package com.twitter.recosinjector.edges

import com.twitter.recos.internal.thriftscala.RecosHoseMessage

import com.twitter.recos.recos\_injector.thriftscala.{Features, UserTweetAuthorGraphMessage}

import com.twitter.recos.util.Action.Action

import com.twitter.recosinjector.util.TweetDetails

import scala.collection.Map

trait Edge {

// RecosHoseMessage is the thrift struct that the graphs consume.

def convertToRecosHoseMessage: RecosHoseMessage

// UserTweetAuthorGraphMessage is the thrift struct that user\_tweet\_author\_graph consumes.

def convertToUserTweetAuthorGraphMessage: UserTweetAuthorGraphMessage

}

/\*\*

\* Edge corresponding to UserTweetEntityEdge.

\* It captures user-tweet interactions: Create, Like, Retweet, Reply etc.

\*/

case class UserTweetEntityEdge(

sourceUser: Long,

targetTweet: Long,

action: Action,

cardInfo: Option[Byte],

metadata: Option[Long],

entitiesMap: Option[Map[Byte, Seq[Int]]],

tweetDetails: Option[TweetDetails])

extends Edge {

override def convertToRecosHoseMessage: RecosHoseMessage = {

RecosHoseMessage(

leftId = sourceUser,

rightId = targetTweet,

action = action.id.toByte,

card = cardInfo,

entities = entitiesMap,

edgeMetadata = metadata

)

}

private def getFeatures(tweetDetails: TweetDetails): Features = {

Features(

hasPhoto = Some(tweetDetails.hasPhoto),

hasVideo = Some(tweetDetails.hasVideo),

hasUrl = Some(tweetDetails.hasUrl),

hasHashtag = Some(tweetDetails.hasHashtag)

)

}

override def convertToUserTweetAuthorGraphMessage: UserTweetAuthorGraphMessage = {

UserTweetAuthorGraphMessage(

leftId = sourceUser,

rightId = targetTweet,

action = action.id.toByte,

card = cardInfo,

authorId = tweetDetails.flatMap(\_.authorId),

features = tweetDetails.map(getFeatures)

)

}

}

/\*\*

\* Edge corresponding to UserUserGraph.

\* It captures user-user interactions: Follow, Mention, Mediatag.

\*/

case class UserUserEdge(

sourceUser: Long,

targetUser: Long,

action: Action,

metadata: Option[Long])

extends Edge {

override def convertToRecosHoseMessage: RecosHoseMessage = {

RecosHoseMessage(

leftId = sourceUser,

rightId = targetUser,

action = action.id.toByte,

edgeMetadata = metadata

)

}

override def convertToUserTweetAuthorGraphMessage: UserTweetAuthorGraphMessage = {

throw new RuntimeException(

"convertToUserTweetAuthorGraphMessage not implemented in UserUserEdge.")

}

}