package com.twitter.search.common.schema.earlybird;

import java.util.Map;

import com.google.common.annotations.VisibleForTesting;

import com.google.common.base.Preconditions;

import com.google.common.collect.Maps;

import com.twitter.common.text.util.TokenStreamSerializer;

import com.twitter.search.common.metrics.SearchCounter;

import com.twitter.search.common.schema.AnalyzerFactory;

import com.twitter.search.common.schema.DynamicSchema;

import com.twitter.search.common.schema.ImmutableSchema;

import com.twitter.search.common.schema.SchemaBuilder;

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

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

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

import com.twitter.search.common.schema.thriftjava.ThriftCSFType;

import com.twitter.search.common.schema.thriftjava.ThriftFeatureUpdateConstraint;

import com.twitter.search.common.schema.thriftjava.ThriftSchema;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.BLINK\_FAVORITE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.BLINK\_QUOTE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.BLINK\_REPLY\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.BLINK\_RETWEET\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.COMPOSER\_SOURCE\_IS\_CAMERA\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.DECAYED\_FAVORITE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.DECAYED\_QUOTE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.DECAYED\_REPLY\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.DECAYED\_RETWEET\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EMBEDS\_IMPRESSION\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EMBEDS\_IMPRESSION\_COUNT\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EMBEDS\_URL\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EMBEDS\_URL\_COUNT\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_1;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_3;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_4;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_FEATURE\_UNUSED\_BITS\_0\_24\_8;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_12\_30\_2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_13\_30\_2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_14\_10\_22;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_16;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_17;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_18;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_19;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_20;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_4\_31\_1;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_7\_6\_26;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FAKE\_FAVORITE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FAKE\_QUOTE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FAKE\_REPLY\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FAKE\_RETWEET\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FAVORITE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FAVORITE\_COUNT\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FROM\_BLUE\_VERIFIED\_ACCOUNT\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.FROM\_VERIFIED\_ACCOUNT\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_CARD\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_CONSUMER\_VIDEO\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_EXPANDO\_CARD\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_IMAGE\_URL\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_LINK\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_MULTIPLE\_HASHTAGS\_OR\_TRENDS\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_MULTIPLE\_MEDIA\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_NATIVE\_IMAGE\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_NEWS\_URL\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_PERISCOPE\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_PRO\_VIDEO\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_QUOTE\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_TREND\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_VIDEO\_URL\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_VINE\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.HAS\_VISIBLE\_LINK\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_NULLCAST\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_OFFENSIVE\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_REPLY\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_RETWEET\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_SENSITIVE\_CONTENT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_TRENDING\_NOW\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_USER\_BOT\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_USER\_NEW\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_USER\_NSFW\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.IS\_USER\_SPAM\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LABEL\_ABUSIVE\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LABEL\_ABUSIVE\_HI\_RCL\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LABEL\_DUP\_CONTENT\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LABEL\_NSFW\_HI\_PRC\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LABEL\_NSFW\_HI\_RCL\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LABEL\_SPAM\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LABEL\_SPAM\_HI\_RCL\_FLAG;

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

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LAST\_FAVORITE\_SINCE\_CREATION\_HRS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LAST\_QUOTE\_SINCE\_CREATION\_HRS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LAST\_REPLY\_SINCE\_CREATION\_HRS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LAST\_RETWEET\_SINCE\_CREATION\_HRS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.LINK\_LANGUAGE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NORMALIZED\_FAVORITE\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NORMALIZED\_REPLY\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NORMALIZED\_RETWEET\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NUM\_HASHTAGS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NUM\_HASHTAGS\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NUM\_MENTIONS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NUM\_MENTIONS\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.NUM\_STOCKS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PARUS\_SCORE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PBLOCK\_SCORE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PERISCOPE\_EXISTS;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PERISCOPE\_HAS\_BEEN\_FEATURED;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PERISCOPE\_IS\_CURRENTLY\_FEATURED;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PERISCOPE\_IS\_FROM\_QUALITY\_SOURCE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PERISCOPE\_IS\_LIVE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PREV\_USER\_TWEET\_ENGAGEMENT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.PROFILE\_IS\_EGG\_FLAG;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.P\_REPORTED\_TWEET\_SCORE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.P\_SPAMMY\_TWEET\_SCORE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.QUOTE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.REFERENCE\_AUTHOR\_ID\_LEAST\_SIGNIFICANT\_INT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.REFERENCE\_AUTHOR\_ID\_MOST\_SIGNIFICANT\_INT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.REPLY\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.REPLY\_COUNT\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.RETWEET\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.RETWEET\_COUNT\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.SPAMMY\_TWEET\_CONTENT\_SCORE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.TEXT\_SCORE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.TOXICITY\_SCORE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.TWEET\_SIGNATURE;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.USER\_REPUTATION;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.VIDEO\_VIEW\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.VIDEO\_VIEW\_COUNT\_V2;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.VISIBLE\_TOKEN\_RATIO;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.WEIGHTED\_FAVORITE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.WEIGHTED\_QUOTE\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.WEIGHTED\_REPLY\_COUNT;

import static com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant.WEIGHTED\_RETWEET\_COUNT;

/\*\*

\* Field configurations for Earlybird.

