目前應該確定模型需要甚麼資料來進行訓練，看看如何套在模型上面。

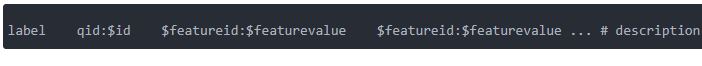
**用Ranklib**

[**https://sourceforge.net/p/lemur/wiki/RankLib/**](https://sourceforge.net/p/lemur/wiki/RankLib/)

**->** [**https://blog.csdn.net/super\_chiry/article/details/103434474?spm=1001.2101.3001.6650.2&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-2-103434474-blog-42263647.235%5Ev43%5Epc\_blog\_bottom\_relevance\_base7&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-2-103434474-blog-42263647.235%5Ev43%5Epc\_blog\_bottom\_relevance\_base7&utm\_relevant\_index=3**](https://blog.csdn.net/super_chiry/article/details/103434474?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-2-103434474-blog-42263647.235%5Ev43%5Epc_blog_bottom_relevance_base7&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-2-103434474-blog-42263647.235%5Ev43%5Epc_blog_bottom_relevance_base7&utm_relevant_index=3)

**->** [**https://blog.51cto.com/u\_16099215/8919744**](https://blog.51cto.com/u_16099215/8919744)

**->** [**https://blog.csdn.net/chikily\_yongfeng/article/details/81396607**](https://blog.csdn.net/chikily_yongfeng/article/details/81396607)

****

****

**應該就是下面準備的資料樣式就長這樣**

**Label 就是label**

**Qid就是version**

**Featureid 就是使用的feature**

**Featurevalue就是這個feature的值是甚麼**

**#後面通常都是無視**

**Description就是**

2024/08/14-16:26

後面又有發現指令中的label是表示相關度，數字越高表示這個結果如上面圖片的1A相關度是3，是qid=1裡面最相關的([由這邊得知](https://blog.csdn.net/chikily_yongfeng/article/details/81396607))**，因此還需要再修正，可能是修復正確的數量作為這個相關性的label(不確定待確認，但是認為目的就是要找出正確patch多的組合，因此我認這樣子設計應該是合理的)。**

**以及LIBSVM**

**XGBoost**

**LightGBM**

* [**https://github.com/jiangnanboy/learning\_to\_rank**](https://github.com/jiangnanboy/learning_to_rank)

**這幾種方法試試看**

可能會先整理出需要的feature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fault Element是否在top-1 | Fault Element是否在top-3 | Fault Element是否在top-5 | Fault Element是否在top-10 | 生成Patch 0(無)  是否正確 1、2 | Patch數量 |

Label標示使用0~N(FL種類)\*M(APR種類)來表示這個錯誤用哪個FL配合APR會比較好。

例如

0就是Jaccard配上Tbar

1就是Ochiai配上Tbar

以此類推作為最後的label標示。

**2024/08/14-16:26**

**可能要改成正確的patch數量作為label，而原本的FL與APR配對改成feature(不確定待確認)。**

chart

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bugid i | Com. j | top-1 | top-3 | top-5 | top-10 | 生成/正確 | Patch數量/label | ~~label~~ |  |
| 1 | 1 | 1 | 1 | 0 | 0 | 2 | 2 | ~~0~~ |  |
| 1 | 2 | 1 | 0 | 0 | 0 | 1 | 3 | ~~1~~ |  |
| 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | ~~2~~ |  |
|  |  |  |  |  |  |  |  |  |  |

例如

chart-1的第1個combination(例如jaccard配上Tbar)，錯誤定位後有在top-1、3標示為1，沒有在5、10標示為0，patch是正確的標示為2，有2個patch生成，也是作為label使用~~，label是這個結果是用哪種組合得出的結果~~。

chart-1的第2個combination，錯誤定位後有在top-1標示為1，沒有在3、5、10標示為0，patch是錯誤的標示為1，有3個patch生成，也是作為label使用~~，label是這個結果是用哪種組合得出的結果~~。

chart-1的第3個combination，錯誤定位後沒有在top1、3、5、10出現，也沒有生成出patch標示為0，也是作為label使用~~，label是這個結果是用哪種組合得出的結果~~。

**我的理解是會是一組向量，例如:{1,1,0,0,1}，label是0，類似這樣的表現。**

**可能還需要再修正。**

**2024/08/14-13:00**

**問題:**

**目前可能會有在chart-1使用Jaccard配上Tbar會有多可patch這種可能。**

**因此還需要再修正。**

用這些來預測chart-1用哪個組合的相關程度，依序排出結果。

如果錯誤定位有找到，但是APR沒有出現結果那就不會作為訓練資料。

所以訓練資料會呈現，

Chart-2

FL-APR組合0 ->

.

