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

import java.io.IOException;

import java.util.List;

import com.twitter.common.collections.Pair;

/\*\*

\* The collect() method is called for every document for which facets shall be counted.

\* This iterator then calls the FacetAccumulators for all facets that belong to the

\* current document.

\*/

public abstract class FacetCountIterator implements FacetTermCollector {

public static class IncrementData {

public FacetAccumulator[] accumulators;

public int weightedCountIncrement;

public int penaltyIncrement;

public int tweepCred;

public int languageId;

}

public IncrementData incrementData = new IncrementData();

private List<Pair<Integer, Long>> proofs = null;

void setIncrementData(IncrementData incrementData) {

this.incrementData = incrementData;

}

public void setProofs(List<Pair<Integer, Long>> proofs) {

this.proofs = proofs;

}

// interface method that collects a specific term in a specific field for this document.

@Override

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

FacetAccumulator accumulator = incrementData.accumulators[fieldID];

accumulator.add(termID, incrementData.weightedCountIncrement, incrementData.penaltyIncrement,

incrementData.tweepCred);

accumulator.recordLanguage(incrementData.languageId);

if (proofs != null) {

addProof(docID, termID, fieldID);

}

return true;

}

protected void addProof(int docID, long termID, int fieldID) {

proofs.add(new Pair<>(fieldID, termID));

}

/\*\*

\* Collected facets for the given document.

\*/

public abstract void collect(int docID) throws IOException;

}