\*/

public final class EarlybirdSchemaCreateTool {

// How many times a schema is built

private static final SearchCounter SCHEMA\_BUILD\_COUNT =

SearchCounter.export("schema\_build\_count");

// Number of integers for the column of ENCODED\_TWEET\_FEATURES\_FIELD.

@VisibleForTesting

public static final int NUMBER\_OF\_INTEGERS\_FOR\_FEATURES = 5;

// Number of integers for the column of EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD.

// extra 80 bytes

// In realtime cluster, assuming 19 segments total, and 8388608 docs per segment

// this would amount to about 12.75GB of memory needed

//

@VisibleForTesting

public static final int NUMBER\_OF\_INTEGERS\_FOR\_EXTENDED\_FEATURES = 20;

@VisibleForTesting

public static final Map<String, FeatureConfiguration> FEATURE\_CONFIGURATION\_MAP

= Maps.newLinkedHashMap();

public static final String BASE\_FIELD\_NAME =

EarlybirdFieldConstant.ENCODED\_TWEET\_FEATURES\_FIELD.getFieldName();

private static String getBaseFieldName(String fullName) {

int index = fullName.indexOf(SchemaBuilder.CSF\_VIEW\_NAME\_SEPARATOR);

Preconditions.checkArgument(index > 0);

return fullName.substring(0, index);

}

private static String getBaseFieldName(EarlybirdFieldConstant fieldConstant) {

return getBaseFieldName(fieldConstant.getFieldName());

}

private static String getFeatureNameInField(EarlybirdFieldConstant fieldConstant) {

int index = fieldConstant.getFieldName().indexOf(SchemaBuilder.CSF\_VIEW\_NAME\_SEPARATOR);

Preconditions.checkArgument(index > 0);

return fieldConstant.getFieldName().substring(index + 1);

}

// defining all features

static {

// Add individual tweet encoded features as views on top of

// EarlybirdFieldConstant.ENCODED\_TWEET\_FEATURES\_FIELD

// int intIndex, int bitStartPos, int bitLength

newEarlybirdFeatureConfiguration(IS\_RETWEET\_FLAG, ThriftCSFType.BOOLEAN, 0, 0, 1);

newEarlybirdFeatureConfiguration(IS\_OFFENSIVE\_FLAG, ThriftCSFType.BOOLEAN, 0, 1, 1);

newEarlybirdFeatureConfiguration(HAS\_LINK\_FLAG, ThriftCSFType.BOOLEAN, 0, 2, 1);

newEarlybirdFeatureConfiguration(HAS\_TREND\_FLAG, ThriftCSFType.BOOLEAN, 0, 3, 1);

newEarlybirdFeatureConfiguration(IS\_REPLY\_FLAG, ThriftCSFType.BOOLEAN, 0, 4, 1);

newEarlybirdFeatureConfiguration(IS\_SENSITIVE\_CONTENT, ThriftCSFType.BOOLEAN, 0, 5, 1);

newEarlybirdFeatureConfiguration(HAS\_MULTIPLE\_HASHTAGS\_OR\_TRENDS\_FLAG,

ThriftCSFType.BOOLEAN, 0, 6, 1);

newEarlybirdFeatureConfiguration(FROM\_VERIFIED\_ACCOUNT\_FLAG, ThriftCSFType.BOOLEAN, 0, 7, 1);

newEarlybirdFeatureConfiguration(TEXT\_SCORE, ThriftCSFType.INT, 0, 8, 8);

newEarlybirdFeatureConfiguration(LANGUAGE, ThriftCSFType.INT, 0, 16, 8);

newEarlybirdFeatureConfiguration(LINK\_LANGUAGE, ThriftCSFType.INT, 0, 24, 8);

newEarlybirdFeatureConfiguration(HAS\_IMAGE\_URL\_FLAG, ThriftCSFType.BOOLEAN, 1, 0, 1);

newEarlybirdFeatureConfiguration(HAS\_VIDEO\_URL\_FLAG, ThriftCSFType.BOOLEAN, 1, 1, 1);

newEarlybirdFeatureConfiguration(HAS\_NEWS\_URL\_FLAG, ThriftCSFType.BOOLEAN, 1, 2, 1);

newEarlybirdFeatureConfiguration(HAS\_EXPANDO\_CARD\_FLAG, ThriftCSFType.BOOLEAN, 1, 3, 1);

newEarlybirdFeatureConfiguration(HAS\_MULTIPLE\_MEDIA\_FLAG, ThriftCSFType.BOOLEAN, 1, 4, 1);

newEarlybirdFeatureConfiguration(PROFILE\_IS\_EGG\_FLAG, ThriftCSFType.BOOLEAN, 1, 5, 1);

newEarlybirdFeatureConfiguration(NUM\_MENTIONS, ThriftCSFType.INT, 1, 6, 2); // 0, 1, 2, 3+

newEarlybirdFeatureConfiguration(NUM\_HASHTAGS, ThriftCSFType.INT, 1, 8, 2); // 0, 1, 2, 3+

newEarlybirdFeatureConfiguration(HAS\_CARD\_FLAG, ThriftCSFType.BOOLEAN, 1, 10, 1);

