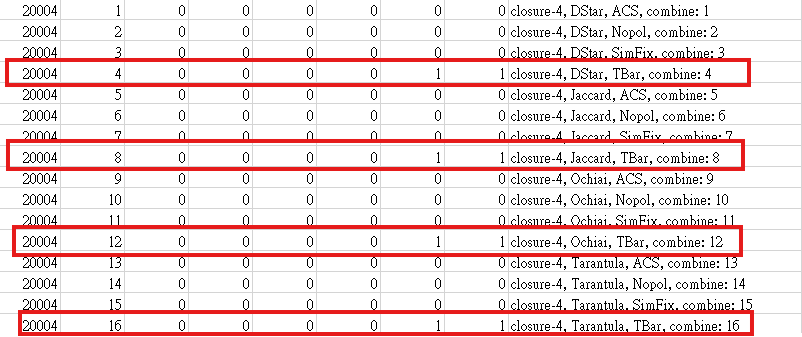
結果表明確實有比較好一點。

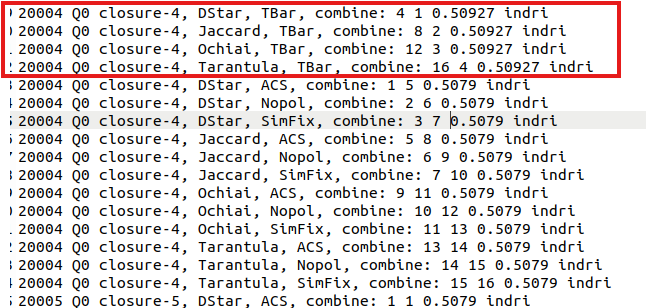




而根據closure-4的原始資料，沒有任何的SBFL在Top10，但是有生成correct patch。

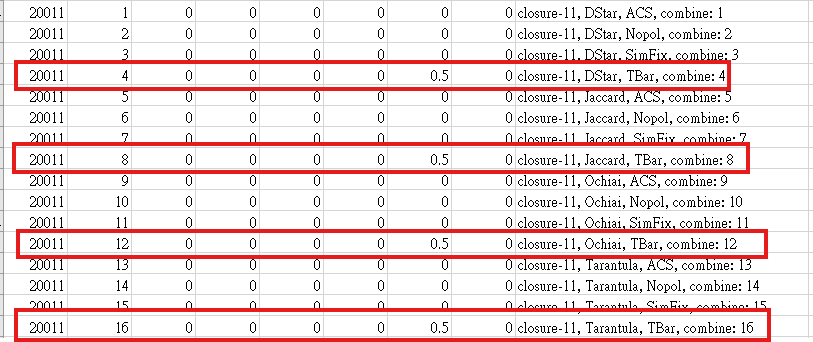
確實有將正確生成patch的排在前面，但是因為沒有任何的top，因此分數的優先序也是比較低的。

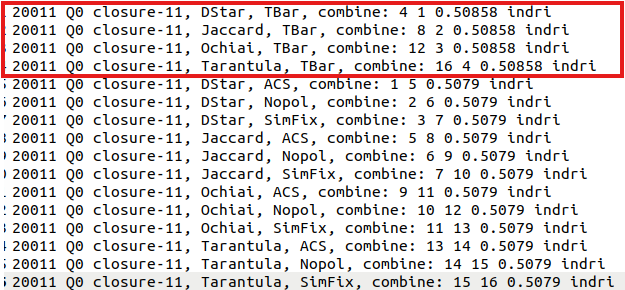




根據原始資料，closure-11是有找到SBFL的TOP結果但是沒有任何patch的情況。

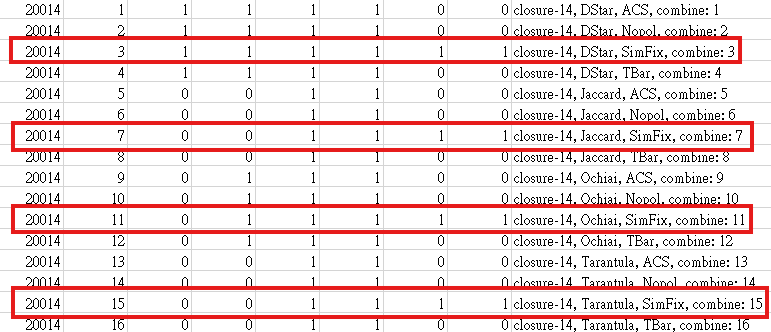
有正確排出，但是也有一個相同的情況，因為沒有SBFL的結果，本實驗主要是討論SBFL與APR的配合，這樣就只有單純的APR結果，互相配合的情況是沒有的，因此結果差距不大。

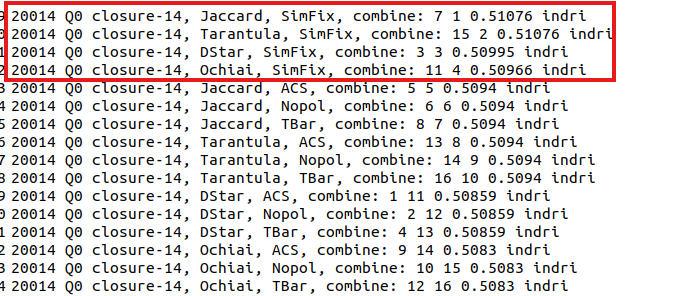




討論closure-14，SBFL有成功定位，且有正確patch的產生。

這邊有個問題，根據SBFL的情況來看，應該是要3最好，其次11，再來跟7跟15。但是結果來看，怪怪的，還是沒有解決之前的問題。





解決想法:

1: 將top的數字越大的數值影響越大，例如如果是在top1的話，那他的Top-1、3、5、10的數值都設定為1;而如果只有在top3的話，那就Top-3、5、10就設定為0.8，這樣來改feature的影響度，不然top越高的影響好像看不太到。

2: 更改id呈現方式，目前訓練資料的qid是以project的bugid表示，改成以SBFL作為qid的編號作為訓練資料。或是說將其作為另一個結果比較，最後會有兩種的資料作呈現。

