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

import java.util.List;

import java.util.Map;

import com.google.common.collect.ImmutableList;

import com.google.common.collect.ImmutableMap;

import com.google.common.collect.Lists;

import org.apache.commons.lang.StringUtils;

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

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

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

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

public class NamedEntityCollector extends AbstractFacetTermCollector {

private static final Map<String, NamedEntitySource> NAMED\_ENTITY\_WITH\_TYPE\_FIELDS =

ImmutableMap.of(

EarlybirdFieldConstant.NAMED\_ENTITY\_WITH\_TYPE\_FROM\_TEXT\_FIELD.getFieldName(),

NamedEntitySource.TEXT,

EarlybirdFieldConstant.NAMED\_ENTITY\_WITH\_TYPE\_FROM\_URL\_FIELD.getFieldName(),

NamedEntitySource.URL);

private List<ThriftSearchResultNamedEntity> namedEntities = Lists.newArrayList();

@Override

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

String term = getTermFromFacet(termID, fieldID, NAMED\_ENTITY\_WITH\_TYPE\_FIELDS.keySet());

if (StringUtils.isEmpty(term)) {

return false;

}

int index = term.lastIndexOf(":");

namedEntities.add(new ThriftSearchResultNamedEntity(

term.substring(0, index),

term.substring(index + 1),

NAMED\_ENTITY\_WITH\_TYPE\_FIELDS.get(findFacetName(fieldID))));

return true;

}

@Override

public void fillResultAndClear(ThriftSearchResult result) {

getExtraMetadata(result).setNamedEntities(ImmutableList.copyOf(namedEntities));

namedEntities.clear();

}

}