package com.twitter.simclusters\_v2.common.clustering

/\*\*

\* Partitions a set of entities into clusters.

\* NOTE: The selection/construction of the cluster representatives (e.g. medoid, random, average) is implemented in ClusterRepresentativeSelectionMethod.scala

\*/

trait ClusteringMethod {

/\*\*

\* The main external-facing method. Sub-classes should implement this method.

\*

\* @param embeddings map of entity IDs and corresponding embeddings

\* @param similarityFn function that outputs similarity (>=0, the larger, more similar), given two embeddings

\* @tparam T embedding type. e.g. SimClustersEmbedding

\*

\* @return A set of sets of entity IDs, each set representing a distinct cluster.

\*/

def cluster[T](

embeddings: Map[Long, T],

similarityFn: (T, T) => Double,

recordStatCallback: (String, Long) => Unit = (\_, \_) => ()

): Set[Set[Long]]

}

object ClusteringStatistics {

// Statistics, to be imported where recorded.

val StatSimilarityGraphTotalBuildTime = "similarity\_graph\_total\_build\_time\_ms"

val StatClusteringAlgorithmRunTime = "clustering\_algorithm\_total\_run\_time\_ms"

val StatMedoidSelectionTime = "medoid\_selection\_total\_time\_ms"

val StatComputedSimilarityBeforeFilter = "computed\_similarity\_before\_filter"

}