package com.twitter.search.common.relevance.features;

import java.util.Map;

import com.google.common.collect.Maps;

import com.twitter.search.common.encoding.features.IntegerEncodedFeatures;

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

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

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

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

/\*\*

\* FeatureSink is used to write features based on feature configuration or feature name. After

\* all feature is written, the class can return the base field integer array values.

\*

\* This class is not thread-safe.

\*/

public class FeatureSink {

private ImmutableSchemaInterface schema;

private final Map<String, IntegerEncodedFeatures> encodedFeatureMap;

/\*\* Creates a new FeatureSink instance. \*/

public FeatureSink(ImmutableSchemaInterface schema) {

this.schema = schema;

this.encodedFeatureMap = Maps.newHashMap();

}

private IntegerEncodedFeatures getFeatures(String baseFieldName) {

IntegerEncodedFeatures features = encodedFeatureMap.get(baseFieldName);

if (features == null) {

features = EarlybirdEncodedFeatures.newEncodedTweetFeatures(schema, baseFieldName);

encodedFeatureMap.put(baseFieldName, features);

}

return features;

}

/\*\* Sets the given numeric value for the field. \*/

public FeatureSink setNumericValue(EarlybirdFieldConstant field, int value) {

return setNumericValue(field.getFieldName(), value);

}

/\*\* Sets the given numeric value for the feature with the given name. \*/

public FeatureSink setNumericValue(String featureName, int value) {

final FeatureConfiguration featureConfig = schema.getFeatureConfigurationByName(featureName);

if (featureConfig != null) {

getFeatures(featureConfig.getBaseField()).setFeatureValue(featureConfig, value);

}

return this;

}

/\*\* Sets the given boolean value for the given field. \*/

public FeatureSink setBooleanValue(EarlybirdFieldConstant field, boolean value) {

return setBooleanValue(field.getFieldName(), value);

}

/\*\* Sets the given boolean value for the feature with the given name. \*/

public FeatureSink setBooleanValue(String featureName, boolean value) {

final FeatureConfiguration featureConfig = schema.getFeatureConfigurationByName(featureName);

if (featureConfig != null) {

getFeatures(featureConfig.getBaseField()).setFlagValue(featureConfig, value);

}

return this;

}

/\*\* Returns the features for the given base field. \*/

public IntegerEncodedFeatures getFeaturesForBaseField(EarlybirdFieldConstant baseField) {

return getFeaturesForBaseField(baseField.getFieldName());

}

/\*\* Returns the features for the given base field. \*/

public IntegerEncodedFeatures getFeaturesForBaseField(String baseFieldName) {

return encodedFeatureMap.get(baseFieldName);

}

}