package com.twitter.ann.serialization

import com.twitter.ann.common.EntityEmbedding

import com.twitter.ann.common.EmbeddingType.\_

import com.twitter.ann.serialization.thriftscala.PersistedEmbedding

import com.twitter.bijection.Injection

import com.twitter.mediaservices.commons.codec.ArrayByteBufferCodec

import java.nio.ByteBuffer

import scala.util.Try

/\*\*

\* Injection that converts from the ann.common.Embedding to the thrift PersistedEmbedding.

\*/

class PersistedEmbeddingInjection[T](

idByteInjection: Injection[T, Array[Byte]])

extends Injection[EntityEmbedding[T], PersistedEmbedding] {

override def apply(entity: EntityEmbedding[T]): PersistedEmbedding = {

val byteBuffer = ByteBuffer.wrap(idByteInjection(entity.id))

PersistedEmbedding(byteBuffer, embeddingSerDe.toThrift(entity.embedding))

}

override def invert(persistedEmbedding: PersistedEmbedding): Try[EntityEmbedding[T]] = {

val idTry = idByteInjection.invert(ArrayByteBufferCodec.decode(persistedEmbedding.id))

idTry.map { id =>

EntityEmbedding(id, embeddingSerDe.fromThrift(persistedEmbedding.embedding))

}

}

}