package com.twitter.simclusters\_v2.scalding.embedding.producer

import com.twitter.scalding.\_

import com.twitter.scalding\_internal.dalv2.DALWrite.\_

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

import com.twitter.scalding\_internal.source.lzo\_scrooge.FixedPathLzoScrooge

import com.twitter.simclusters\_v2.hdfs\_sources.AggregatableProducerSimclustersEmbeddingsByFollowScore2020ScalaDataset

import com.twitter.simclusters\_v2.hdfs\_sources.AggregatableProducerSimclustersEmbeddingsByFollowScore2020ThriftScalaDataset

import com.twitter.simclusters\_v2.scalding.embedding.common.EmbeddingUtil

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

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

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

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

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

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

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

import com.twitter.wtf.scalding.jobs.common.AdhocExecutionApp

import com.twitter.wtf.scalding.jobs.common.ScheduledExecutionApp

import java.util.TimeZone

/\*\*

\* This file implements a new Producer SimClusters Embeddings.

\* The differences with existing producer embeddings are:

\*

\* 1) the embedding scores are not normalized, so that one can aggregate multiple producer embeddings by adding them.

\* 2) we use follow scores in the user-producer graph and user-simclusters graph.

\*/

/\*\*

\* Production job:

capesospy-v2 update --build\_locally --start\_cron aggregatable\_producer\_embeddings\_by\_follow\_score\_2020 src/scala/com/twitter/simclusters\_v2/capesos\_config/atla\_proc3.yaml

\*/

object AggregatableFollowBasedProducerEmbeddings2020ScheduledApp

extends AggregatableFollowBasedProducerEmbeddingsBaseApp

with ScheduledExecutionApp {

override val modelVersion: ModelVersion = ModelVersion.Model20m145k2020

// Not using the EmbeddingUtil.getHdfsPath to preserve the previous functionality.

private val outputPath: String =

"/user/cassowary/manhattan\_sequence\_files/producer\_simclusters\_aggregatable\_embeddings\_by\_follow\_score\_20m145k2020"

private val outputPathThrift: String = EmbeddingUtil.getHdfsPath(

isAdhoc = false,

isManhattanKeyVal = false,

modelVersion = modelVersion,

pathSuffix = "producer\_simclusters\_aggregatable\_embeddings\_by\_follow\_score\_thrift"

)

override def batchIncrement: Duration = Days(7)

override def firstTime: RichDate = RichDate("2021-11-10")

override def writeToManhattan(

output: TypedPipe[KeyVal[SimClustersEmbeddingId, SimClustersEmbedding]]

)(

implicit dateRange: DateRange,

timeZone: TimeZone,

uniqueID: UniqueID

): Execution[Unit] = {

output

.writeDALVersionedKeyValExecution(

AggregatableProducerSimclustersEmbeddingsByFollowScore2020ScalaDataset,

D.Suffix(outputPath),

version = ExplicitEndTime(dateRange.end)

)

}

override def writeToThrift(

output: TypedPipe[SimClustersEmbeddingWithId]

)(

implicit dateRange: DateRange,

timeZone: TimeZone,

uniqueID: UniqueID

): Execution[Unit] = {

output

.writeDALSnapshotExecution(

dataset = AggregatableProducerSimclustersEmbeddingsByFollowScore2020ThriftScalaDataset,

updateStep = D.Daily,

pathLayout = D.Suffix(outputPathThrift),

fmt = D.Parquet,

endDate = dateRange.end

)

}

}

/\*\*

./bazel bundle src/scala/com/twitter/simclusters\_v2/scalding/embedding/producer:aggregatable\_follow\_based\_producer\_embeddings\_job\_2020-adhoc

scalding remote run \

--user cassowary \

--keytab /var/lib/tss/keys/fluffy/keytabs/client/cassowary.keytab \

--principal service\_acoount@TWITTER.BIZ \

--cluster bluebird-qus1 \

--main-class com.twitter.simclusters\_v2.scalding.embedding.producer.AggregatableFollowBasedProducerEmbeddings2020AdhocApp \

--target src/scala/com/twitter/simclusters\_v2/scalding/embedding/producer:aggregatable\_follow\_based\_producer\_embeddings\_job\_2020-adhoc \

--hadoop-properties "scalding.with.reducers.set.explicitly=true mapreduce.job.reduces=4000" \

-- --date 2021-11-10

\*/

object AggregatableFollowBasedProducerEmbeddings2020AdhocApp

extends AggregatableFollowBasedProducerEmbeddingsBaseApp

with AdhocExecutionApp {

override val modelVersion: ModelVersion = ModelVersion.Model20m145k2020

private val outputPath: String = EmbeddingUtil.getHdfsPath(

isAdhoc = true,

isManhattanKeyVal = true,

modelVersion = modelVersion,

pathSuffix = "producer\_simclusters\_aggregatable\_embeddings\_by\_follow\_score"

)

private val outputPathThrift: String = EmbeddingUtil.getHdfsPath(

isAdhoc = true,

isManhattanKeyVal = false,

modelVersion = modelVersion,

pathSuffix = "producer\_simclusters\_aggregatable\_embeddings\_by\_follow\_score\_thrift"

)

override def writeToManhattan(

output: TypedPipe[KeyVal[SimClustersEmbeddingId, SimClustersEmbedding]]

)(

implicit dateRange: DateRange,

timeZone: TimeZone,

uniqueID: UniqueID

): Execution[Unit] = {

output

.flatMap { keyVal =>

keyVal.value.embedding.map { simClusterWithScore =>

(

keyVal.key.embeddingType,

keyVal.key.modelVersion,

keyVal.key.internalId,

simClusterWithScore.clusterId,

simClusterWithScore.score

)

}

}

.writeExecution(

// Write to TSV for easier debugging of the adhoc job.

TypedTsv(outputPath)

)

}

override def writeToThrift(

output: TypedPipe[SimClustersEmbeddingWithId]

)(

implicit dateRange: DateRange,

timeZone: TimeZone,

uniqueID: UniqueID

): Execution[Unit] = {

output

.writeExecution(

new FixedPathLzoScrooge(outputPathThrift, SimClustersEmbeddingWithId)

)

}

}

trait AggregatableFollowBasedProducerEmbeddingsBaseApp

extends AggregatableProducerEmbeddingsBaseApp {

override val userToProducerScoringFn: NeighborWithWeights => Double =

\_.followScoreNormalizedByNeighborFollowersL2.getOrElse(0.0)

override val userToClusterScoringFn: UserToInterestedInClusterScores => Double =

\_.followScoreClusterNormalizedOnly.getOrElse(0.0)

override val embeddingType: EmbeddingType = EmbeddingType.AggregatableFollowBasedProducer

}