package com.twitter.timelineranker.model

import com.twitter.search.common.features.thriftscala.ThriftTweetFeatures

import com.twitter.timelineranker.{thriftscala => thrift}

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

import com.twitter.tweetypie.thriftscala

object CandidateTweet {

val DefaultFeatures: ThriftTweetFeatures = ThriftTweetFeatures()

def fromThrift(candidate: thrift.CandidateTweet): CandidateTweet = {

val tweet: thriftscala.Tweet = candidate.tweet.getOrElse(

throw new IllegalArgumentException(s"CandidateTweet.tweet must have a value")

)

val features = candidate.features.getOrElse(

throw new IllegalArgumentException(s"CandidateTweet.features must have a value")

)

CandidateTweet(HydratedTweet(tweet), features)

}

}

/\*\*

\* A candidate Tweet and associated information.

\* Model object for CandidateTweet thrift struct.

\*/

case class CandidateTweet(hydratedTweet: HydratedTweet, features: ThriftTweetFeatures) {

def toThrift: thrift.CandidateTweet = {

thrift.CandidateTweet(

tweet = Some(hydratedTweet.tweet),

features = Some(features)

)

}

}