EVAL

Initialize

Coding convention.

```
## ------------------------------------------------------------------------------------------
## NAMING:
## ------------------------------------------------------------------------------------------
## T = Test-collection
## M = Measure
## Q = Query
## S = Score
## A = Algorithm (that which is usually 'system')
## ------------------------------------------------------------------------------------------
## VARIABLE PREFIX:
## ------------------------------------------------------------------------------------------
## v = vector
## l = list
## m = matrix
## a = array
## s = string (should be 'c' as in 'character' for R?)
## d = data frame
## w/l = wide/long table format
## f = file name (then what is a string?)
## ------------------------------------------------------------------------------------------
```

Required libraries.

```
library("reshape2")
library("ggplot2")
```

Functions.

Read treceval file into a table, which happens to be in long-format, but, because of the 'runid' line, which has a string in the 3rd column, all columns are read in as characters. So, aftern reading in the file, mark and drop rows with 'all' in column 2. Then convert it to a wide-table using dcast(). Finally, create a matrix from the data frame.

```
# Build the Measure x Query x Score Matrix
MQSMatrix <- function(fEval) {
    vEvalHeader = c("measure", "query", "score")
    dlMQS = read.table(fEval, header = FALSE, col.names = vEvalHeader, na.strings = c("runid", "all"))
    dlMQS = na.omit(dlMQS)
    ## Convert long-format table to wide-format table.
    ## Use col 1 as row names and then drop it.
    ## Create matrix from table
    dwMQS = dcast(dlMQS, measure ~ query, value.var = "score")
    rownames(dwMQS) = dwMQS[, 1]
    dwMQS = dwMQS[, -1]
    # write.table(dwMQS, "tables/DEMO.a.p.bm25.20.D.x", quote = FALSE, row.names = FALSE)
    mMQS = data.matrix(dwMQS)
    return(mMQS)
```

```
}

# Build the Algorithm x Query x Score matrix
AQSMatrix <- function(vfEval) {
    lmEval = lapply(vfEval, function(x) MQSMatrix(x))
    lmAQS = lapply(lmEval, function(z) z["map",])
    vAName = basename(vfEval)
    vQName = names(lmAQS[[1]])
    mAQS = matrix(unlist(lmAQS), nrow = length(vfEval), byrow = T, dimnames = list(vAName, vQName))
    return(mAQS)
}

# Get list of eval files whose names match a regex.
getEvalFileList <- function(regex) {
    vfEval = list.files("data/LTR/evals", pattern = regex, full.names = TRUE)
}
```

DEMO

The following transformation is demonstrated here: TRECEVAL output files -> matrix -> list of matrices; one each for a test-collection

TRECEVAL output file -> Measure x Query x Score matrix (mMQS)

```
fEval = "data/LTR/evals/AP.d.p.bm25.196.T.x"
mMQS = MQSMatrix(fEval)
```