newEarlybirdFeatureConfiguration(HAS\_VISIBLE\_LINK\_FLAG, ThriftCSFType.BOOLEAN, 1, 11, 1);

newEarlybirdFeatureConfiguration(USER\_REPUTATION, ThriftCSFType.INT, 1, 12, 8);

newEarlybirdFeatureConfiguration(IS\_USER\_SPAM\_FLAG, ThriftCSFType.BOOLEAN, 1, 20, 1);

newEarlybirdFeatureConfiguration(IS\_USER\_NSFW\_FLAG, ThriftCSFType.BOOLEAN, 1, 21, 1);

newEarlybirdFeatureConfiguration(IS\_USER\_BOT\_FLAG, ThriftCSFType.BOOLEAN, 1, 22, 1);

newEarlybirdFeatureConfiguration(IS\_USER\_NEW\_FLAG, ThriftCSFType.BOOLEAN, 1, 23, 1);

newEarlybirdFeatureConfiguration(PREV\_USER\_TWEET\_ENGAGEMENT, ThriftCSFType.INT, 1, 24, 6);

newEarlybirdFeatureConfiguration(COMPOSER\_SOURCE\_IS\_CAMERA\_FLAG,

ThriftCSFType.BOOLEAN, 1, 30, 1);

newEarlybirdFeatureConfiguration(IS\_NULLCAST\_FLAG, ThriftCSFType.BOOLEAN, 1, 31, 1);

newEarlybirdFeatureConfiguration(RETWEET\_COUNT, ThriftCSFType.DOUBLE, 2, 0, 8,

ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(FAVORITE\_COUNT, ThriftCSFType.DOUBLE, 2, 8, 8,

ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(REPLY\_COUNT, ThriftCSFType.DOUBLE, 2, 16, 8,

ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(PARUS\_SCORE, ThriftCSFType.DOUBLE, 2, 24, 8);

newEarlybirdFeatureConfiguration(HAS\_CONSUMER\_VIDEO\_FLAG, ThriftCSFType.BOOLEAN, 3, 0, 1);

newEarlybirdFeatureConfiguration(HAS\_PRO\_VIDEO\_FLAG, ThriftCSFType.BOOLEAN, 3, 1, 1);

newEarlybirdFeatureConfiguration(HAS\_VINE\_FLAG, ThriftCSFType.BOOLEAN, 3, 2, 1);

newEarlybirdFeatureConfiguration(HAS\_PERISCOPE\_FLAG, ThriftCSFType.BOOLEAN, 3, 3, 1);

newEarlybirdFeatureConfiguration(HAS\_NATIVE\_IMAGE\_FLAG, ThriftCSFType.BOOLEAN, 3, 4, 1);

// NOTE: There are 3 bits left in the first byte of INT 3, if possible, please reserve them

// for future media types (SEARCH-9131)

// newEarlybirdFeatureConfiguration(FUTURE\_MEDIA\_BITS, ThriftCSFType.INT, 3, 5, 3);

newEarlybirdFeatureConfiguration(VISIBLE\_TOKEN\_RATIO, ThriftCSFType.INT, 3, 8, 4);

newEarlybirdFeatureConfiguration(HAS\_QUOTE\_FLAG, ThriftCSFType.BOOLEAN, 3, 12, 1);

newEarlybirdFeatureConfiguration(FROM\_BLUE\_VERIFIED\_ACCOUNT\_FLAG,

ThriftCSFType.BOOLEAN, 3, 13, 1);

// Unused bits from bit 14 to bit 31 (18 bits)

// newEarlybirdFeatureConfiguration(UNUSED\_BITS, ThriftCSFType.INT, 3, 14, 18);

newEarlybirdFeatureConfiguration(TWEET\_SIGNATURE, ThriftCSFType.INT, 4, 0, 32);

newEarlybirdFeatureConfiguration(EMBEDS\_IMPRESSION\_COUNT,

ThriftCSFType.DOUBLE, 0, 0, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(EMBEDS\_URL\_COUNT,

ThriftCSFType.DOUBLE, 0, 8, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(VIDEO\_VIEW\_COUNT,

ThriftCSFType.DOUBLE, 0, 16, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

// Unused bits from bit 24 to bit 31 (8 bits).

// This used to be a feature that was decommissioned (SEARCHQUAL-10321)

newEarlybirdFeatureConfiguration(EXTENDED\_FEATURE\_UNUSED\_BITS\_0\_24\_8,

ThriftCSFType.INT, 0, 24, 8);

newEarlybirdFeatureConfiguration(REFERENCE\_AUTHOR\_ID\_LEAST\_SIGNIFICANT\_INT,

ThriftCSFType.INT, 1, 0, 32, ThriftFeatureUpdateConstraint.IMMUTABLE);

newEarlybirdFeatureConfiguration(REFERENCE\_AUTHOR\_ID\_MOST\_SIGNIFICANT\_INT,

ThriftCSFType.INT, 2, 0, 32, ThriftFeatureUpdateConstraint.IMMUTABLE);

newEarlybirdFeatureConfiguration(RETWEET\_COUNT\_V2,

ThriftCSFType.DOUBLE, 3, 0, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(FAVORITE\_COUNT\_V2,

