Assignment 2

Evaluation and Fusion

3 scripts or applications. language of your choice but no additional libraries!

submission date: End of term.

Part 1:

You are required to write an application or script in the language of your choice but without any additional libraries beyond those that come with the basic installation.

Your application/script must run from the command line or shell prompt and accept one argument.

The argument will be the name of a file that will be stored in the same directory as the application. And will contain retrieval engine results.

Eg: assignment2 sourcefile.txt

The source file will contain a list of Information retrieval engine results in the form of

1;A;RNNRUNNRURRRUNUNNNNRR;10

1;B;RUURRNNRNNNNRRURUNNRN;15

2;A;RNRRRNNNRUUUUNNRRNRRN;10

2;B;RNNNNRURUNNRUNRRNNNNR;12

Etc

These represent 2 test runs from each engine A and Engine B. There will be a maximum of 26 engines represented by a single letter and a maximum of 3 runs from each engine.

Where R = a relevant result, N = a non-relevant result , U = an unknown result

Ranking is from left to right in descending order (1 2 3 4 5 6 ….)

Each Run will be on a separate line and will be preceded with the number of the run and the letter of the engine it was taken from. The final number is the number of relevant documents in the corpus for that query run.

Output: given the input your script/application must calculate the performance using the following techniques to compare :

- Precision
- Recall
- P@5
- P@R=0.5
- Average Precision
- Mean Average precision
- Inverted index output as a list of 11 Precision values, 1 at each of the recall thresholds.

Your script must output these measurements for each engine query run and for each engine overall (in the case of average precision) . You must output the top three engines (not query runs!) to a single file that is clearly laid out and titled.

You must explain your ranking and evaluation system in a readme document that also describes how your application or script must be compiled and run.

Applications must compile without editing. You must state which language version is required and what OS is required (linux or windows, no mac).

Output must be to console and also saved to file.

Code must be commented.

Part 2:

You are required to write an application or script in the language of your choice but without any additional libraries beyond those that come with the basic installation.

Your script or application must run from command line or shell and take in two arguments.

The first argument will be a set of IR engine results in the format:

Weight;Engine#;Doc#;rank_score   weight;engine#;Doc#;rank_score   weight;engine;Doc#;rank_score etc

With one rank per line ranked from most relevant to least relevant.

Eg: (each column is tab delimited)

```
1;100;1         2;24;1          3;5;1000        4;24;900

1;233;0.9       2;13;0.92       3;84;901        4;12;850
```

Engine number will not change in a column, nor will weight.

Document number can be any number between 1 and 1500

Rank score can be any umber but will always decrease or remain constant as the rows increment. Two documents can have the same rank score.

There is no set number of results per engine, your script/application will need to cut each result set off at a fair point or at the shortest result set available.

The second argument will be a list of document Engine IDs and weights in the format

engineID;weight[tab]engineID;weight

eg: A;1.0   B;1.2

the weight will always be a two digit number with one decimal place ranging from 0.1 to 9.9

Your script must use :

1. Interleaving
2. CombSUM
3. LCM

to produce a set of three top 100 documents, one for each fusion technique.

Task 3:

You must write a script or application to run from command line or shell with no graphics based user interface or menu options.

Your script or application must accept two arguments. The first will be the training data, the second will be the live data.

When executed your script or application must generate a probfuse model from the training data which will be given in the format:

1;A;RNNRUNNRURRRUNUNNNNNRR;10

1;B;RUURRNNRNNNNRRURUNNNRN;15

2;A;RNRRRNNNRUUUUNNRRNRRN;10

2;B;RNNNNRURUNNRUNRRNNNNR;12

These represent 2 test runs from each engine A and Engine B. There will be a maximum of 26 engines represented by a single letter and a maximum of 20 runs from each engine.

Where R = a relevant result, N = a non-relevant result , U = an unknown result

Ranking is from left to right in descending order (1 2 3 4 5 6 ….)

Each Run will be on a separate line and will be preceded with the number of the run and the letter of the engine it was taken from. The final number is the number of relevant documents in the corpus for that query run.

The second argument will be a single number in the range 3 to 20 . This will be the number of sectors you must use in your model.

Once the probfuse model is complete you must apply it to the file provided in the third argument.

The Live data file will give a list of Document IDs and the Engine ID they were produced by.

Your script or application must take the live results and by applying the model developed by the training phase, generate a ranked list of a top 20 documents from the results provided.

example live file:

```
EngineID;[DocList]

A;[15,22,1,56,38.....]
```

example run:

```
script_part3  trainingfile  5  livefile
```

Grading:

Part 1: 30 marks

Part 2: 30 marks

Part 3: 30 marks

Overall presentation of results and readme/code: 10 marks

code requires editing to run: -10 marks

code requires editing to compile: -10 marks

code uses non-standard library: -10 marks per part up to a total of -30

While there are no marks awarded for speed, your script /application must run in an acceptable length of time. ie: minutes, not hours.

all input files will be in the same folder as the executable.

All output files should be to the same directory.

All console output should also be written to file.

All code MUST be commented. if you don't comment your code, I will not correct it.