.

.

.

Chart-1

FL-APR組合0 ->

.

.

.

.

2024/08/16

Ranklib

<https://github.com/sin-of-sloth/learning-to-rank> -> 可能有用

<https://github.com/ehsangolshani/ranklib-demo>

首先安裝elasticsearch跟request

Elasticsearch:

<https://blog.csdn.net/UbuntuTouch/article/details/124294773>

2024/08/21

目前已經將環境準備完成。

之後要處理實驗數據的準備

準備的方法可以是:

寫一個python程式

使用現有工具，如:RankLibDataFormatter這類型的關鍵字搜尋

使用pandas結合。

**因此目前考慮是將資料整理成一個excel檔案，並在寫一個程式碼將這些資料整理出來。**

2024/08/23

目前有一些資料是因為Tbar無法執行而放棄的

Chart 23

Closure 28

Math 12 104

Lang 23 25 29 56

Mockito整個都錯誤無法執行。

需要的資料如下  
例如這是chart的資料

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bugid i | Com. j | top-1 | top-3 | top-5 | top-10 | 生成/正確 | Patch數量 | label |  |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 2 | 0 |  |
| 1 | 2 | 1 | 0 | 0 | 0 | 2 | 3 | 1 |  |
| 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |  |
|  |  |  |  |  |  |  |  |  |  |

Bugid就是像chart1

Com. 就像是jaccard配上Tbar

Combine(目前共有11FL與4APR，44種)

Top-1就是用這個組合的top1有找到多少用一個數字表示

Top-3同上

Top-5同上

Top-10同上

生成/正確 沒有生成標示為0，有生成但是沒有正確的標示為1，有生成且正確標示為2，因此有越高越好的概念。

Patch數量，就是用於表示正確的數量，越高越好，因此用來表示label越高越好，用在LTR應該是合適的。

20240913

Combine-all編號為combine的excel

Combine-all -> 資料整理-all的excel

Combine-SBFL

Combine-LTR

20241021

可能的參考資料: <https://www.cnblogs.com/memento/p/9398047.html>

目前預計用10-fold來試試看會不會讓結果更好，所以先去找能夠參考的資料。

訓練集與測試集

預計採用如:chart1~25訓練chart26驗證，這種方法

目前觀察到的樣子

fold1:

train:10002~15925

validate:15928~18218

test: 18219~19997

fold2

train:11909~18218

validate:18219~19997

test:10002~11893

看起來跟我剛剛提到的概念差不多

所以應該是把它分成

S1到S5(每個S大小依照不同project的bug-id數量而定)

Fold Training.txt Validation.txt Test.txt

Fold1 S1, S2, S3 S4 S5

Fold2 S2, S3, S4 S5 S1

Fold3 S3, S4, S5 S1 S2

Fold4 S4, S5, S1 S2 S3

Fold5 S5, S1, S2 S3 S4

20241023

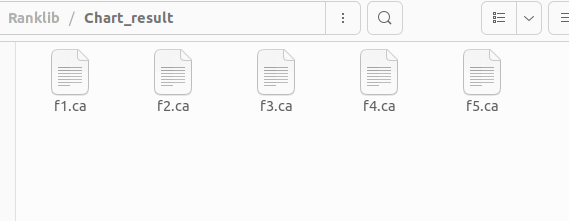
再次修正的紀錄: <https://docs.google.com/document/d/1-uKg26TAJuJ4FfZZ9O3z-ZLlm8TjqkpzQKAqSSVV5F4/edit?tab=t.0>

