package com.twitter.simclusters\_v2.scio.bq\_generation

package tweets\_ann

import com.google.api.services.bigquery.model.TimePartitioning

import com.spotify.scio.ScioContext

import com.spotify.scio.coders.Coder

import com.twitter.beam.io.dal.DAL

import com.twitter.beam.io.fs.multiformat.PathLayout

import com.twitter.beam.job.DateRangeOptions

import com.twitter.conversions.DurationOps.richDurationFromInt

import com.twitter.dal.client.dataset.KeyValDALDataset

import com.twitter.scalding\_internal.multiformat.format.keyval.KeyVal

import com.twitter.scio\_internal.coders.ThriftStructLazyBinaryScroogeCoder

import com.twitter.scio\_internal.job.ScioBeamJob

import com.twitter.scrooge.ThriftStruct

import com.twitter.simclusters\_v2.scio.bq\_generation.common.BQGenerationUtil.getMTSConsumerEmbeddingsFav90P20MSQL

import com.twitter.simclusters\_v2.scio.bq\_generation.common.BQGenerationUtil.getInterestedIn2020SQL

import com.twitter.simclusters\_v2.scio.bq\_generation.tweets\_ann.TweetsANNFromBQ.getTweetRecommendationsBQ

import com.twitter.simclusters\_v2.hdfs\_sources.OfflineTweetRecommendationsFromInterestedIn20M145K2020ScalaDataset

import com.twitter.simclusters\_v2.hdfs\_sources.OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl0El15ScalaDataset

import com.twitter.simclusters\_v2.hdfs\_sources.OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl2El15ScalaDataset

import com.twitter.simclusters\_v2.hdfs\_sources.OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl2El50ScalaDataset

import com.twitter.simclusters\_v2.hdfs\_sources.OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl8El50ScalaDataset

import com.twitter.simclusters\_v2.hdfs\_sources.OfflineTweetRecommendationsFromMtsConsumerEmbeddingsScalaDataset

import com.twitter.simclusters\_v2.scio.bq\_generation.common.BQTableDetails

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

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

import com.twitter.tcdc.bqblaster.beam.syntax.BigQueryIOHelpers

import com.twitter.tcdc.bqblaster.beam.BQBlasterIO.AvroConverter

import com.twitter.tcdc.bqblaster.core.avro.TypedProjection

import com.twitter.tcdc.bqblaster.core.transform.RootTransform

import java.time.Instant

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

import org.joda.time.DateTime

trait TweetsANNJob extends ScioBeamJob[DateRangeOptions] {

// Configs to set for different type of embeddings and jobs

val isAdhoc: Boolean

val getConsumerEmbeddingsSQLFunc: (DateTime, Int) => String

val outputTable: BQTableDetails

val keyValDatasetOutputPath: String

val tweetRecommentationsSnapshotDataset: KeyValDALDataset[KeyVal[Long, CandidateTweetsList]]

val tweetEmbeddingsGenerationHalfLife: Int = Config.SimClustersTweetEmbeddingsGenerationHalfLife

val tweetEmbeddingsGenerationEmbeddingLength: Int =

Config.SimClustersTweetEmbeddingsGenerationEmbeddingLength

// Base configs

val projectId = "twttr-recos-ml-prod"

val environment: DAL.Env = if (isAdhoc) DAL.Environment.Dev else DAL.Environment.Prod

override implicit def scroogeCoder[T <: ThriftStruct: Manifest]: Coder[T] =

ThriftStructLazyBinaryScroogeCoder.scroogeCoder

override def configurePipeline(sc: ScioContext, opts: DateRangeOptions): Unit = {

// The time when the job is scheduled

val queryTimestamp = opts.interval.getEnd

// Read consumer embeddings SQL

val consumerEmbeddingsSQL = getConsumerEmbeddingsSQLFunc(queryTimestamp, 14)

// Generate tweet embeddings and tweet ANN results

val tweetRecommendations =

getTweetRecommendationsBQ(

sc,

queryTimestamp,

consumerEmbeddingsSQL,

tweetEmbeddingsGenerationHalfLife,

tweetEmbeddingsGenerationEmbeddingLength

)

// Setup BQ writer

val ingestionTime = opts.getDate().value.getEnd.toDate

val bqFieldsTransform = RootTransform

.Builder()

.withPrependedFields("ingestionTime" -> TypedProjection.fromConstant(ingestionTime))

val timePartitioning = new TimePartitioning()

.setType("HOUR").setField("ingestionTime").setExpirationMs(3.days.inMilliseconds)

val bqWriter = BigQueryIO

.write[CandidateTweets]