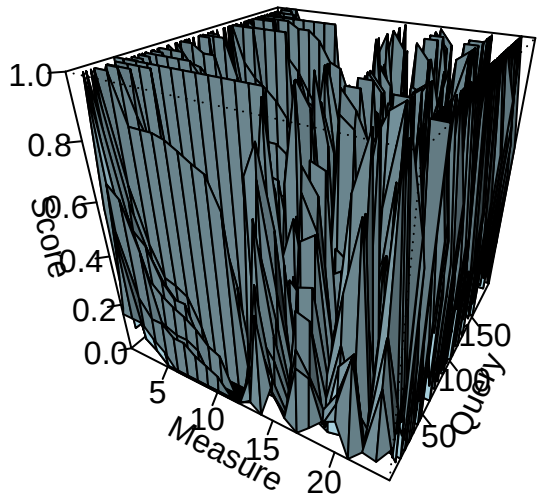
mMQS

```
##                              1         2         3         4         5
## bpref                   0.1588    0.1810    0.1757    0.2439    0.5669
## iprec_at_recall_0.00    0.1585    0.5000    0.5000    0.6667    1.0000
## iprec_at_recall_0.10    0.1585    0.0815    0.2571    0.6250    0.2308
## iprec_at_recall_0.20    0.1057    0.0483    0.1866    0.4500    0.1698
## iprec_at_recall_0.30    0.0663    0.0000    0.1866    0.1489    0.1290
## iprec_at_recall_0.40    0.0000    0.0000    0.1439    0.0455    0.0474
## iprec_at_recall_0.50    0.0000    0.0000    0.1439    0.0429    0.0474
## iprec_at_recall_0.60    0.0000    0.0000    0.1056    0.0429    0.0385
## iprec_at_recall_0.70    0.0000    0.0000    0.0843    0.0420    0.0000
## iprec_at_recall_0.80    0.0000    0.0000    0.0609    0.0000    0.0000
## iprec_at_recall_0.90    0.0000    0.0000    0.0000    0.0000    0.0000
## iprec_at_recall_1.00    0.0000    0.0000    0.0000    0.0000    0.0000
## map                     0.0367    0.0256    0.1286    0.1624    0.0906
## num_rel               111.0000  214.0000   74.0000   41.0000   40.0000
## num_rel_ret            40.0000   48.0000   60.0000   32.0000   27.0000
## num_ret              1000.0000 1000.0000 1000.0000 1000.0000 1000.0000
## P_10                    0.0000    0.4000    0.3000    0.5000    0.2000
## P_100                   0.1400    0.1200    0.1700    0.1400    0.1200
## P_1000                  0.0400    0.0480    0.0600    0.0320    0.0270
## P_15                    0.0000    0.2667    0.3333    0.4667    0.1333
## P_20                    0.0500    0.2500    0.2500    0.4500    0.2000
## P_200                   0.1050    0.0900    0.1450    0.0700    0.0600
## P_30                    0.1333    0.2000    0.2333    0.3333    0.2000
## P_5                     0.0000    0.4000    0.4000    0.4000    0.2000
```

```
## P_500                    0.0600    0.0680    0.0920    0.0400    0.0440
## recip_rank               0.0500    0.3333    0.5000    0.5000    1.0000
## Rprec                    0.1351    0.0935    0.1622    0.2683    0.1750
```

Just for fun: 3D plot for mMQS matrix. Rows 14, 15 and 16(num _rel, num _rel_ret, num _ret) were dropped to keep scores within a range that creates a decent picture.

```
mMQS = mMQS[ - c(14, 15, 16),]
 persp(x = 1:nrow(mMQS), y = 1:ncol(mMQS), z = mMQS, xlab = "Measure", ylab = "Query", zlab = "Score",
```



Set of TRECEVAL output files -> list of mMQS matrices -> Algorithm x Query x Measure matrix (mAQS)

```
vfEval = getEvalFileList("^AP\\..*")
mAQS = AQSMatrix(vfEval)
```

mAQS

```
##                                1      2      3      4      5
## AP.d.p.bm25.196.T.x          0.0367 0.0256 0.1286 0.1624 0.0906
## AP.d.p.bm25e.196.T.x         0.0317 0.0256 0.1289 0.1626 0.0934
## AP.d.p.bm25L.196.T.x         0.0442 0.0144 0.1176 0.1925 0.0937
## AP.d.p.defaultL.196.T.x      0.0358 0.0094 0.1119 0.1356 0.1007
## AP.d.p.dfrL.196.T.x          0.0393 0.0146 0.1159 0.2073 0.0810
## AP.d.p.lmdirichletL.196.T.x 0.0474 0.0221 0.1283 0.1590 0.0697
## AP.d.x.bm25.196.T.x          0.0282 0.0269 0.0385 0.0726 0.1224
```

```
## AP.d.x.bm25e.196.T.x        0.0230 0.0269 0.0384 0.0726 0.1224
## AP.d.x.bm25L.196.T.x        0.0357 0.0123 0.0351 0.1127 0.1449
## AP.d.x.defaultL.196.T.x     0.0252 0.0109 0.0291 0.0561 0.1458
## AP.d.x.dfrL.196.T.x         0.0275 0.0124 0.0376 0.1044 0.1393
## AP.d.x.lmdirichletL.196.T.x 0.0390 0.0213 0.0315 0.0758 0.1387
```

Construct a list of mAQS matrices (lmAQS); one each for a test-collections.

```
vTName = c("AP", "DOE", "FR")
lTIndex = setNames(as.list(1:length(vTName)), vTName)
vfEvalRgx = paste("^", vTName, "\\.", sep = "")
lmAQS = lapply(vfEvalRgx, function(x) {y = getEvalFileList(x); AQSMatrix(y)})
```

