package com.twitter.visibility.builder.tweets

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.search.common.constants.thriftscala.ThriftQuerySource

import com.twitter.visibility.builder.FeatureMapBuilder

import com.twitter.visibility.features.SearchCandidateCount

import com.twitter.visibility.features.SearchQueryHasUser

import com.twitter.visibility.features.SearchQuerySource

import com.twitter.visibility.features.SearchResultsPageNumber

import com.twitter.visibility.interfaces.common.search.SearchVFRequestContext

class SearchContextFeatures(

statsReceiver: StatsReceiver) {

private[this] val scopedStatsReceiver = statsReceiver.scope("search\_context\_features")

private[this] val requests = scopedStatsReceiver.counter("requests")

private[this] val searchResultsPageNumber =

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

private[this] val searchCandidateCount =

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

private[this] val searchQuerySource =

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

private[this] val searchQueryHasUser =

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

def forSearchContext(

searchContext: SearchVFRequestContext

): FeatureMapBuilder => FeatureMapBuilder = {

requests.incr()

searchResultsPageNumber.incr()

searchCandidateCount.incr()

searchQuerySource.incr()

searchQueryHasUser.incr()

\_.withConstantFeature(SearchResultsPageNumber, searchContext.resultsPageNumber)

.withConstantFeature(SearchCandidateCount, searchContext.candidateCount)

.withConstantFeature(

SearchQuerySource,

searchContext.querySourceOption match {

case Some(querySource) => querySource

case \_ => ThriftQuerySource.Unknown

})

.withConstantFeature(SearchQueryHasUser, searchContext.queryHasUser)

}

}