package com.twitter.cr\_mixer.module.similarity\_engine

import com.google.inject.Provides

import com.twitter.ann.common.thriftscala.AnnQueryService

import com.twitter.cr\_mixer.model.ModelConfig

import com.twitter.cr\_mixer.module.EmbeddingStoreModule

import com.twitter.cr\_mixer.module.thrift\_client.AnnQueryServiceClientModule

import com.twitter.cr\_mixer.similarity\_engine.HnswANNSimilarityEngine

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.simclusters\_v2.thriftscala.InternalId

import com.twitter.storehaus.ReadableStore

import javax.inject.Named

import com.twitter.ml.api.{thriftscala => api}

import com.twitter.conversions.DurationOps.\_

import com.twitter.cr\_mixer.model.ModuleNames

import com.twitter.cr\_mixer.config.TimeoutConfig

import com.twitter.cr\_mixer.similarity\_engine.HnswANNEngineQuery

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.GatingConfig

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.SimilarityEngineConfig

import com.twitter.cr\_mixer.thriftscala.SimilarityEngineType

import com.twitter.finagle.memcached.{Client => MemcachedClient}

object TweetBasedTwHINSimlarityEngineModule extends TwitterModule {

@Provides

@Named(ModuleNames.TweetBasedTwHINANNSimilarityEngine)

def providesTweetBasedTwHINANNSimilarityEngine(

// MH stores

@Named(EmbeddingStoreModule.TwHINEmbeddingRegularUpdateMhStoreName)

twHINEmbeddingRegularUpdateMhStore: ReadableStore[InternalId, api.Embedding],

@Named(EmbeddingStoreModule.DebuggerDemoTweetEmbeddingMhStoreName)

debuggerDemoTweetEmbeddingMhStore: ReadableStore[InternalId, api.Embedding],

// ANN clients

@Named(AnnQueryServiceClientModule.TwHINRegularUpdateAnnServiceClientName)

twHINRegularUpdateAnnService: AnnQueryService.MethodPerEndpoint,

@Named(AnnQueryServiceClientModule.DebuggerDemoAnnServiceClientName)

debuggerDemoAnnService: AnnQueryService.MethodPerEndpoint,

// Other configs

@Named(ModuleNames.UnifiedCache) crMixerUnifiedCacheClient: MemcachedClient,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver

): HnswANNSimilarityEngine = {

new HnswANNSimilarityEngine(

embeddingStoreLookUpMap = Map(

ModelConfig.TweetBasedTwHINRegularUpdateAll20221024 -> twHINEmbeddingRegularUpdateMhStore,

ModelConfig.DebuggerDemo -> debuggerDemoTweetEmbeddingMhStore,

),

annServiceLookUpMap = Map(

ModelConfig.TweetBasedTwHINRegularUpdateAll20221024 -> twHINRegularUpdateAnnService,

ModelConfig.DebuggerDemo -> debuggerDemoAnnService,

),

globalStats = statsReceiver,

identifier = SimilarityEngineType.TweetBasedTwHINANN,

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.similarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig = None,

enableFeatureSwitch = None

)

),

memCacheConfigOpt = Some(

SimilarityEngine.MemCacheConfig[HnswANNEngineQuery](

cacheClient = crMixerUnifiedCacheClient,

ttl = 30.minutes,

keyToString = (query: HnswANNEngineQuery) =>

SimilarityEngine.keyHasher.hashKey(query.cacheKey.getBytes).toString

))

)

}

}