package com.twitter.simclusters\_v2.scio

package bq\_generation.common

import com.twitter.algebird\_internal.thriftscala.DecayedValue

import com.twitter.simclusters\_v2.thriftscala.FullClusterId

import com.twitter.simclusters\_v2.thriftscala.ModelVersion

import com.twitter.simclusters\_v2.thriftscala.Scores

import com.twitter.simclusters\_v2.thriftscala.TopKTweetsWithScores

import com.twitter.snowflake.id.SnowflakeId

import org.apache.avro.generic.GenericRecord

import org.apache.beam.sdk.io.gcp.bigquery.SchemaAndRecord

import org.apache.beam.sdk.transforms.SerializableFunction

import scala.collection.JavaConverters.\_

object IndexGenerationUtil {

// Function that parses [GenericRecord] results we read from BQ into [TopKTweetsForClusterKey]

def parseClusterTopKTweetsFn(tweetEmbeddingsHalfLife: Int) =

new SerializableFunction[SchemaAndRecord, TopKTweetsForClusterKey] {

override def apply(record: SchemaAndRecord): TopKTweetsForClusterKey = {

val genericRecord: GenericRecord = record.getRecord()

TopKTweetsForClusterKey(

clusterId = FullClusterId(

modelVersion = ModelVersion.Model20m145k2020,

clusterId = genericRecord.get("clusterId").toString.toInt

),

topKTweetsWithScores = parseTopKTweetsForClusterKeyColumn(

genericRecord,

"topKTweetsForClusterKey",

tweetEmbeddingsHalfLife),

)

}

}

// Function that parses the topKTweetsForClusterKey column into [TopKTweetsWithScores]

def parseTopKTweetsForClusterKeyColumn(

genericRecord: GenericRecord,

columnName: String,

tweetEmbeddingsHalfLife: Int

): TopKTweetsWithScores = {

val tweetScorePairs: java.util.List[GenericRecord] =

genericRecord.get(columnName).asInstanceOf[java.util.List[GenericRecord]]

val tweetIdToScoresMap = tweetScorePairs.asScala

.map((gr: GenericRecord) => {

// Retrieve the tweetId and tweetScore

val tweetId = gr.get("tweetId").toString.toLong

val tweetScore = gr.get("tweetScore").toString.toDouble

// Transform tweetScore into DecayedValue

// Ref: https://github.com/twitter/algebird/blob/develop/algebird-core/src/main/scala/com/twitter/algebird/DecayedValue.scala

val scaledTime =

SnowflakeId.unixTimeMillisFromId(tweetId) \* math.log(2.0) / tweetEmbeddingsHalfLife

val decayedValue = DecayedValue(tweetScore, scaledTime)

// Update the TopTweets Map

tweetId -> Scores(favClusterNormalized8HrHalfLifeScore = Some(decayedValue))

}).toMap

TopKTweetsWithScores(topTweetsByFavClusterNormalizedScore = Some(tweetIdToScoresMap))

}

case class TopKTweetsForClusterKey(

clusterId: FullClusterId,

topKTweetsWithScores: TopKTweetsWithScores)

}