package com.twitter.timelineranker.common

import com.twitter.search.earlybird.thriftscala.ThriftSearchResult

import com.twitter.servo.util.FutureArrow

import com.twitter.timelineranker.core.CandidateEnvelope

import com.twitter.timelines.model.tweet.HydratedTweet

import com.twitter.util.Future

/\*\*

\* trims searchResults to match with hydratedTweets

\* (if we previously filtered out hydrated tweets, this transform filters the search result set

\* down to match the hydrated tweets.)

\*/

object TrimToMatchHydratedTweetsTransform

extends FutureArrow[CandidateEnvelope, CandidateEnvelope] {

override def apply(envelope: CandidateEnvelope): Future[CandidateEnvelope] = {

val filteredSearchResults =

trimSearchResults(envelope.searchResults, envelope.hydratedTweets.outerTweets)

val filteredSourceSearchResults =

trimSearchResults(envelope.sourceSearchResults, envelope.sourceHydratedTweets.outerTweets)

Future.value(

envelope.copy(

searchResults = filteredSearchResults,

sourceSearchResults = filteredSourceSearchResults

)

)

}

private def trimSearchResults(

searchResults: Seq[ThriftSearchResult],

hydratedTweets: Seq[HydratedTweet]

): Seq[ThriftSearchResult] = {

val filteredTweetIds = hydratedTweets.map(\_.tweetId).toSet

searchResults.filter(result => filteredTweetIds.contains(result.id))

}

}