package com.twitter.cr\_mixer.similarity\_engine

import com.twitter.cr\_mixer.model.TweetWithCandidateGenerationInfo

import com.twitter.simclusters\_v2.common.TweetId

import scala.collection.mutable

import scala.collection.mutable.ArrayBuffer

object SimilaritySourceOrderingUtil {

/\*\*

\* This function flatten and dedup input candidates according to the order in the input Seq

\* [[candidate10, candidate11], [candidate20, candidate21]] => [candidate10, candidate11, candidate20, candidate21]

\*/

def keepGivenOrder(

candidates: Seq[Seq[TweetWithCandidateGenerationInfo]],

): Seq[TweetWithCandidateGenerationInfo] = {

val seen = mutable.Set[TweetId]()

val combinedCandidates = candidates.flatten

val result = ArrayBuffer[TweetWithCandidateGenerationInfo]()

combinedCandidates.foreach { candidate =>

val candidateTweetId = candidate.tweetId

val seenCandidate = seen.contains(candidateTweetId) // de-dup

if (!seenCandidate) {

result += candidate

seen.add(candidate.tweetId)

}

}

//convert result to immutable seq

result.toList

}

}