ThriftCSFType.DOUBLE, 3, 8, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(REPLY\_COUNT\_V2,

ThriftCSFType.DOUBLE, 3, 16, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(EMBEDS\_IMPRESSION\_COUNT\_V2,

ThriftCSFType.DOUBLE, 3, 24, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(EMBEDS\_URL\_COUNT\_V2,

ThriftCSFType.DOUBLE, 4, 0, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(VIDEO\_VIEW\_COUNT\_V2,

ThriftCSFType.DOUBLE, 4, 8, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(QUOTE\_COUNT,

ThriftCSFType.DOUBLE, 4, 16, 8);

newEarlybirdFeatureConfiguration(LABEL\_ABUSIVE\_FLAG, ThriftCSFType.BOOLEAN, 4, 24, 1);

newEarlybirdFeatureConfiguration(LABEL\_ABUSIVE\_HI\_RCL\_FLAG, ThriftCSFType.BOOLEAN, 4, 25, 1);

newEarlybirdFeatureConfiguration(LABEL\_DUP\_CONTENT\_FLAG, ThriftCSFType.BOOLEAN, 4, 26, 1);

newEarlybirdFeatureConfiguration(LABEL\_NSFW\_HI\_PRC\_FLAG, ThriftCSFType.BOOLEAN, 4, 27, 1);

newEarlybirdFeatureConfiguration(LABEL\_NSFW\_HI\_RCL\_FLAG, ThriftCSFType.BOOLEAN, 4, 28, 1);

newEarlybirdFeatureConfiguration(LABEL\_SPAM\_FLAG, ThriftCSFType.BOOLEAN, 4, 29, 1);

newEarlybirdFeatureConfiguration(LABEL\_SPAM\_HI\_RCL\_FLAG, ThriftCSFType.BOOLEAN, 4, 30, 1);

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_4\_31\_1,

ThriftCSFType.INT, 4, 31, 1);

newEarlybirdFeatureConfiguration(WEIGHTED\_RETWEET\_COUNT,

ThriftCSFType.DOUBLE, 5, 0, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(WEIGHTED\_REPLY\_COUNT,

ThriftCSFType.DOUBLE, 5, 8, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(WEIGHTED\_FAVORITE\_COUNT,

ThriftCSFType.DOUBLE, 5, 16, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(WEIGHTED\_QUOTE\_COUNT,

ThriftCSFType.DOUBLE, 5, 24, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(PERISCOPE\_EXISTS,

ThriftCSFType.BOOLEAN, 6, 0, 1);

newEarlybirdFeatureConfiguration(PERISCOPE\_HAS\_BEEN\_FEATURED,

ThriftCSFType.BOOLEAN, 6, 1, 1);

newEarlybirdFeatureConfiguration(PERISCOPE\_IS\_CURRENTLY\_FEATURED,

ThriftCSFType.BOOLEAN, 6, 2, 1);

newEarlybirdFeatureConfiguration(PERISCOPE\_IS\_FROM\_QUALITY\_SOURCE,

ThriftCSFType.BOOLEAN, 6, 3, 1);

newEarlybirdFeatureConfiguration(PERISCOPE\_IS\_LIVE,

ThriftCSFType.BOOLEAN, 6, 4, 1);

newEarlybirdFeatureConfiguration(IS\_TRENDING\_NOW\_FLAG,

ThriftCSFType.BOOLEAN, 6, 5, 1);

// remaining bits for integer 6

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_7\_6\_26,

ThriftCSFType.INT, 6, 6, 26);

// The decaying counters can become smaller

newEarlybirdFeatureConfiguration(DECAYED\_RETWEET\_COUNT,

ThriftCSFType.DOUBLE, 7, 0, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(DECAYED\_REPLY\_COUNT,

ThriftCSFType.DOUBLE, 7, 8, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(DECAYED\_FAVORITE\_COUNT,

ThriftCSFType.DOUBLE, 7, 16, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(DECAYED\_QUOTE\_COUNT,

ThriftCSFType.DOUBLE, 7, 24, 8, ThriftFeatureUpdateConstraint.POSITIVE);

// The fake engagement counters.

newEarlybirdFeatureConfiguration(FAKE\_RETWEET\_COUNT,

ThriftCSFType.DOUBLE, 8, 0, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(FAKE\_REPLY\_COUNT,

ThriftCSFType.DOUBLE, 8, 8, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(FAKE\_FAVORITE\_COUNT,

ThriftCSFType.DOUBLE, 8, 16, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(FAKE\_QUOTE\_COUNT,

ThriftCSFType.DOUBLE, 8, 24, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(LAST\_RETWEET\_SINCE\_CREATION\_HRS,

ThriftCSFType.INT, 9, 0, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(LAST\_REPLY\_SINCE\_CREATION\_HRS,

ThriftCSFType.INT, 9, 8, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(LAST\_FAVORITE\_SINCE\_CREATION\_HRS,

ThriftCSFType.INT, 9, 16, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(LAST\_QUOTE\_SINCE\_CREATION\_HRS,

