package com.twitter.tweetypie

package hydrator

import com.twitter.tweetypie.serverutil.ExtendedTweetMetadataBuilder

import com.twitter.tweetypie.tweettext.Preprocessor.\_

import com.twitter.tweetypie.tweettext.TextModification

import com.twitter.tweetypie.thriftscala.entities.Implicits.\_

object TextRepairer {

def apply(replace: String => Option[TextModification]): Mutation[Tweet] =

Mutation { tweet =>

replace(getText(tweet)).map { mod =>

val repairedTweet = tweet.copy(

coreData = tweet.coreData.map(c => c.copy(text = mod.updated)),

urls = Some(getUrls(tweet).flatMap(mod.reindexEntity(\_))),

mentions = Some(getMentions(tweet).flatMap(mod.reindexEntity(\_))),

hashtags = Some(getHashtags(tweet).flatMap(mod.reindexEntity(\_))),

cashtags = Some(getCashtags(tweet).flatMap(mod.reindexEntity(\_))),

media = Some(getMedia(tweet).flatMap(mod.reindexEntity(\_))),

visibleTextRange = tweet.visibleTextRange.flatMap(mod.reindexEntity(\_))

)

val repairedExtendedTweetMetadata = repairedTweet.selfPermalink.flatMap { permalink =>

val extendedTweetMetadata = ExtendedTweetMetadataBuilder(repairedTweet, permalink)

val repairedTextLength = getText(repairedTweet).length

if (extendedTweetMetadata.apiCompatibleTruncationIndex == repairedTextLength) {

None

} else {

Some(extendedTweetMetadata)

}

}

repairedTweet.copy(extendedTweetMetadata = repairedExtendedTweetMetadata)

}

}

/\*\*

\* Removes whitespace from the tweet, and updates all entity indices.

\*/

val BlankLineCollapser: Mutation[Tweet] = TextRepairer(collapseBlankLinesModification \_)

/\*\*

\* Replace a special unicode string that crashes ios app with '\ufffd'

\*/

val CoreTextBugPatcher: Mutation[Tweet] = TextRepairer(replaceCoreTextBugModification \_)

}