.to(outputTable.toString)

.withExtendedErrorInfo()

.withTimePartitioning(timePartitioning)

.withLoadJobProjectId(projectId)

.withThriftSupport(bqFieldsTransform.build(), AvroConverter.Legacy)

.withCreateDisposition(BigQueryIO.Write.CreateDisposition.CREATE\_IF\_NEEDED)

.withWriteDisposition(BigQueryIO.Write.WriteDisposition.WRITE\_APPEND)

// Save Tweet ANN results to BQ

tweetRecommendations

.map { userToTweetRecommendations =>

{

CandidateTweets(

targetUserId = userToTweetRecommendations.userId,

recommendedTweets = userToTweetRecommendations.tweetCandidates)

}

}

.saveAsCustomOutput(s"WriteToBQTable - ${outputTable}", bqWriter)

// Save Tweet ANN results as KeyValSnapshotDataset

tweetRecommendations

.map { userToTweetRecommendations =>

KeyVal(

userToTweetRecommendations.userId,

CandidateTweetsList(userToTweetRecommendations.tweetCandidates))

}.saveAsCustomOutput(

name = "WriteTweetRecommendationsToKeyValDataset",

DAL.writeVersionedKeyVal(

tweetRecommentationsSnapshotDataset,

PathLayout.VersionedPath(prefix =

((if (!isAdhoc)

Config.RootMHPath

else

Config.AdhocRootPath)

+ keyValDatasetOutputPath)),

instant = Instant.ofEpochMilli(opts.interval.getEndMillis - 1L),

environmentOverride = environment,

)

)

}

}

/\*\*

\* Scio job for adhoc run for tweet recommendations from IIKF 2020

\*/

object IIKF2020TweetsANNBQAdhocJob extends TweetsANNJob {

override val isAdhoc = true

override val getConsumerEmbeddingsSQLFunc = getInterestedIn2020SQL

override val outputTable = BQTableDetails(

"twttr-recos-ml-prod",

"multi\_type\_simclusters",

"offline\_tweet\_recommendations\_from\_interested\_in\_20M\_145K\_2020\_adhoc")

override val keyValDatasetOutputPath = Config.IIKFANNOutputPath

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromInterestedIn20M145K2020ScalaDataset

}

/\*\*

\* Scio job for adhoc run for tweet recommendations from IIKF 2020 with

\* - Half life = 8hrs

\* - Embedding Length = 50

\*/

object IIKF2020Hl8El50TweetsANNBQAdhocJob extends TweetsANNJob {

override val isAdhoc = true

override val getConsumerEmbeddingsSQLFunc = getInterestedIn2020SQL

override val outputTable = BQTableDetails(

"twttr-recos-ml-prod",

"multi\_type\_simclusters",

"offline\_tweet\_recommendations\_from\_interested\_in\_20M\_145K\_2020\_HL\_8\_EL\_50\_adhoc")

override val keyValDatasetOutputPath = Config.IIKFHL8EL50ANNOutputPath

override val tweetEmbeddingsGenerationEmbeddingLength: Int = 50

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] = {

OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl8El50ScalaDataset

}

}

/\*\*

\* Scio job for adhoc run for tweet recommendations from MTS Consumer Embeddings

\*/

object MTSConsumerEmbeddingsTweetsANNBQAdhocJob extends TweetsANNJob {

override val isAdhoc = true

override val getConsumerEmbeddingsSQLFunc = getMTSConsumerEmbeddingsFav90P20MSQL

override val outputTable = BQTableDetails(

"twttr-recos-ml-prod",

"multi\_type\_simclusters",

"offline\_tweet\_recommendations\_from\_mts\_consumer\_embeddings\_adhoc")

override val keyValDatasetOutputPath = Config.MTSConsumerEmbeddingsANNOutputPath

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromMtsConsumerEmbeddingsScalaDataset

}

/\*\*

Scio job for batch run for tweet recommendations from IIKF 2020

The schedule cmd needs to be run only if there is any change in the config

\*/

object IIKF2020TweetsANNBQBatchJob extends TweetsANNJob {

override val isAdhoc = false

override val getConsumerEmbeddingsSQLFunc = getInterestedIn2020SQL

override val outputTable = BQTableDetails(

"twttr-bq-cassowary-prod",

"user",

"offline\_tweet\_recommendations\_from\_interested\_in\_20M\_145K\_2020")

override val keyValDatasetOutputPath = Config.IIKFANNOutputPath

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromInterestedIn20M145K2020ScalaDataset