在20241021說的fold方法是錯誤的。

目前就是一樣把所有的資料作為一個txt，使用這個指令能夠自動分成5-fold做k-fold。

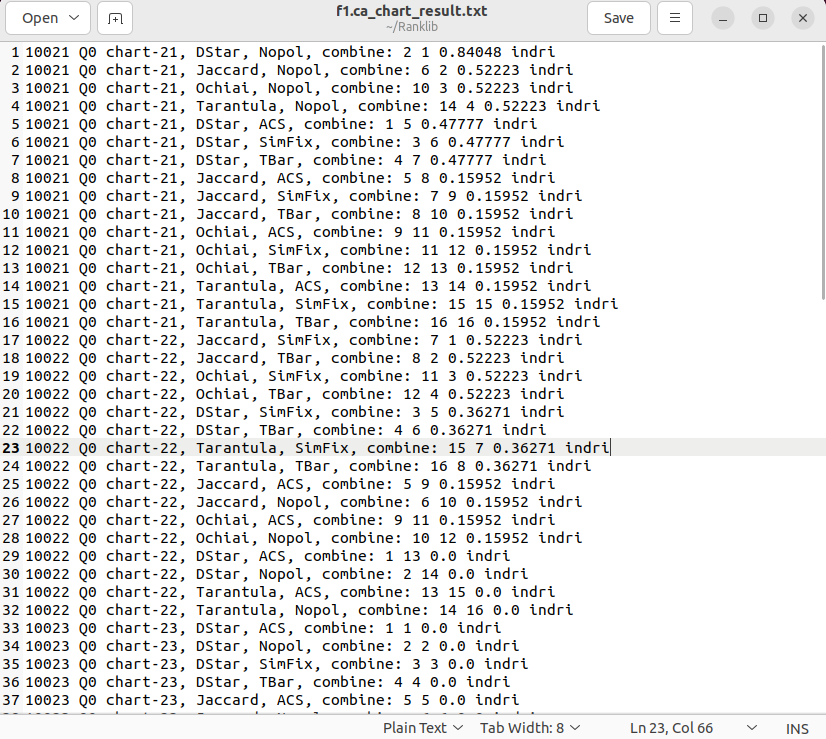
“java -jar RankLib-2.18.jar -train Chart/Chart\_all.txt -ranker 4 -r 6 -kcv 5 -kcvmd Chart\_result/ -kcvmn ca -metric2t NDCG@10 -metric2T ERR@10 -tvs 0.8”

輸出的結果長這樣



然後直接這樣使用

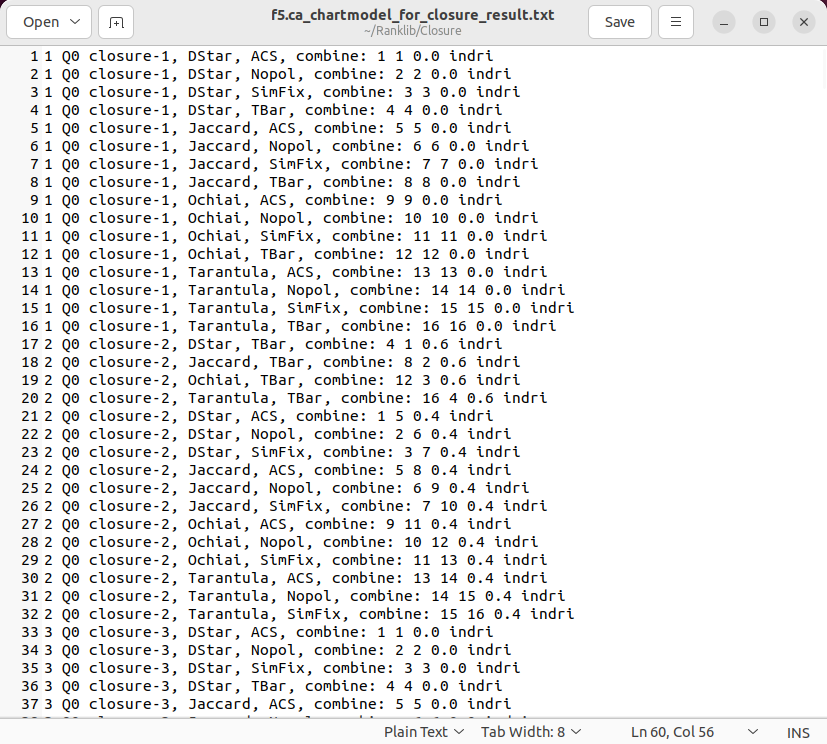
“java -jar RankLib-2.18.jar -load Chart\_result/f1.ca -rank Chart/Fold1/test.txt -indri f1.ca\_chart\_result.txt”



目前發現應該就是把訓練資料分成5分，其中4分作為訓練集，1種做為測試集，然後得出的不同fold下的模型，我們再各自執行看看結果如何。

接下來把chart的執行結果拿來預測closure

“java -jar RankLib-2.18.jar -load Chart\_result/f5.ca -rank Closure/SBFL\_closure.txt -indri f5.ca\_chartmodel\_for\_closure\_result.txt”



根據結果來看，確實能夠將有正確的執行結果的組合放在前面。

目前的問題:  
是要預測一個整體的project用哪個FL技術與APR的結果是最好的嗎?

要不要換一個預測模式，目前的預測模式是每個project的每個bug-id的不同FL與APR的組合作討論，並討論各別的bug-id用哪個組合最好。

想說要不要改其他的資料做為資料去訓練，但是目前還沒有想到好的方法。