package com.twitter.simclustersann.filters

import com.twitter.finagle.Service

import com.twitter.finagle.SimpleFilter

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.scrooge.Request

import com.twitter.scrooge.Response

import com.twitter.simclustersann.thriftscala.SimClustersANNService

import com.twitter.util.Future

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class GetTweetCandidatesResponseStatsFilter @Inject() (

statsReceiver: StatsReceiver)

extends SimpleFilter[Request[SimClustersANNService.GetTweetCandidates.Args], Response[

SimClustersANNService.GetTweetCandidates.SuccessType

]] {

private[this] val stats = statsReceiver.scope("method\_response\_stats").scope("getTweetCandidates")

private[this] val candidateScoreStats = stats.stat("candidate\_score\_x1000")

private[this] val emptyResponseCounter = stats.counter("empty")

private[this] val nonEmptyResponseCounter = stats.counter("non\_empty")

override def apply(

request: Request[SimClustersANNService.GetTweetCandidates.Args],

service: Service[Request[SimClustersANNService.GetTweetCandidates.Args], Response[

SimClustersANNService.GetTweetCandidates.SuccessType

]]

): Future[Response[SimClustersANNService.GetTweetCandidates.SuccessType]] = {

val response = service(request)

response.onSuccess { successResponse =>

if (successResponse.value.size == 0)

emptyResponseCounter.incr()

else

nonEmptyResponseCounter.incr()

successResponse.value.foreach { candidate =>

candidateScoreStats.add(candidate.score.toFloat \* 1000)

}

}

response

}

}