ThriftCSFType.INT, 9, 24, 8, ThriftFeatureUpdateConstraint.INC\_ONLY);

newEarlybirdFeatureConfiguration(NUM\_HASHTAGS\_V2,

ThriftCSFType.INT, 10, 0, 4);

newEarlybirdFeatureConfiguration(NUM\_MENTIONS\_V2,

ThriftCSFType.INT, 10, 4, 4);

newEarlybirdFeatureConfiguration(NUM\_STOCKS,

ThriftCSFType.INT, 10, 8, 4);

// Remaining bits for integer 10

// Production Toxicity and PBlock score from HML (go/toxicity, go/pblock)

newEarlybirdFeatureConfiguration(TOXICITY\_SCORE,

ThriftCSFType.DOUBLE, 10, 12, 10);

newEarlybirdFeatureConfiguration(PBLOCK\_SCORE,

ThriftCSFType.DOUBLE, 10, 22, 10);

// The blink engagement counters

newEarlybirdFeatureConfiguration(BLINK\_RETWEET\_COUNT,

ThriftCSFType.DOUBLE, 11, 0, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(BLINK\_REPLY\_COUNT,

ThriftCSFType.DOUBLE, 11, 8, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(BLINK\_FAVORITE\_COUNT,

ThriftCSFType.DOUBLE, 11, 16, 8, ThriftFeatureUpdateConstraint.POSITIVE);

newEarlybirdFeatureConfiguration(BLINK\_QUOTE\_COUNT,

ThriftCSFType.DOUBLE, 11, 24, 8, ThriftFeatureUpdateConstraint.POSITIVE);

// Experimental health model scores from HML

newEarlybirdFeatureConfiguration(EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_1,

ThriftCSFType.DOUBLE, 12, 0, 10);

newEarlybirdFeatureConfiguration(EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_2,

ThriftCSFType.DOUBLE, 12, 10, 10);

newEarlybirdFeatureConfiguration(EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_3,

ThriftCSFType.DOUBLE, 12, 20, 10);

// remaining bits for integer 12

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_12\_30\_2,

ThriftCSFType.INT, 12, 30, 2);

// Experimental health model scores from HML (cont.)

newEarlybirdFeatureConfiguration(EXPERIMENTAL\_HEALTH\_MODEL\_SCORE\_4,

ThriftCSFType.DOUBLE, 13, 0, 10);

// Production pSpammyTweet score from HML (go/pspammytweet)

newEarlybirdFeatureConfiguration(P\_SPAMMY\_TWEET\_SCORE,

ThriftCSFType.DOUBLE, 13, 10, 10);

// Production pReportedTweet score from HML (go/preportedtweet)

newEarlybirdFeatureConfiguration(P\_REPORTED\_TWEET\_SCORE,

ThriftCSFType.DOUBLE, 13, 20, 10);

// remaining bits for integer 13

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_13\_30\_2,

ThriftCSFType.INT, 13, 30, 2);

// Experimental health model scores from HML (cont.)

// Prod Spammy Tweet Content model score from Platform Manipulation (go/spammy-tweet-content)

newEarlybirdFeatureConfiguration(SPAMMY\_TWEET\_CONTENT\_SCORE,

ThriftCSFType.DOUBLE, 14, 0, 10);

// remaining bits for integer 14

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_14\_10\_22,

ThriftCSFType.INT, 14, 10, 22);

// Note that the integer index below is 0-based, but the index j in UNUSED\_BITS\_{j} below

// is 1-based.

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_16,

ThriftCSFType.INT, 15, 0, 32);

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_17,

ThriftCSFType.INT, 16, 0, 32);

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_18,

ThriftCSFType.INT, 17, 0, 32);

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_19,

ThriftCSFType.INT, 18, 0, 32);

newEarlybirdFeatureConfiguration(EXTENDED\_TEST\_FEATURE\_UNUSED\_BITS\_20,

ThriftCSFType.INT, 19, 0, 32);

}

private EarlybirdSchemaCreateTool() { }

/\*\*

\* Get schema for the Earlybird.

\*/

public static DynamicSchema buildSchema(EarlybirdCluster cluster)

throws Schema.SchemaValidationException {

SCHEMA\_BUILD\_COUNT.increment();

return new DynamicSchema(new ImmutableSchema(buildThriftSchema(cluster),

new AnalyzerFactory(),

cluster.getNameForStats()));

}

/\*\*

\* Get schema for the Earlybird, can throw runtime exception. This is mostly for static schema

\* usage, which does not care about schema updates.

\*/

@VisibleForTesting

public static DynamicSchema buildSchemaWithRuntimeException(EarlybirdCluster cluster) {

try {

return buildSchema(cluster);

} catch (Schema.SchemaValidationException e) {

throw new RuntimeException(e);

}

}

private static FeatureConfiguration newEarlybirdFeatureConfiguration(

EarlybirdFieldConstant fieldConstant,

ThriftCSFType type,

int intIndex, int bitStartPos, int bitLength,

ThriftFeatureUpdateConstraint... constraints) {

if (!fieldConstant.isFlagFeatureField() && type == ThriftCSFType.BOOLEAN) {

throw new IllegalArgumentException(

"Non-flag feature field configured with boolean Thrift type: " + fieldConstant);

