# -----------------------------------------------

# Information Retrieval Evaluation Exercise

# Computes:

# 1. Precision@10 (per topic, per system)

# 2. Average Precision@10 per system

# 3. Correlation coefficient between system rankings

# 4. Significance testing between systems

# -----------------------------------------------

# Required libraries

library(dplyr)

library(tidyr)

**# ---------- CONFIGURATION ----------**

qrels\_file <- "qrels.trec8.adhoc" # Path to qrels file

runs\_folder <- "system\_runs" # Folder containing ~15 system run files

top\_k <- 10

**# ---------- STEP 1: LOAD QRELS ----------**

~~setwd("C:/Users/User/Documents")~~

load\_qrels <- function(qrels.trec8.adhoc) {

qrels <- read.table(qrels.trec8.adhoc, header = FALSE, stringsAsFactors = FALSE)

colnames(qrels) <- c("topic", "iter", "docid", "relevance")

return(qrels)

}

qrels <- load\_qrels(qrels\_file)

**# ---------- STEP 2: LOAD SYSTEM RUN ----------**

load\_run <- function(set20) {

run <- read.table(set20, header = FALSE, stringsAsFactors = FALSE)

colnames(run) <- c("topic", "Q0", "docid", "rank", "score", "system")

return(run)

}

**# ---------- STEP 3: COMPUTE PRECISION@10 ----------**

precision\_at\_k <- function(run, qrels, k = 10) {

topics <- unique(run$topic)

results <- data.frame(topic = integer(), Pk = numeric())

for (t in topics) {

docs <- run %>% filter(topic == t) %>% arrange(rank) %>% head(k) %>%

pull(docid)

relevant\_docs <- qrels %>% filter(topic == t & relevance > 0) %>% pull(docid)

precision <- length(intersect(docs, relevant\_docs)) / k

results <- rbind(results, data.frame(topic = t, Pk = precision))

}

return(results)

}

**# ---------- STEP 4: LOOP THROUGH SYSTEMS ----------**

setwd("C:/Users/User/Documents/set20")

files <- list.files("C:/Users/User/Documents/set20")

all\_scores <- data.frame()

for (f in files) {

system\_name <- tools::file\_path\_sans\_ext(basename(f))

run <- load\_run(f)

p10\_df <- precision\_at\_k(run, qrels, top\_k)

p10\_df$system <- system\_name

all\_scores <- rbind(all\_scores, p10\_df)

}

**# ---------- STEP 5: COMPUTE AVERAGE P@10 ----------**

avg\_scores <- all\_scores %>%

group\_by(system) %>%

summarise(avg\_P10 = mean(Pk))

cat("=== Average Precision@10 per System ===\n")

print(avg\_scores)

**Expected Results:**

**system avg\_P10**

**<chr> <dbl>**

**1 input.acsys8alo 0.53**

**2 input.apl8n 0.468**

**3 input.CL99SDopt1 0.708**

**4 input.fub99tf 0.526**

**5 input.ibmg99c 0.304**

**6 input.INQ601 0.436**

**7 input.kdd8ps16 0.28**

**8 input.mds08a5 0.39**

**9 input.nttd8ale 0.494**

**10 input.ric8tpx 0.456**

**11 input.surffal2 0.012**

**12 input.unc8al42 0.316**

**# ---------- STEP 6: CORRELATION COEFFICIENT ----------**

cat("\n=== Correlation Coefficients Between Systems ===\n")

systems <- unique(all\_scores$system)

for (i in 1:(length(systems)-1)) {

for (j in (i+1):length(systems)) {

sys1 <- systems[i]

sys2 <- systems[j]

s1 <- all\_scores %>% filter(system == sys1) %>% arrange(topic) %>% pull(Pk)

s2 <- all\_scores %>% filter(system == sys2) %>% arrange(topic) %>% pull(Pk)

corr <- cor(s1, s2, method = "pearson")

cat(paste(sys1, "vs", sys2, ": r =", round(corr, 3), "\n"))

}

}

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**# ---------- STEP 7: SIGNIFICANCE TESTING ----------**

cat("\n=== Significance Testing (Paired t-test) ===\n")

significant\_pairs <- 0

for (i in 1:(length(systems)-1)) {

for (j in (i+1):length(systems)) {

sys1 <- systems[i]

sys2 <- systems[j]

s1 <- all\_scores %>% filter(system == sys1) %>% arrange(topic) %>% pull(Pk)

s2 <- all\_scores %>% filter(system == sys2) %>% arrange(topic) %>% pull(Pk)

test <- t.test(s1, s2, paired = TRUE)

p\_val <- test$p.value

if (p\_val < 0.05) {

significant\_pairs <- significant\_pairs + 1

}

cat(paste(sys1, "vs", sys2, ": p =", round(p\_val, 4), "\n"))

}

}

cat(paste("\nTotal significantly different pairs (p < 0.05):", significant\_pairs, "\n"))

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.