package com.twitter.representationscorer.scorestore

import com.twitter.simclusters\_v2.score.WeightedSumAggregatedScoreStore

import com.twitter.simclusters\_v2.score.WeightedSumAggregatedScoreStore.WeightedSumAggregatedScoreParameter

import com.twitter.simclusters\_v2.thriftscala.{EmbeddingType, ModelVersion, ScoringAlgorithm}

object TopicTweetRankingScoreStore {

val producerEmbeddingScoreMultiplier = 1.0

val consumerEmbeddingScoreMultiplier = 1.0

/\*\*

\* Build the scoring store for TopicTweet Ranking based on Default Multipliers.

\* If you want to compare the ranking between different multipliers, register a new

\* ScoringAlgorithm and let the upstream uses different scoringAlgorithm by params.

\*/

def buildTopicTweetRankingStore(

consumerEmbeddingType: EmbeddingType,

producerEmbeddingType: EmbeddingType,

tweetEmbeddingType: EmbeddingType,

modelVersion: ModelVersion,

consumerEmbeddingMultiplier: Double = consumerEmbeddingScoreMultiplier,

producerEmbeddingMultiplier: Double = producerEmbeddingScoreMultiplier

): WeightedSumAggregatedScoreStore = {

WeightedSumAggregatedScoreStore(

List(

WeightedSumAggregatedScoreParameter(

ScoringAlgorithm.PairEmbeddingCosineSimilarity,

consumerEmbeddingMultiplier,

WeightedSumAggregatedScoreStore.genericPairScoreIdToSimClustersEmbeddingPairScoreId(

consumerEmbeddingType,

tweetEmbeddingType,

modelVersion

)

),

WeightedSumAggregatedScoreParameter(

ScoringAlgorithm.PairEmbeddingCosineSimilarity,

producerEmbeddingMultiplier,

WeightedSumAggregatedScoreStore.genericPairScoreIdToSimClustersEmbeddingPairScoreId(

producerEmbeddingType,

tweetEmbeddingType,

modelVersion

)

)

)

)

}

}