}

if (fieldConstant.isFlagFeatureField() && type != ThriftCSFType.BOOLEAN) {

throw new IllegalArgumentException(

"Flag feature field configured with non-boolean Thrift type: " + fieldConstant);

}

String baseFieldName = getBaseFieldName(fieldConstant);

String name = getFeatureNameInField(fieldConstant);

FeatureConfiguration.Builder builder = FeatureConfiguration.builder()

.withName(name)

.withType(type)

.withBitRange(intIndex, bitStartPos, bitLength);

// remove the following line once we configure features purely by the schema

builder.withBaseField(baseFieldName);

if (!fieldConstant.isUnusedField()) {

builder.withOutputType(type);

}

if (fieldConstant.getFeatureNormalizationType() != null) {

builder.withFeatureNormalizationType(fieldConstant.getFeatureNormalizationType());

}

for (ThriftFeatureUpdateConstraint constraint : constraints) {

builder.withFeatureUpdateConstraint(constraint);

}

FeatureConfiguration featureConfiguration = builder.build();

FEATURE\_CONFIGURATION\_MAP.put(fieldConstant.getFieldName(), featureConfiguration);

return featureConfiguration;

}

/\*\*

\* Build ThriftSchema for the Earlybird. Note that the schema returned can be used

\* all Earlybird clusters. However, some clusters may not use all the field configurations.

\*/

@VisibleForTesting

public static ThriftSchema buildThriftSchema(EarlybirdCluster cluster) {

EarlybirdSchemaBuilder builder = new EarlybirdSchemaBuilder(

new EarlybirdFieldConstants(), cluster, TokenStreamSerializer.Version.VERSION\_2);

builder.withSchemaVersion(

FlushVersion.CURRENT\_FLUSH\_VERSION.getVersionNumber(),

FlushVersion.CURRENT\_FLUSH\_VERSION.getMinorVersion(),

FlushVersion.CURRENT\_FLUSH\_VERSION.getDescription(),

FlushVersion.CURRENT\_FLUSH\_VERSION.isOfficial());

// ID field, used for partitioning

builder.withPartitionFieldId(0)

.withSortableLongTermField(EarlybirdFieldConstant.ID\_FIELD.getFieldName())

// Text Fields that are searched by default

.withTextField(EarlybirdFieldConstant.RESOLVED\_LINKS\_TEXT\_FIELD.getFieldName(), true)

.withSearchFieldByDefault(

EarlybirdFieldConstant.RESOLVED\_LINKS\_TEXT\_FIELD.getFieldName(), 0.1f)

.withPretokenizedTextField(EarlybirdFieldConstant.TEXT\_FIELD.getFieldName(), true)

.withSearchFieldByDefault(EarlybirdFieldConstant.TEXT\_FIELD.getFieldName(), 1.0f);

builder.withTweetSpecificNormalization(EarlybirdFieldConstant.TEXT\_FIELD.getFieldName())

.withTextField(EarlybirdFieldConstant.TOKENIZED\_FROM\_USER\_FIELD.getFieldName(), true)

.withSearchFieldByDefault(

EarlybirdFieldConstant.TOKENIZED\_FROM\_USER\_FIELD.getFieldName(), 0.2f)

// Text fields not searched by default

.withTextField(EarlybirdFieldConstant.FROM\_USER\_FIELD.getFieldName(), false)

.withTextField(EarlybirdFieldConstant.TO\_USER\_FIELD.getFieldName(), false)

// cards are not searched by default, and have weight 0.

.withPretokenizedTextField(EarlybirdFieldConstant.CARD\_TITLE\_FIELD.getFieldName(), false)

.withPretokenizedTextField(

EarlybirdFieldConstant.CARD\_DESCRIPTION\_FIELD.getFieldName(), false)

.withTextField(EarlybirdFieldConstant.CARD\_LANG.getFieldName(), false)

// Out-of-order append fields

.withLongTermField(EarlybirdFieldConstant.LIKED\_BY\_USER\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.RETWEETED\_BY\_USER\_ID.getFieldName())

.withLongTermField(EarlybirdFieldConstant.REPLIED\_TO\_BY\_USER\_ID.getFieldName())

// No Position fields, sorted alphabetically

.withPretokenizedNoPositionField(EarlybirdFieldConstant.CARD\_DOMAIN\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.CARD\_NAME\_FIELD.getFieldName())

.withIntTermField(EarlybirdFieldConstant.CREATED\_AT\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.ENTITY\_ID\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.GEO\_HASH\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.FROM\_USER\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.IN\_REPLY\_TO\_TWEET\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.IN\_REPLY\_TO\_USER\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.RETWEET\_SOURCE\_TWEET\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.RETWEET\_SOURCE\_USER\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.CONVERSATION\_ID\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.PLACE\_ID\_FIELD.getFieldName())

.withTextField(EarlybirdFieldConstant.PLACE\_FULL\_NAME\_FIELD.getFieldName(), false)

.withIndexedNotTokenizedField(

EarlybirdFieldConstant.PLACE\_COUNTRY\_CODE\_FIELD.getFieldName())

.withIndexedNotTokenizedField(

EarlybirdFieldConstant.PROFILE\_GEO\_COUNTRY\_CODE\_FIELD.getFieldName())

