package com.twitter.search.earlybird.search.relevance.collectors;

import java.io.IOException;

import java.util.List;

import com.google.common.collect.Lists;

import com.twitter.common.util.Clock;

import com.twitter.search.common.relevance.features.TweetIntegerShingleSignature;

import com.twitter.search.common.schema.base.ImmutableSchemaInterface;

import com.twitter.search.common.schema.earlybird.EarlybirdCluster;

import com.twitter.search.earlybird.common.userupdates.UserTable;

import com.twitter.search.earlybird.search.relevance.RelevanceHit;

import com.twitter.search.earlybird.search.relevance.RelevanceSearchRequestInfo;

import com.twitter.search.earlybird.search.relevance.RelevanceSearchResults;

import com.twitter.search.earlybird.search.relevance.scoring.ScoringFunction;

import com.twitter.search.earlybird.stats.EarlybirdSearcherStats;

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

/\*\*

\* RelevanceAllCollector is a results collector that collects all results sorted by score,

\* including signature-duplicates and results skipped by the scoring function.

\*/

public class RelevanceAllCollector extends AbstractRelevanceCollector {

// All results.

protected final List<RelevanceHit> results;

public RelevanceAllCollector(

ImmutableSchemaInterface schema,

RelevanceSearchRequestInfo searchRequestInfo,

ScoringFunction scoringFunction,

EarlybirdSearcherStats searcherStats,

EarlybirdCluster cluster,

UserTable userTable,

Clock clock,

int requestDebugMode) {

super(schema, searchRequestInfo, scoringFunction, searcherStats, cluster, userTable, clock,

requestDebugMode);

this.results = Lists.newArrayList();

}

@Override

protected void doCollectWithScore(long tweetID, float score) throws IOException {

ThriftSearchResultMetadata metadata = collectMetadata();

scoringFunction.populateResultMetadataBasedOnScoringData(

searchRequestInfo.getSearchQuery().getResultMetadataOptions(),

metadata,

scoringFunction.getScoringDataForCurrentDocument());

results.add(new RelevanceHit(

currTimeSliceID,

tweetID,

TweetIntegerShingleSignature.deserialize(metadata.getSignature()),

metadata));

}

@Override

protected RelevanceSearchResults doGetRelevanceResults() {

final int numResults = results.size();

RelevanceSearchResults searchResults = new RelevanceSearchResults(numResults);

// Insert hits in decreasing order by score.

results.sort(RelevanceHit.COMPARATOR\_BY\_SCORE);

for (int i = 0; i < numResults; i++) {

searchResults.setHit(results.get(i), i);

}

searchResults.setRelevanceStats(getRelevanceStats());

searchResults.setNumHits(numResults);

return searchResults;

}

}