package com.twitter.timelineranker.common

import com.twitter.servo.util.FutureArrow

import com.twitter.timelineranker.core.CandidateEnvelope

import com.twitter.timelineranker.core.HydratedCandidatesAndFeaturesEnvelope

import com.twitter.timelineranker.model.RecapQuery

import com.twitter.util.Future

/\*\*

\* Fetches all data required for feature hydration and generates the HydratedCandidatesAndFeaturesEnvelope

\* @param tweetHydrationAndFilteringPipeline Pipeline which fetches the candidate tweets, hydrates and filters them

\* @param languagesService Fetch user languages, required for feature hydration

\* @param userProfileInfoService Fetch user profile info, required for feature hydration

\*/

class FeatureHydrationDataTransform(

tweetHydrationAndFilteringPipeline: FutureArrow[RecapQuery, CandidateEnvelope],

languagesService: UserLanguagesTransform,

userProfileInfoService: UserProfileInfoTransform)

extends FutureArrow[RecapQuery, HydratedCandidatesAndFeaturesEnvelope] {

override def apply(request: RecapQuery): Future[HydratedCandidatesAndFeaturesEnvelope] = {

Future

.join(

languagesService(request),

userProfileInfoService(request),

tweetHydrationAndFilteringPipeline(request)).map {

case (languages, userProfileInfo, transformedCandidateEnvelope) =>

HydratedCandidatesAndFeaturesEnvelope(

transformedCandidateEnvelope,

languages,

userProfileInfo)

}

}

}