Part of one matrix from lmAQS for the AP test-collection:

```
##                                1      2      3      4      5
## AP.d.p.bm25.196.T.x         0.0367 0.0256 0.1286 0.1624 0.0906
## AP.d.p.bm25e.196.T.x        0.0317 0.0256 0.1289 0.1626 0.0934
## AP.d.p.bm25L.196.T.x        0.0442 0.0144 0.1176 0.1925 0.0937
## AP.d.p.defaultL.196.T.x     0.0358 0.0094 0.1119 0.1356 0.1007
## AP.d.p.dfrL.196.T.x         0.0393 0.0146 0.1159 0.2073 0.0810
## AP.d.p.lmdirichletL.196.T.x 0.0474 0.0221 0.1283 0.1590 0.0697
## AP.d.x.bm25.196.T.x         0.0282 0.0269 0.0385 0.0726 0.1224
## AP.d.x.bm25e.196.T.x        0.0230 0.0269 0.0384 0.0726 0.1224
## AP.d.x.bm25L.196.T.x        0.0357 0.0123 0.0351 0.1127 0.1449
## AP.d.x.defaultL.196.T.x     0.0252 0.0109 0.0291 0.0561 0.1458
## AP.d.x.dfrL.196.T.x         0.0275 0.0124 0.0376 0.1044 0.1393
## AP.d.x.lmdirichletL.196.T.x 0.0390 0.0213 0.0315 0.0758 0.1387
```

Another for the DOE test-collection:

```
##                                3      7     8     11     12
## DOE.d.p.bm25.80.T.x         1.0000 0.0119 0e+00 0.0098 0.0609
## DOE.d.p.bm25e.80.T.x        1.0000 0.0119 0e+00 0.0055 0.0656
## DOE.d.p.bm25L.80.T.x        0.3213 0.0286 0e+00 0.0156 0.0415
## DOE.d.p.defaultL.80.T.x     0.1412 0.0034 0e+00 0.0085 0.0330
## DOE.d.p.dfrL.80.T.x         0.3213 0.0286 0e+00 0.0424 0.0368
## DOE.d.p.lmdirichletL.80.T.x 1.0000 0.0061 0e+00 0.0555 0.0340
## DOE.d.x.bm25.80.T.x         0.0339 0.0172 6e-04 0.0133 0.0129
## DOE.d.x.bm25e.80.T.x        0.0291 0.0172 6e-04 0.0079 0.0138
## DOE.d.x.bm25L.80.T.x        0.0085 0.0435 0e+00 0.0211 0.0108
## DOE.d.x.defaultL.80.T.x     0.0067 0.0625 0e+00 0.0114 0.0106
## DOE.d.x.dfrL.80.T.x         0.0085 0.0435 0e+00 0.0563 0.0093
## DOE.d.x.lmdirichletL.80.T.x 0.0207 0.0071 5e-04 0.0665 0.0048
```

Derive the mean-score (MAP) table from all test-collection matrices

```
lvRowMean = lapply(vTName, function(x) rowMeans(lmAQS[[lTIndex[[x]]]]))
## Chris's table: Algorithm x Testcol x Mean Score
vColName = rownames(lmAQS[[lTIndex[[1]]]])
vAName = sapply(strsplit(vColName, "[.]"), function(x) paste(x[2:4], collapse = "."))
mATS = matrix(unlist(lvRowMean), nrow = length(vTName), byrow = T, dimnames = list(vTName, vAName))
```

mATS

```
##      d.p.bm25 d.p.bm25e d.p.bm25L d.p.defaultL  d.p.dfrL
## AP  0.2195990 0.2094714 0.2247745     0.216199 0.2215148
## DOE 0.1728263 0.1818350 0.1477962     0.146970 0.1449250
## FR  0.2013282 0.1998427 0.1910736     0.182650 0.1436773
```

Plot mATS

```
dATS = data.frame(mATS)
dATS[, "Algorithm"] = rownames(dATS)
dlATS = melt(dATS)
```

```
## Using Algorithm as id variables
```

```
ggplot(dlATS, aes(variable, value, fill = variable)) + geom_bar(width = 0.4, stat = "identity") + facet_
```