package com.twitter.search.earlybird\_root.filters;

import java.util.HashSet;

import java.util.Set;

import com.google.common.annotations.VisibleForTesting;

import com.google.common.collect.ImmutableSet;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.search.common.metrics.SearchCounter;

import com.twitter.search.common.util.earlybird.EarlybirdResponseUtil;

import com.twitter.search.earlybird.thrift.EarlybirdRequest;

import com.twitter.search.earlybird.thrift.EarlybirdResponse;

import com.twitter.search.earlybird\_root.common.EarlybirdRequestContext;

import com.twitter.search.queryparser.query.search.SearchOperatorConstants;

import com.twitter.search.queryparser.visitors.DetectPositiveOperatorVisitor;

/\*\*

\* Filter that is tracking the unexpected nullcast results from Earlybirds.

\*/

public class NullcastTrackingFilter extends SensitiveResultsTrackingFilter {

public NullcastTrackingFilter() {

super("unexpected nullcast tweets", true);

}

private static final Logger LOG = LoggerFactory.getLogger(NullcastTrackingFilter.class);

@VisibleForTesting

static final SearchCounter BAD\_NULLCAST\_QUERY\_COUNT =

SearchCounter.export("unexpected\_nullcast\_query\_count");

@VisibleForTesting

static final SearchCounter BAD\_NULLCAST\_RESULT\_COUNT =

SearchCounter.export("unexpected\_nullcast\_result\_count");

@Override

protected Logger getLogger() {

return LOG;

}

@Override

protected SearchCounter getSensitiveQueryCounter() {

return BAD\_NULLCAST\_QUERY\_COUNT;

}

@Override

protected SearchCounter getSensitiveResultsCounter() {

return BAD\_NULLCAST\_RESULT\_COUNT;

}

@Override

protected Set<Long> getSensitiveResults(EarlybirdRequestContext requestContext,

EarlybirdResponse earlybirdResponse) throws Exception {

if (!requestContext.getParsedQuery().accept(

new DetectPositiveOperatorVisitor(SearchOperatorConstants.NULLCAST))) {

return EarlybirdResponseUtil.findUnexpectedNullcastStatusIds(

earlybirdResponse.getSearchResults(), requestContext.getRequest());

} else {

return new HashSet<>();

}

}

/\*\*

\* Some Earlybird requests are not searches, instead, they are scoring requests.

\* These requests supply a list of IDs to be scored.

\* It is OK to return nullcast tweet result if the ID is supplied in the request.

\* This extracts the scoring request tweet IDs.

\*/

@Override

protected Set<Long> getExceptedResults(EarlybirdRequestContext requestContext) {

EarlybirdRequest request = requestContext.getRequest();

if (request == null

|| !request.isSetSearchQuery()

|| request.getSearchQuery().getSearchStatusIdsSize() == 0) {

return ImmutableSet.of();

}

return request.getSearchQuery().getSearchStatusIds();

}

}