package com.twitter.home\_mixer.util.tweetypie.content

import com.twitter.home\_mixer.model.ContentFeatures

import com.twitter.tweetypie.{thriftscala => tp}

object TweetTextFeaturesExtractor {

private val QUESTION\_MARK\_CHARS = Set(

'\u003F', '\u00BF', '\u037E', '\u055E', '\u061F', '\u1367', '\u1945', '\u2047', '\u2048',

'\u2049', '\u2753', '\u2754', '\u2CFA', '\u2CFB', '\u2E2E', '\uA60F', '\uA6F7', '\uFE16',

'\uFE56', '\uFF1F', '\u1114', '\u1E95'

)

private val NEW\_LINE\_REGEX = "\r\n|\r|\n".r

def addTextFeaturesFromTweet(

inputFeatures: ContentFeatures,

tweet: tp.Tweet

): ContentFeatures = {

tweet.coreData

.map { coreData =>

val tweetText = coreData.text

inputFeatures.copy(

hasQuestion = hasQuestionCharacter(tweetText),

length = getLength(tweetText).toShort,

numCaps = getCaps(tweetText).toShort,

numWhiteSpaces = getSpaces(tweetText).toShort,

numNewlines = Some(getNumNewlines(tweetText)),

)

}

.getOrElse(inputFeatures)

}

def getLength(text: String): Int =

text.codePointCount(0, text.length())

def getCaps(text: String): Int = text.count(Character.isUpperCase)

def getSpaces(text: String): Int = text.count(Character.isWhitespace)

def hasQuestionCharacter(text: String): Boolean = text.exists(QUESTION\_MARK\_CHARS.contains)

def getNumNewlines(text: String): Short = NEW\_LINE\_REGEX.findAllIn(text).length.toShort

}