package com.twitter.simclusters\_v2.tweet\_similarity

import com.twitter.ml.api.{DataRecord, DataRecordMerger}

import com.twitter.simclusters\_v2.common.ml.{

SimClustersEmbeddingAdapter,

NormalizedSimClustersEmbeddingAdapter

}

import com.twitter.simclusters\_v2.common.SimClustersEmbedding

object ModelBasedTweetSimilaritySimClustersEmbeddingAdapter {

val QueryEmbAdapter = new SimClustersEmbeddingAdapter(TweetSimilarityFeatures.QueryTweetEmbedding)

val CandidateEmbAdapter = new SimClustersEmbeddingAdapter(

TweetSimilarityFeatures.CandidateTweetEmbedding)

val NormalizedQueryEmbAdapter = new NormalizedSimClustersEmbeddingAdapter(

TweetSimilarityFeatures.QueryTweetEmbedding,

TweetSimilarityFeatures.QueryTweetEmbeddingNorm)

val NormalizedCandidateEmbAdapter = new NormalizedSimClustersEmbeddingAdapter(

TweetSimilarityFeatures.CandidateTweetEmbedding,

TweetSimilarityFeatures.CandidateTweetEmbeddingNorm)

def adaptEmbeddingPairToDataRecord(

queryEmbedding: SimClustersEmbedding,

candidateEmbedding: SimClustersEmbedding,

normalized: Boolean

): DataRecord = {

val DataRecordMerger = new DataRecordMerger()

val queryAdapter = if (normalized) NormalizedQueryEmbAdapter else QueryEmbAdapter

val candidateAdapter = if (normalized) NormalizedCandidateEmbAdapter else CandidateEmbAdapter

val featureDataRecord = queryAdapter.adaptToDataRecord(queryEmbedding)

DataRecordMerger.merge(

featureDataRecord,

candidateAdapter.adaptToDataRecord(candidateEmbedding))

featureDataRecord

}

}