package com.twitter.ann.service.loadtest

import com.twitter.ann.common.{Appendable, Distance, EntityEmbedding, Queryable, RuntimeParams}

import com.twitter.ann.util.IndexBuilderUtils

import com.twitter.util.{Future, Stopwatch}

class EmbeddingIndexer {

// Index embeddings into Appendable and return the (appendable, latency) pair

// we need to return appendable itself here because for Annoy, we need to build

// appendable and serialize it first, and then we could query with index directory

// once we are confident to remove Annoy, should clean up this method.

def indexEmbeddings[T, P <: RuntimeParams, D <: Distance[D]](

appendable: Appendable[T, P, D],

recorder: LoadTestBuildRecorder,

indexSet: Seq[EntityEmbedding[T]],

concurrencyLevel: Int

): Future[Queryable[T, P, D]] = {

val indexBuildingTimeElapsed = Stopwatch.start()

val future = IndexBuilderUtils.addToIndex(appendable, indexSet, concurrencyLevel)

future.map { \_ =>

val indexBuildingTime = indexBuildingTimeElapsed()

val toQueryableElapsed = Stopwatch.start()

val queryable = appendable.toQueryable

recorder.recordIndexCreation(indexSet.size, indexBuildingTime, toQueryableElapsed())

queryable

}

}

}