package com.twitter.visibility.builder.tweets

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.tweetypie.thriftscala.EditControl

import com.twitter.tweetypie.thriftscala.Tweet

import com.twitter.visibility.builder.FeatureMapBuilder

import com.twitter.visibility.features.TweetIsEditTweet

import com.twitter.visibility.features.TweetIsInitialTweet

import com.twitter.visibility.features.TweetIsLatestTweet

import com.twitter.visibility.features.TweetIsStaleTweet

class EditTweetFeatures(

statsReceiver: StatsReceiver) {

private[this] val scopedStatsReceiver = statsReceiver.scope("edit\_tweet\_features")

private[this] val tweetIsEditTweet =

scopedStatsReceiver.scope(TweetIsEditTweet.name).counter("requests")

private[this] val tweetIsStaleTweet =

scopedStatsReceiver.scope(TweetIsStaleTweet.name).counter("requests")

private[this] val tweetIsLatestTweet =

scopedStatsReceiver.scope(TweetIsLatestTweet.name).counter("requests")

private[this] val tweetIsInitialTweet =

scopedStatsReceiver.scope(TweetIsInitialTweet.name).counter("requests")

def forTweet(

tweet: Tweet

): FeatureMapBuilder => FeatureMapBuilder = {

\_.withConstantFeature(TweetIsEditTweet, tweetIsEditTweet(tweet))

.withConstantFeature(TweetIsStaleTweet, tweetIsStaleTweet(tweet))

.withConstantFeature(TweetIsLatestTweet, tweetIsLatestTweet(tweet))

.withConstantFeature(TweetIsInitialTweet, tweetIsInitialTweet(tweet))

}

def tweetIsStaleTweet(tweet: Tweet, incrementMetric: Boolean = true): Boolean = {

if (incrementMetric) tweetIsStaleTweet.incr()

tweet.editControl match {

case None => false

case Some(ec) =>

ec match {

case eci: EditControl.Initial => eci.initial.editTweetIds.last != tweet.id

case ece: EditControl.Edit =>

ece.edit.editControlInitial.exists(\_.editTweetIds.last != tweet.id)

case \_ => false

}

}

}

def tweetIsEditTweet(tweet: Tweet, incrementMetric: Boolean = true): Boolean = {

if (incrementMetric) tweetIsEditTweet.incr()

tweet.editControl match {

case None => false

case Some(ec) =>

ec match {

case \_: EditControl.Initial => false

case \_ => true

}

}

}

def tweetIsLatestTweet(tweet: Tweet): Boolean = {

tweetIsLatestTweet.incr()

!tweetIsStaleTweet(tweet = tweet, incrementMetric = false)

}

def tweetIsInitialTweet(tweet: Tweet): Boolean = {

tweetIsInitialTweet.incr()

!tweetIsEditTweet(tweet = tweet, incrementMetric = false)

}

}