package com.twitter.follow\_recommendations.common.constants

import com.twitter.hermit.constants.AlgorithmFeedbackTokens.AlgorithmToFeedbackTokenMap

import com.twitter.hermit.model.Algorithm.\_

import com.twitter.follow\_recommendations.common.models.AlgorithmType

object CandidateAlgorithmTypeConstants {

/\*\*

\* Each algorithm is based on one, or more, of the 4 types of information we have on users,

\* described in [[AlgorithmType]]. Assignment of algorithms to these categories are based on

\*/

private val AlgorithmIdToType: Map[String, Set[AlgorithmType.Value]] = Map(

// Activity Algorithms:

AlgorithmToFeedbackTokenMap(NewFollowingSimilarUser).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(Sims).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(NewFollowingSimilarUserSalsa).toString -> Set(

AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(RecentEngagementNonDirectFollow).toString -> Set(

AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(RecentEngagementSimilarUser).toString -> Set(

AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(RecentEngagementSarusOcCur).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(RecentSearchBasedRec).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(TwistlyTweetAuthors).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(Follow2VecNearestNeighbors).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(EmailTweetClick).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(RepeatedProfileVisits).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(GoodTweetClickEngagements).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(TweetShareEngagements).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(TweetSharerToShareRecipientEngagements).toString -> Set(

AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(TweetAuthorToShareRecipientEngagements).toString -> Set(

AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(LinearRegressionFollow2VecNearestNeighbors).toString -> Set(

AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(NUXLOHistory).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(TrafficAttributionAccounts).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(RealGraphOonV2).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(MagicRecsRecentEngagements).toString -> Set(AlgorithmType.Activity),

AlgorithmToFeedbackTokenMap(NotificationEngagement).toString -> Set(AlgorithmType.Activity),

// Social Algorithms:

AlgorithmToFeedbackTokenMap(TwoHopRandomWalk).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(RealTimeMutualFollow).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(ForwardPhoneBook).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(ForwardEmailBook).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(NewFollowingNewFollowingExpansion).toString -> Set(

AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(NewFollowingSarusCoOccurSocialProof).toString -> Set(

AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(ReverseEmailBookIbis).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(ReversePhoneBook).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(StrongTiePredictionRec).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(StrongTiePredictionRecWithSocialProof).toString -> Set(

AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(OnlineStrongTiePredictionRec).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(OnlineStrongTiePredictionRecNoCaching).toString -> Set(

AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(TriangularLoop).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(StrongTiePredictionPmi).toString -> Set(AlgorithmType.Social),

AlgorithmToFeedbackTokenMap(OnlineStrongTiePredictionRAB).toString -> Set(AlgorithmType.Social),

// Geo Algorithms:

AlgorithmToFeedbackTokenMap(PopCountryBackFill).toString -> Set(AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(PopCountry).toString -> Set(AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(PopGeohash).toString -> Set(AlgorithmType.Geo),

// AlgorithmToFeedbackTokenMap(PopGeohashRealGraph).toString -> Set(AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(EngagedFollowerRatio).toString -> Set(AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(CrowdSearchAccounts).toString -> Set(AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(OrganicFollowAccounts).toString -> Set(AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(PopGeohashQualityFollow).toString -> Set(AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(PPMILocaleFollow).toString -> Set(AlgorithmType.Geo),

// Interest Algorithms:

AlgorithmToFeedbackTokenMap(TttInterest).toString -> Set(AlgorithmType.Interest),

AlgorithmToFeedbackTokenMap(UttInterestRelatedUsers).toString -> Set(AlgorithmType.Interest),

AlgorithmToFeedbackTokenMap(UttSeedAccounts).toString -> Set(AlgorithmType.Interest),

AlgorithmToFeedbackTokenMap(UttProducerExpansion).toString -> Set(AlgorithmType.Interest),

// Hybrid (more than one type) Algorithms:

AlgorithmToFeedbackTokenMap(UttProducerOfflineMbcgV1).toString -> Set(

AlgorithmType.Interest,

AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(CuratedAccounts).toString -> Set(

AlgorithmType.Interest,

AlgorithmType.Geo),

AlgorithmToFeedbackTokenMap(UserUserGraph).toString -> Set(

AlgorithmType.Social,

AlgorithmType.Activity),

)

def getAlgorithmTypes(algoId: String): Set[String] = {

AlgorithmIdToType.get(algoId).map(\_.map(\_.toString)).getOrElse(Set.empty)

}

}