.withTextField(EarlybirdFieldConstant.PROFILE\_GEO\_REGION\_FIELD.getFieldName(), false)

.withTextField(EarlybirdFieldConstant.PROFILE\_GEO\_LOCALITY\_FIELD.getFieldName(), false)

.withTermTextLookup(EarlybirdFieldConstant.FROM\_USER\_ID\_FIELD.getFieldName())

.withTermTextLookup(EarlybirdFieldConstant.IN\_REPLY\_TO\_USER\_ID\_FIELD.getFieldName())

.withPretokenizedNoPositionField(EarlybirdFieldConstant.HASHTAGS\_FIELD.getFieldName())

.withIndexedNotTokenizedField(ImmutableSchema.HF\_PHRASE\_PAIRS\_FIELD)

.withIndexedNotTokenizedField(ImmutableSchema.HF\_TERM\_PAIRS\_FIELD)

.withIndexedNotTokenizedField(EarlybirdFieldConstant.IMAGE\_LINKS\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.INTERNAL\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.ISO\_LANGUAGE\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.LINKS\_FIELD.getFieldName())

.withIntTermField(EarlybirdFieldConstant.LINK\_CATEGORY\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.MENTIONS\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.NEWS\_LINKS\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.NORMALIZED\_SOURCE\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.PLACE\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.SOURCE\_FIELD.getFieldName())

.withPretokenizedNoPositionField(EarlybirdFieldConstant.STOCKS\_FIELD.getFieldName())

.withIndexedNotTokenizedField(EarlybirdFieldConstant.VIDEO\_LINKS\_FIELD.getFieldName())

.withIntTermField(NORMALIZED\_FAVORITE\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD.getFieldName())

.withIntTermField(NORMALIZED\_REPLY\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD.getFieldName())

.withIntTermField(NORMALIZED\_RETWEET\_COUNT\_GREATER\_THAN\_OR\_EQUAL\_TO\_FIELD.getFieldName())

.withIntTermField(EarlybirdFieldConstant.COMPOSER\_SOURCE.getFieldName())

.withLongTermField(EarlybirdFieldConstant.QUOTED\_TWEET\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.QUOTED\_USER\_ID\_FIELD.getFieldName())

.withLongTermField(EarlybirdFieldConstant.DIRECTED\_AT\_USER\_ID\_FIELD.getFieldName())

// Named entity fields

.withIndexedNotTokenizedField(

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

.withIndexedNotTokenizedField(

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

.withIndexedNotTokenizedField(

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

.withIndexedNotTokenizedField(

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

// camelCase-tokenized user handles and tokenized user names, not searchable by default

.withPretokenizedTextField(

EarlybirdFieldConstant.CAMELCASE\_USER\_HANDLE\_FIELD.getFieldName(), false)

.withPretokenizedTextField(

EarlybirdFieldConstant.TOKENIZED\_USER\_NAME\_FIELD.getFieldName(), false)

.withIndexedNotTokenizedField(

EarlybirdFieldConstant.SPACE\_ID\_FIELD.getFieldName())

.withTextField(EarlybirdFieldConstant.SPACE\_ADMIN\_FIELD.getFieldName(), false)

.withPretokenizedTextField(EarlybirdFieldConstant.SPACE\_TITLE\_FIELD.getFieldName(), false)

.withTextField(EarlybirdFieldConstant.TOKENIZED\_SPACE\_ADMIN\_FIELD.getFieldName(), true)

.withPretokenizedTextField(

EarlybirdFieldConstant.CAMELCASE\_TOKENIZED\_SPACE\_ADMIN\_FIELD.getFieldName(), false)

.withPretokenizedTextField(

EarlybirdFieldConstant.TOKENIZED\_SPACE\_ADMIN\_DISPLAY\_NAME\_FIELD.getFieldName(), false)

.withPretokenizedTextField(

EarlybirdFieldConstant.URL\_DESCRIPTION\_FIELD.getFieldName(), false)

.withPretokenizedTextField(

EarlybirdFieldConstant.URL\_TITLE\_FIELD.getFieldName(), false);

builder

.withPhotoUrlFacetField(EarlybirdFieldConstant.TWIMG\_LINKS\_FIELD.getFieldName())

.withOutOfOrderEnabledForField(

EarlybirdFieldConstant.LIKED\_BY\_USER\_ID\_FIELD.getFieldName())

.withOutOfOrderEnabledForField(

EarlybirdFieldConstant.RETWEETED\_BY\_USER\_ID.getFieldName())

.withOutOfOrderEnabledForField(

EarlybirdFieldConstant.REPLIED\_TO\_BY\_USER\_ID.getFieldName());

// ColumnStrideFields.

boolean loadCSFIntoRAMDefault = cluster != EarlybirdCluster.FULL\_ARCHIVE;

builder

.withColumnStrideField(EarlybirdFieldConstants.ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

ThriftCSFType.INT, NUMBER\_OF\_INTEGERS\_FOR\_FEATURES,

true, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.FROM\_USER\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, /\* the full archive loads this field into RAM \*/ true)

.withColumnStrideField(EarlybirdFieldConstant.SHARED\_STATUS\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.CARD\_TYPE\_CSF\_FIELD.getFieldName(),

