Task 1.

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.

Task 2.

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 number but will always decrease or remain constant as the rows increment.

Two documents can have the same rank score.

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.

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;RNNRUNNRURRRUNUNNNNRR;10

1;B;RUURRNNRNNNNRRURUNNRN;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