}

/\*\*

Scio job for batch run for tweet recommendations from IIKF 2020 with parameter setup:

- Half Life: None, no decay, direct sum

- Embedding Length: 15

The schedule cmd needs to be run only if there is any change in the config

\*/

object IIKF2020Hl0El15TweetsANNBQBatchJob extends TweetsANNJob {

override val isAdhoc = false

override val getConsumerEmbeddingsSQLFunc = getInterestedIn2020SQL

override val outputTable = BQTableDetails(

"twttr-bq-cassowary-prod",

"user",

"offline\_tweet\_recommendations\_from\_interested\_in\_20M\_145K\_2020\_HL\_0\_EL\_15")

override val keyValDatasetOutputPath = Config.IIKFHL0EL15ANNOutputPath

override val tweetEmbeddingsGenerationHalfLife: Int = -1

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl0El15ScalaDataset

}

/\*\*

Scio job for batch run for tweet recommendations from IIKF 2020 with parameter setup:

- Half Life: 2hrs

- Embedding Length: 15

The schedule cmd needs to be run only if there is any change in the config

\*/

object IIKF2020Hl2El15TweetsANNBQBatchJob extends TweetsANNJob {

override val isAdhoc = false

override val getConsumerEmbeddingsSQLFunc = getInterestedIn2020SQL

override val outputTable = BQTableDetails(

"twttr-bq-cassowary-prod",

"user",

"offline\_tweet\_recommendations\_from\_interested\_in\_20M\_145K\_2020\_HL\_2\_EL\_15")

override val keyValDatasetOutputPath = Config.IIKFHL2EL15ANNOutputPath

override val tweetEmbeddingsGenerationHalfLife: Int = 7200000 // 2hrs in ms

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl2El15ScalaDataset

}

/\*\*

Scio job for batch run for tweet recommendations from IIKF 2020 with parameter setup:

- Half Life: 2hrs

- Embedding Length: 50

The schedule cmd needs to be run only if there is any change in the config

\*/

object IIKF2020Hl2El50TweetsANNBQBatchJob extends TweetsANNJob {

override val isAdhoc = false

override val getConsumerEmbeddingsSQLFunc = getInterestedIn2020SQL

override val outputTable = BQTableDetails(

"twttr-bq-cassowary-prod",

"user",

"offline\_tweet\_recommendations\_from\_interested\_in\_20M\_145K\_2020\_HL\_2\_EL\_50")

override val keyValDatasetOutputPath = Config.IIKFHL2EL50ANNOutputPath

override val tweetEmbeddingsGenerationHalfLife: Int = 7200000 // 2hrs in ms

override val tweetEmbeddingsGenerationEmbeddingLength: Int = 50

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl2El50ScalaDataset

}

/\*\*

Scio job for batch run for tweet recommendations from IIKF 2020 with parameter setup:

- Half Life: 8hrs

- Embedding Length: 50

The schedule cmd needs to be run only if there is any change in the config

\*/

object IIKF2020Hl8El50TweetsANNBQBatchJob extends TweetsANNJob {

override val isAdhoc = false

override val getConsumerEmbeddingsSQLFunc = getInterestedIn2020SQL

override val outputTable = BQTableDetails(

"twttr-bq-cassowary-prod",

"user",

"offline\_tweet\_recommendations\_from\_interested\_in\_20M\_145K\_2020\_HL\_8\_EL\_50")

override val keyValDatasetOutputPath = Config.IIKFHL8EL50ANNOutputPath

override val tweetEmbeddingsGenerationEmbeddingLength: Int = 50

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromInterestedIn20M145K2020Hl8El50ScalaDataset

}

/\*\*

Scio job for batch run for tweet recommendations from MTS Consumer Embeddings

The schedule cmd needs to be run only if there is any change in the config

\*/

object MTSConsumerEmbeddingsTweetsANNBQBatchJob extends TweetsANNJob {

override val isAdhoc = false

override val getConsumerEmbeddingsSQLFunc = getMTSConsumerEmbeddingsFav90P20MSQL

override val outputTable = BQTableDetails(

"twttr-bq-cassowary-prod",

"user",

"offline\_tweet\_recommendations\_from\_mts\_consumer\_embeddings")

override val keyValDatasetOutputPath = Config.MTSConsumerEmbeddingsANNOutputPath

override val tweetRecommentationsSnapshotDataset: KeyValDALDataset[

KeyVal[Long, CandidateTweetsList]

] =

OfflineTweetRecommendationsFromMtsConsumerEmbeddingsScalaDataset

}