# Triplet Judgments using Mechanical Turk

Praveen Chandar, University of Delaware

## 1. INTRODUCTION

The various steps involved are as follows:

- Document Pooling
- Generate and Sample triplets
- Create HITs
- Set Qualifications
- Submit HITs

## 2. DOCUMENT POOLING

The *evalIR* package has a function to pool documents at a specified depth given a list of runs in TREC format.

```r
library(evalIR, quietly=T)
library(plyr)
trec09 <- list.files(path='../demo/data/diversity/trec2009', full.names=T)
trec09_runids <- basename(trec09)
pooling_depth = 5
runs <- read.runs(runPaths= trec09,
                  runids= trec09_runids,
                  limit= pooling_depth)
topK_pooling <- function(x, pooling_depth=5){
  pooled_docs <- pooling.topk(runs$getRankMatrix(x$query), pooling_depth)
  return(data.frame(docID=names(pooled_docs)))
}
pooled_docs <- adply(data.frame(query=runs$getQueries()), 1, topK_pooling)
head(pooled_docs, n=5)
```

```
##   query                     docID
## 1     1 clueweb09-en0001-02-21652
## 2     1 clueweb09-en0010-57-32918
## 3     1 clueweb09-en0010-79-02218
## 4     1 clueweb09-en0010-93-11767
## 5     1 clueweb09-en0025-89-06994
```

## 3. GENERATE AND SAMPLE TRIPLETS

```r
generate_triplets <- function(x, pooled_docs){
  documents <- subset(pooled_docs, query == x)
  documents$docID <- as.character(documents$docID)
  doc_ids <- documents$docID
  triplets <- ddply(documents,
                    .(docID),
                    function(x)
```

```
                      t(combn(doc_ids[!doc_ids %in% c(x$docID)],2)))
  colnames(triplets) <- c('top_doc','left_doc','right_doc')
  return(triplets)
}
triplets <- ddply(data.frame(query=runs$getQueries()),
                  .(query),
                  function(x) generate_triplets((x$query), pooled_docs))
```

### 4. CREATE HITS

*MTurkR* can be used to automatically submit jobs from R to Amazon Mechanical Turk.