package com.twitter.search.earlybird.search.facets;

import java.util.List;

import com.google.common.collect.ImmutableList;

import com.google.common.collect.Lists;

import com.google.common.collect.Sets;

import org.apache.commons.lang.StringUtils;

import com.twitter.escherbird.thriftjava.TweetEntityAnnotation;

import com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant;

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

public class EntityAnnotationCollector extends AbstractFacetTermCollector {

private List<TweetEntityAnnotation> annotations = Lists.newArrayList();

@Override

public boolean collect(int docID, long termID, int fieldID) {

String term = getTermFromFacet(termID, fieldID,

Sets.newHashSet(EarlybirdFieldConstant.ENTITY\_ID\_FIELD.getFieldName()));

if (StringUtils.isEmpty(term)) {

return false;

}

String[] idParts = term.split("\\.");

// Only include the full three-part form of the entity ID: "groupId.domainId.entityId"

// Exclude the less-specific forms we index: "domainId.entityId" and "entityId"

if (idParts.length < 3) {

return false;

}

annotations.add(new TweetEntityAnnotation(

Long.valueOf(idParts[0]),

Long.valueOf(idParts[1]),

Long.valueOf(idParts[2])));

return true;

}

@Override

public void fillResultAndClear(ThriftSearchResult result) {

getExtraMetadata(result).setEntityAnnotations(ImmutableList.copyOf(annotations));

annotations.clear();

}

}