ThriftCSFType.BYTE, 1, false, loadCSFIntoRAMDefault)

// CSF Used by archive mappers

.withColumnStrideField(EarlybirdFieldConstant.CREATED\_AT\_CSF\_FIELD.getFieldName(),

ThriftCSFType.INT, 1, false, /\* the full archive loads this field into RAM \*/ true)

.withColumnStrideField(EarlybirdFieldConstant.ID\_CSF\_FIELD.getFieldName(),

ThriftCSFType.LONG, 1, false, /\* the full archive loads this field into RAM \*/ true)

.withColumnStrideField(EarlybirdFieldConstant.LAT\_LON\_CSF\_FIELD.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.CONVERSATION\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.QUOTED\_TWEET\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.QUOTED\_USER\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.CARD\_LANG\_CSF.getFieldName(),

ThriftCSFType.INT, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.CARD\_URI\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.DIRECTED\_AT\_USER\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(EarlybirdFieldConstant.REFERENCE\_AUTHOR\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

.withColumnStrideField(

EarlybirdFieldConstant.EXCLUSIVE\_CONVERSATION\_AUTHOR\_ID\_CSF.getFieldName(),

ThriftCSFType.LONG, 1, false, loadCSFIntoRAMDefault)

/\* Semicolon on separate line to preserve git blame. \*/;

builder.withColumnStrideField(

EarlybirdFieldConstants.EXTENDED\_ENCODED\_TWEET\_FEATURES\_FIELD\_NAME,

ThriftCSFType.INT, NUMBER\_OF\_INTEGERS\_FOR\_EXTENDED\_FEATURES,

true, loadCSFIntoRAMDefault);

for (Map.Entry<String, FeatureConfiguration> entry : FEATURE\_CONFIGURATION\_MAP.entrySet()) {

String fullName = entry.getKey();

String baseName = getBaseFieldName(fullName);

EarlybirdFieldConstant fieldConstant = EarlybirdFieldConstants.getFieldConstant(fullName);

if (fieldConstant.isValidFieldInCluster(cluster)) {

builder.withFeatureConfiguration(baseName, fullName, entry.getValue());

}

}

// Add facet settings for facet fields

// boolean args are respectively whether to use skiplist, whether offensive, whether to use CSF

builder

.withFacetConfigs(EarlybirdFieldConstant.MENTIONS\_FIELD.getFieldName(),

EarlybirdFieldConstant.MENTIONS\_FACET, true, false, false)

.withFacetConfigs(EarlybirdFieldConstant.HASHTAGS\_FIELD.getFieldName(),

EarlybirdFieldConstant.HASHTAGS\_FACET, true, false, false)

.withFacetConfigs(EarlybirdFieldConstant.STOCKS\_FIELD.getFieldName(),

EarlybirdFieldConstant.STOCKS\_FACET, true, false, false)

.withFacetConfigs(EarlybirdFieldConstant.IMAGE\_LINKS\_FIELD.getFieldName(),

EarlybirdFieldConstant.IMAGES\_FACET, true, true, false)

.withFacetConfigs(EarlybirdFieldConstant.VIDEO\_LINKS\_FIELD.getFieldName(),

EarlybirdFieldConstant.VIDEOS\_FACET, true, true, false)

.withFacetConfigs(EarlybirdFieldConstant.NEWS\_LINKS\_FIELD.getFieldName(),

EarlybirdFieldConstant.NEWS\_FACET, true, false, false)

.withFacetConfigs(EarlybirdFieldConstant.ISO\_LANGUAGE\_FIELD.getFieldName(),

EarlybirdFieldConstant.LANGUAGES\_FACET, false, false, false)

.withFacetConfigs(EarlybirdFieldConstant.SOURCE\_FIELD.getFieldName(),

EarlybirdFieldConstant.SOURCES\_FACET, false, false, false)

.withFacetConfigs(EarlybirdFieldConstant.TWIMG\_LINKS\_FIELD.getFieldName(),

EarlybirdFieldConstant.TWIMG\_FACET, true, true, false)

.withFacetConfigs(EarlybirdFieldConstant.FROM\_USER\_ID\_CSF.getFieldName(),

EarlybirdFieldConstant.FROM\_USER\_ID\_FACET, false, false, true /\* facet on CSF \*/)

.withFacetConfigs(EarlybirdFieldConstant.SHARED\_STATUS\_ID\_CSF.getFieldName(),

EarlybirdFieldConstant.RETWEETS\_FACET, false, false, true /\* facet on CSF \*/)

.withFacetConfigs(EarlybirdFieldConstant.LINKS\_FIELD.getFieldName(),

EarlybirdFieldConstant.LINKS\_FACET, true, false, false)

.withFacetConfigs(

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

true, false, false)

.withFacetConfigs(

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

true, false, false)

.withFacetConfigs(

EarlybirdFieldConstant.ENTITY\_ID\_FIELD.getFieldName(),

true, false, false)

.withFacetConfigs(EarlybirdFieldConstant.SPACE\_ID\_FIELD.getFieldName(),

EarlybirdFieldConstant.SPACES\_FACET, true, false, false);

return builder.build();

}

}