package com.twitter.tweetypie.tweettext

import com.ibm.icu.text.BreakIterator

/\*\*

\* Adapt the [[BreakIterator]] interface to a scala [[Iterator]]

\* over the offsets of user-perceived characters in a String.

\*/

object GraphemeIndexIterator {

/\*\*

\* Produce an iterator over indices in the string that mark the end

\* of a user-perceived character (grapheme)

\*/

def ends(s: String): Iterator[Offset.CodeUnit] =

// The start of every grapheme but the first is also a grapheme

// end. The last grapheme ends at the end of the string.

starts(s).drop(1) ++ Iterator(Offset.CodeUnit.length(s))

/\*\*

\* Produce an iterator over indices in the string that mark the start

\* of a user-perceived character (grapheme)

\*/

def starts(s: String): Iterator[Offset.CodeUnit] =

new Iterator[Offset.CodeUnit] {

private[this] val it = BreakIterator.getCharacterInstance()

it.setText(s)

override def hasNext: Boolean = it.current < s.length

override def next: Offset.CodeUnit = {

if (!hasNext) throw new IllegalArgumentException(s"${it.current()}, ${s.length}")

// No matter what, we will be returning the value of `current`,

// which is the index of the start of the next grapheme.

val result = it.current()

it.next()

Offset.CodeUnit(result)

}

}

}