package com.twitter.home\_mixer.product.scored\_tweets.response\_transformer.earlybird

import com.twitter.home\_mixer.model.HomeFeatures.AuthorIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.CandidateSourceIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.DirectedAtUserIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.EarlybirdScoreFeature

import com.twitter.home\_mixer.model.HomeFeatures.EarlybirdSearchResultFeature

import com.twitter.home\_mixer.model.HomeFeatures.ExclusiveConversationAuthorIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.FromInNetworkSourceFeature

import com.twitter.home\_mixer.model.HomeFeatures.HasImageFeature

import com.twitter.home\_mixer.model.HomeFeatures.HasVideoFeature

import com.twitter.home\_mixer.model.HomeFeatures.InReplyToTweetIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.InReplyToUserIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.IsRandomTweetFeature

import com.twitter.home\_mixer.model.HomeFeatures.IsRetweetFeature

import com.twitter.home\_mixer.model.HomeFeatures.MentionScreenNameFeature

import com.twitter.home\_mixer.model.HomeFeatures.MentionUserIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.QuotedTweetIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.QuotedUserIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.SourceTweetIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.SourceUserIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.StreamToKafkaFeature

import com.twitter.home\_mixer.model.HomeFeatures.SuggestTypeFeature

import com.twitter.home\_mixer.model.HomeFeatures.TweetUrlsFeature

import com.twitter.home\_mixer.util.tweetypie.content.TweetMediaFeaturesExtractor

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMapBuilder

import com.twitter.search.earlybird.{thriftscala => eb}

object EarlybirdResponseTransformer {

val features: Set[Feature[\_, \_]] = Set(

AuthorIdFeature,

CandidateSourceIdFeature,

DirectedAtUserIdFeature,

EarlybirdScoreFeature,

EarlybirdSearchResultFeature,

ExclusiveConversationAuthorIdFeature,

FromInNetworkSourceFeature,

HasImageFeature,

HasVideoFeature,

InReplyToTweetIdFeature,

InReplyToUserIdFeature,

IsRandomTweetFeature,

IsRetweetFeature,

MentionScreenNameFeature,

MentionUserIdFeature,

StreamToKafkaFeature,

QuotedTweetIdFeature,

QuotedUserIdFeature,

SourceTweetIdFeature,

SourceUserIdFeature,

SuggestTypeFeature,

TweetUrlsFeature

)

def transform(candidate: eb.ThriftSearchResult): FeatureMap = {

val tweet = candidate.tweetypieTweet

val quotedTweet = tweet.flatMap(\_.quotedTweet)

val mentions = tweet.flatMap(\_.mentions).getOrElse(Seq.empty)

val coreData = tweet.flatMap(\_.coreData)

val share = coreData.flatMap(\_.share)

val reply = coreData.flatMap(\_.reply)

FeatureMapBuilder()

.add(AuthorIdFeature, coreData.map(\_.userId))

.add(DirectedAtUserIdFeature, coreData.flatMap(\_.directedAtUser.map(\_.userId)))

.add(EarlybirdSearchResultFeature, Some(candidate))

.add(EarlybirdScoreFeature, candidate.metadata.flatMap(\_.score))

.add(

ExclusiveConversationAuthorIdFeature,

tweet.flatMap(\_.exclusiveTweetControl.map(\_.conversationAuthorId)))

.add(FromInNetworkSourceFeature, false)

.add(HasImageFeature, tweet.exists(TweetMediaFeaturesExtractor.hasImage))

.add(HasVideoFeature, tweet.exists(TweetMediaFeaturesExtractor.hasVideo))

.add(InReplyToTweetIdFeature, reply.flatMap(\_.inReplyToStatusId))

.add(InReplyToUserIdFeature, reply.map(\_.inReplyToUserId))

.add(IsRandomTweetFeature, candidate.tweetFeatures.exists(\_.isRandomTweet.getOrElse(false)))

.add(IsRetweetFeature, share.isDefined)

.add(MentionScreenNameFeature, mentions.map(\_.screenName))

.add(MentionUserIdFeature, mentions.flatMap(\_.userId))

.add(StreamToKafkaFeature, true)

.add(QuotedTweetIdFeature, quotedTweet.map(\_.tweetId))

.add(QuotedUserIdFeature, quotedTweet.map(\_.userId))

.add(SourceTweetIdFeature, share.map(\_.sourceStatusId))

.add(SourceUserIdFeature, share.map(\_.sourceUserId))

.add(

TweetUrlsFeature,

candidate.metadata.flatMap(\_.tweetUrls.map(\_.map(\_.originalUrl))).getOrElse(Seq.empty))

.build()

}

}